CA MICS® Resource Management

Planning, Installation, Operation, and Maintenance Guide

Release 12.6.00

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 may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, 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.

Contact CA Technologies

Contact CA Support

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

- 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

Provide Feedback

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

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

Contents

Chapter 4: Operation 629

Chapter 1: Introduction

The Planning, Installation, Operation, and Maintenance Guide (PIOM) is the system administrator's primary source of information for questions about installing and operating CA MICS Resource Management. The following guides are also information sources for the system administrator:

- How to Use the PSP explains the tools and the methodology to use when applying product changes to your CA MICS system.

- The System Modification Guide (SMG) contains a detailed description of the maintenance policies for CA MICS, which are designed to support an environment where user modifications are common. You should refer to this material before making any modifications.

- The System Administrator Guide (SAG) provides detailed information on support services, incident logging, user modifications, and system requirements.

- The MICF Reference Guide has a chapter on MICF administration.

The remaining chapters of the PIOM cover the following topics:

> Chapter 2, Planning for Installation and Use of CA MICS, tells you how to tailor the CA MICS system to your data center's particular needs. The discussion focuses on creating the members of the MICS.PARMS library that guide the CA MICS installation and generation processes.
>
> Chapter 3, Installation, shows the system administrator how to use the members created in MICS.PARMS to install and generate the CA MICS database. This chapter uses checklists to guide you through installing and generating the database.
>
> Chapter 4, Operation, discusses how to run the CA MICS system on a day-to-day basis and highlights the mechanisms built into the system to ensure database integrity and recoverability.

Chapter 5, Maintenance, describes the different categories of CA MICS maintenance available to you. It also explains the activities involved in maintaining CA MICS and tells you how to perform them.

Appendix A, Special CA MICS Processing, documents facilities that allow you to create special-purpose databases containing data for all of the timespans based on raw measurement data. These facilities help you perform special studies and help you create MONTHS-level data from data collected before CA MICS was installed.

If you have questions about the installation, operation, or maintenance of the system that are not answered in the documentation, contact CA Technical Support at http://ca.com/support.

The other guides in the CA MICS documentation set are:

```
***************************
* GUIDES FOR THE END-USER *
***************************
```

CA MICS Database Structure and Content Guide
CA MICS Document Access Guide
CA MICS MICF Reference Guide
CA MICS MICF User Guide

```
***************************************
* GUIDES FOR THE SYSTEM ADMINISTRATOR *
***************************************
```

CA MICS How to Use the PSP
CA MICS Planning, Installation, Operation, and Maintenance Guide (PIOM)
CA MICS Standard Reports Guide
CA MICS System Administrator Guide
CA MICS System Modification Guide

```
******************
* PRODUCT GUIDES *
******************
```

CA MICS Accounting and Chargeback Option Concepts and Overview Guide
CA MICS Accounting and Chargeback Option User Guide
CA MICS Analyzer for TSO Guide
CA MICS Analyzer Option for CA IDMS Guide
CA MICS Analyzer Option for CICS Guide
CA MICS Analyzer Option for DB2 Guide
CA MICS Analyzer Option for IMS Guide
CA MICS Analyzer Option for MeasureWare Guide
CA MICS Analyzer Option for MQSeries Guide
CA MICS Analyzer Option for VAX/VMS Guide
CA MICS Analyzer Option for VM/CMS Guide
CA MICS Analyzer Option for VSE/Power Guide
CA MICS Batch and Operations Analyzer Guide
CA MICS CA ASTEX Option Guide
CA MICS Capacity Planner Option Guide
CA MICS Data Transfer Option for VM/CMS Guide

```
CA MICS Hardware and SCP Analyzer Guide
CA MICS IMS Log Data Extractor Option Guide
CA MICS Network Analyzer Option Guide
CA MICS Performance Manager Option Guide
CA MICS Space Analyzer Option Guide
CA MICS Space Collector Option Guide
CA MICS StorageMate Option Guide
CA MICS System Reliability Analyzer Option Guide
CA MICS Tandem Option Guide
CA MICS Tape Analyzer Option Guide
CA MICS Web Analyzer Option Guide
```

Chapter 2: Planning for Installation and Use of CA MICS

This chapter shows you how to plan for the installation of CA MICS, from the general concepts which must be understood to the details of coding the members of the CA MICS parameter library used by the actual installation process. The organization and staffing tasks are identified as well. If you have a question at any time during your review of the material presented here, feel free to contact CA MICS Product Support.

The mechanics of the CA MICS installation process are documented in Chapter 3 of this manual, which includes a set of checklists that describe each installation step. This process has many steps, but none of them are difficult. The "work" of installing CA MICS is gaining the necessary understanding of your installation and its needs, and translating that understanding into CA MICS parameters. This chapter leads you through all of the planning and parameter specifications required for a successful installation. In particular, this chapter calls for you to make a number of policy decisions, fill out a number of worksheets, and translate the worksheet entries into the corresponding CA MICS parameter library member entries. This effort represents the most time consuming portion of the installation process.

Note: The focus of this chapter is on initial installation of CA MICS. Bringing up a new version of CA MICS at a site where it is currently installed or applying maintenance to the installed system is covered in special write-ups shipped with new releases and maintenance.

We recommend that the installation process be managed as a five-phase project:

Phase 1: Personnel assignment and preparation.

Phase 2: Preparing your site for CA MICS.

Phase 3: CA MICS implementation planning and parameter specification.

Phase 4: CA MICS installation and check-out.

Phase 5: Turnover of CA MICS as a production system.

Phases 1 through 3 are covered in this chapter. The Installation chapter provides a detailed, step-by-step plan for phases 4 and 5. The following are short descriptions of each of the tasks in the above five phases.

Phase 1: Personnel Assignment and Preparation.

Task 1-1: Establish and staff the positions of CA MICS System Administrator and CA MICS Information Coordinator.

Task 1-2: Have these people attend CA MICS courses:

o CA MICS Concepts and Facilities
o Implementing CA MICS

Task 1-3: Have the assigned personnel review the contents of the CA MICS System Administrator Guide so that the latest information on the state of the CA MICS system will be used in succeeding tasks.

Phase 2: Preparing Your Site for CA MICS.

Task 2-1: CA MICS requires CA LMP (License Management Program), one of the CA Common Services for z/OS, before you can install or run CA MICS.

For a CA MICS oriented discussion of CA LMP, see section 3.2.2 of this guide.

For a full description of the procedure for defining the CA LMP execution key to the CAIRIM parameters, see the section on CA LMP in the CA Common Services for z/OS documentation.

Task 2-2: We recommend that a separate SAS library be established for exclusive use by the CA MICS database update process. While any level of SAS may be used for inquiries into the database, CA MICS is extremely sensitive to the version of SAS with which the database is updated.

Task 2-3: Ensure that the data collection processes in place are adequate to support CA MICS. CA MICS may require the use of various options in the data collection programs themselves to provide the best advantages of the product. Review collection options in use at your site for compatibility with CA MICS for each of the products whose data you will analyze with CA MICS.

Task 2-4: Task 2-4 is necessary only if you will be using the CA MICS Space Analyzer.

To begin DASD data collection, ensure that the CA MICS Space Collector is installed and operational. See the CA MICS Space Collector Guide for installation and operation instructions.

Phase 3: CA MICS Implementation Planning and Parameter Specification.

Task 3-1: Reproduce the CA MICS Installation Worksheets (distributed with the CA MICS system).

Task 3-2: Define the structure of the database complex that you will install.

Task 3-3: Define the database complex JCL parameters in JCLDEFC and USERJCL.

Task 3-4: Define the account code structure to be used by the CA MICS products your site is licensed to use. Also define the SAS routines that assign values to the account code data elements. Code the structure and SAS logic in sharedprefix.MICS.PARMS. See the individual product guides for information on account code definition for each product.

Task 3-5: Decide how many CA MICS database units you want to install and what products and data sources you want in each unit. We recommend installing a test database unit.

Task 3-6: Describe the site and computer systems for each database unit. Parameters may be specified at the complex-level in the following members of sharedprefix.MICS.PARMS:

Member	Defines
CPLXSID	Computer systerm description
CPLXDEF	Complex and site characteristics
CPLXZONE	Time zones (shifts) to summarize
JCLDEFC	JCL parameters
USERJCL	User additions to JCL

Parameters for each may be coded at the unit-level in the following members of the prefix.MICS.PARMS library to override and/or supplement complex-level definitions:

Member	Defines
SYSID	Computer system descriptions
SITE	Certain site characteristics
ZONE	Time zones (shifts) to summarize
JCLDEF	JCL parameters
USERJCL	User additions to JCL

Task 3-7: Describe the collected data which each database unit is to process and code CA MICS processing exit routines for the products to be installed in each unit, with parameters to be coded in members in the prefix.MICS.PARMS library. See sections 2.3.2 and 2.3.3 and the individual product guides for lists of these installation planning parameters.

Task 3-8: Study your input data volumes and determine your CA MICS online data retention requirements. Specify these to CA MICS via parameters in prefix.MICS.PARMS.

See the chapter on planning, installation definitions, or parameters for information on the options for each product.

Phase 4: CA MICS Installation and Check-Out.

Task 4-1: Establish catalog entries and pointers.

Task 4-2: Load the distribution libraries.

Task 4-3: Run the job in sharedprefix.MICS.CNTL(JCLGEN0) to provide the database complex level CLISTs and JCL procedures.

Task 4-4: Add the procedures produced in 4-3 to the installation procedure library (PROCLIB).

Task 4-5: Load the SAS tables and screens using the SAS LDSASFLS job.

Task 4-6: Alter the CA MICS MONEY format if you use a currency other than U.S. dollars or U.K. Pounds Sterling. To use dual currency MONEY formats to support euro conversion, see the CA MICS Accounting and Chargeback User Guide, Appendix C.

Task 4-7: Run the DOCAGEN1 job to build the ISPF tables needed by CA MICS Document Access.

Task 4-8: Initialize the CA MICS Workstation Facility (MWF).

Task 4-9: For each product being installed, use the CA MICS Component Generator to produce the code and generated documentation for the product.

```
********************************************************
* NOTE:  Tasks 4-10 through 4-18 must be repeated for  *
*        each database unit being installed.           *
********************************************************
```

```
 Task 4-10: Establish catalog entries and pointers.

 Task 4-11: Create the CA MICS system libraries needed for
            a database unit.

 Task 4-12: Run the job in prefix.MICS.CNTL(JCLGENA) to
            provide JCL procedures for the unit database
            installation process.

 Task 4-13: Copy the product-unique parameters into the
            unit parameter library with the COPYLIBU job.

 Task 4-14: Add the CA MICS procedures to the installation
            procedure library (PROCLIB).

 Task 4-15: Edit parameter members

 Task 4-16: Submit the CA MICS Parameter Generation job to
            create the SAS code necessary to tailor the
            CA MICS products that you will be installing
            in the database unit.

 Task 4-17: Run the job in prefix.MICS.CNTL(JCLGENC) to
            provide the necessary JCL procedures for the
            unit database installation and operation.

 Task 4-18: Configure, allocate, and initialize the
            CA MICS database.

 Task 4-19: Run the job in prefix.MICS.CNTL(JCLGEND) to
            provide the necessary JCL procedures required
            for the standard CA MICS operation.

 Task 4-20: Test all exits, account codes, application
            unit, and relative longevity code assignments
            described in Task 3-4 above.

********************************************************
*   End of steps to perform for each database unit.    *
********************************************************
```

Task 4-21: Print the CA MICS documentation.

Task 4-22: Check that CA MICS is properly established in your DASD and tape management systems and that offsite backup tape storage has been arranged.

Task 4-23: Review all generated code and JCL for conformance to installation standards.

Phase 5: Turnover of CA MICS as a Production System.

Task 5-1: Activate a test system and run the prototype database for 3 to 7 days.

Task 5-2: Code and process each of the student exercises defined in the Implementing CA MICS Course using the prototype database.

Task 5-3: Initiate a regular daily database update, but with manual scheduling. Operate in this mode for three to seven days.

Task 5-4: Switch over to automatic scheduling of the CA MICS update process and turn the system over to the group in your shop responsible for production batch job scheduling.

Task 5-5: Begin using SAS interactive inquiry facilities for problem tracking, one-time questions, etc.

Task 5-6: Begin using MICF and Documentation Access for database inquiry, document browse, data element browse, and subject cross reference.

Task 5-9: Execute the Exception Value Analysis process and set "rough-cut" exception values to address your installation's requirements.

Task 5-10: Adjust the exception thresholds and refine the daily exception reporting process to satisfy your installation's requirements.

Task 5-11: Review the Management Objective Reports and tailor their options to address your installation's requirements.

Task 5-12: Convert existing installation reporting to operate using the CA MICS database. Gradually consolidate your installation's reporting process.

Task 5-13: Define an overall management reporting plan for using the CA MICS system as an integral part of your installation's management process.

Areas that may be included are the following:

- Capacity Planning
- Accounting or Chargeback
- Standards Enforcement
- Programmer Productivity Analysis
- Performance Management
- Network Service
- Storage Management

Task 5-14: Define an application interfacing plan for using extractions from the CA MICS Database system in other strategic applications. Consider use of the MICF CSV file generation facility and or CA MICS Query and Reporting to accomplish this task.

Task 5-15: Define the report distribution and use plan.

The remainder of this chapter covers the following topics:

1 - Personnel Assignment and Preparation
2 - Preparing Your Site for CA MICS
3 - CA MICS Installation Planning and Parameter Specification

Phases 4 and 5 are described in detail in Chapter 3, Installation.

Chapters 2 and 3 enable you to install CA MICS in a simple, straightforward fashion. The CA MICS Product Support Group is available to answer questions or assist if you have problems.

This section contains the following topics:

2.1 Personnel Assignment and Preparation

The CA MICS system is a large, interrelated application system, similar to ones that are run by your installation for major users of your computing facilities. The CA MICS system revolves around its database, and database architecture requires additional operational considerations. As a result, a regular schedule of operation must be followed, and a thorough understanding of the update, archive, restart, and reporting processes is required.

The CA MICS system provides an organization with many benefits. You must, however, recognize that the operation and care of the system is important, if not critical, to its effective application in satisfying your information handling needs.

There are two distinct job responsibilities related to the care and feeding of CA MICS: system administration and information coordination. These two areas may be delegated to different individuals, or may be assumed by the same person, depending upon your management's wishes.

The CA MICS System Administrator should assume the following responsibilities:

- Liaison between your installation and CA Technical Support to handle problem identification and resolution, standard maintenance, etc.

- Keeping the installation's copy of the CA MICS System Administrator's Guide (SAG) up-to-date by correctly filing correspondence from CA Technical Support, and also keeping required system-specific notations (for example, user modifications) in this manual. Of course, besides maintaining the SAG, the CA MICS System Administrator must make those changes called for in the correspondence which are his or her responsibility, and make sure all other persons are made aware of information relevant to their concerns.

- Ensuring the correct operation of the required daily, weekly, monthly, and yearly functions. This requires checking the runs to determine that they successfully completed, and if not, carrying out the necessary restart/recovery steps in a timely manner to restore CA MICS to operational status.

- Application of maintenance releases, improvements, and changes to the CA MICS system at your installation.

- Installation of new CA MICS products.

- Management of user modifications to CA MICS, including the approval of such modifications, their implementation with regard to CA MICS maintenance policies and guidelines, and the logging and documentation of the modifications.

- Responsibility for the operation of the CA MICS Job Scheduling Facility, System Restart Facility, and MICF.

The CA MICS Information Coordinator should assume the following responsibilities:

- Definition and implementation of a plan for the distribution, use, and collection of feedback about the standard reports provided by CA MICS.

- Definition and integration of user-written exceptions into the CA MICS Exception reporting facility.

- Definition and provision of user-written reports from the CA MICS database.

- Authorization and control of the use of MICF.

- Consultation and interpretation of the CA MICS standard and user-written reports for users.

How and to whom the position(s) report within an organization is the responsibility of each installation to define. We feel that the System Administration responsibility should rest with the organization closest to the day-to-day use of the system, which in most organizations would be the computer performance and evaluation department/group. The Information Coordinator may also follow this path or may report higher in the organization, closer to, or directly reporting to, the director of data processing.

The personnel resource commitment required to staff CA MICS is a function of a number of things:

- The number of installations and/or CPUs encompassed under CA MICS control.

- The number of CA MICS products that are to be used.

- The number of uses for which CA MICS will be employed, including performance evaluation, system tuning, capacity planning, data center chargeback or accounting, standards enforcement, security, management progress reporting, adhoc inquiry, etc.

Installation

The actual installation, including preparation work, attending the one-week CA MICS class, installation planning, actual installation, and system certification generally requires three to six weeks.

Operation

The operation of CA MICS on a daily, weekly, monthly, and yearly basis requires constant supervision, even though the system is designed for minimum operational impact. The CA MICS System Administrator will be required to expend 15-30 minutes per day on the average in certifying the system's successful operation and database integrity.

Maintenance

The actual maintenance of the CA MICS system, in terms of applying distributed corrections, improvements, new releases, etc., should require less than one day per month on the average.

Use

How each organization chooses to apply and use CA MICS is up to them. It is our recommendation that the CA MICS Information Coordinator dedicate 50-100% of their time to the use of CA MICS in the role of "information specialist".

The time spent is 100% productive use of the individual's efforts toward the performance, capacity planning, tracking, and organizational problem solving, and is a personnel cost that is incurred one way or the other, whether CA MICS is installed or not.

The following tasks should be accomplished in order to prepare the CA MICS System Administrator and Information Coordinator for their responsibilities. They are repeated here from Section 2 for convenience.

Task 1-1: Establish and staff the positions of CA MICS System Administrator and CA MICS Information Coordinator positions.

Task 1-2: Have these people attend CA MICS courses:

- o CA MICS Concepts and Facilities
- o Implementing CA MICS

Task 1-3: Have the assigned personnel review the contents of the CA MICS System Administrator's Guide (SAG) so that the latest information on the state of the CA MICS system will be used in succeeding tasks.

2.2 Preparing Your Site For CA MICS

There are a number of actions which must be taken to prepare the environment at your site for the successful installation and operation of CA MICS. These are discussed in the following sections:

1 - List of Tasks
2 - Resource Planning for CA MICS
3 - SAS Library Maintenance

2.2.1 List of Tasks

The following tasks must be accomplished in order to prepare your site for installing and operating CA MICS. They are summarized here from the detail in the beginning of this chapter for your convenience.

Task 2-1: Ensure that the CA LMP service of CA Common Services for z/OS is installed and operational before CA MICS installation.

Task 2-2: We recommend that a separate SAS library be established for exclusive use by the CA MICS database update process. While a higher level of SAS may be used for inquiries into the database, CA MICS is extremely sensitive to the version of SAS with which the database is updated.

Task 2-3: Ensure that the data collection processes in place are adequate to support CA MICS.

Task 2-4: Task 2-4 is necessary only if you will be using the CA MICS Space Analyzer.

To begin DASD data collection, ensure that the CA MICS Space Collector is installed and operational. See the Space Collector Option Guide for installation and operation instructions.

2.2.2 Resource Planning for CA MICS

It is difficult to accurately estimate the resources required to install, operate, maintain, and use a system as large and diverse as CA MICS. The following section provides guidelines that may be used in the initial planning phases.

1 - Computer Resource Requirements
2 - DASD Space Requirements
3 - Tape Usage Requirements

2.2.2.1 Computer Resource Requirements

The computing resources required to operate CA MICS on an ongoing basis are a function of the following:

- The number of installations and/or CPUs encompassed in the CA MICS database(s)
- The CA MICS products used
- The volume of work processed in terms of number of users, commands, jobs, steps, etc.
- The CA MICS processing options selected, including the use of SAS compression facilities, internal step restart, and/or incremental update
- The number of Database Units defined in the complex.

CA MICS is a highly customizable system, giving you the option to choose the optimum processing approach for your installation. In many cases, you will make tradeoffs depending on your specific requirements.

- You can use SAS compression facilities to reduce DASD space requirements; however, some additional CPU resources will be required for compressing and de-compressing data records.
- CA MICS CPU resource usage is directly related to the volume of data (or number of observations) retained in the CA MICS database. CA MICS facilities let you easily control database content, enabling a logical, and controlled trade-off between CPU resource usage and database granularity.
- You have total flexibility for assigning account code definitions to ensure adequate data granularity while controlling the volume of data retained in the CA MICS online and archive database files. In addition, CA MICS timespan masking facilities let you reduce data granularity at higher timespans by dropping selected account codes from the file key structure.
- You have many options for tailoring the database to drop measurements and metrics of lesser value to your installation thereby reducing database update resource requirements. Measurement elements and/or

entire database files may be dropped for a single timespan, or for the entire database as needed to meet your specific requirements.

- Internal step restart facilities provide an "insurance policy" against processing failures whereby you pay a small daily premium (i.e., increased resource usage) for checkpoint/restart processing that lets you resume processing near the point of failure, saving the expense of repeating previously completed processing.

- Incremental update facilities let you spread CA MICS database update processing over multiple, smaller updates throughout the day. Due to the cost of managing and accumulating the multiple incremental database updates, using incremental update facilities will increase total resource consumption; however, it also dramatically reduces end-of-day processing, increases parallelism across products, and can enable you to process more input data than may be practical for processing in a single daily database update process.

2.2.2.2 DASD Space Requirements

The DASD space required to operate CA MICS on an ongoing basis is a function of the following:

- The number of installations and/or CPUs included in the CA MICS databases
- The CA MICS products used
- The volume of work processed in terms of number of users, commands, jobs, steps, online system transactions, etc.
- The CA MICS database CYCLE retention levels
- The use of SAS file compression
- The number of account codes defined by user account code exits and account code timespan masking options
- Database tailoring to drop measurements and metrics of lesser value to your installation
- The use of the multiple work file, internal step restart, and incremental update features.

Remember to include the DASD space required for CA MICS program, documentation, and control libraries when estimating DASD space requirements. Refer to section 3.2, "Load Distribution Libraries", for information about CA MICS system data sets.

CA MICS can use space in the Hierarchical File System. The CA MICS Platform requires at least 10 megabytes of space. Additional space may be required for additional CA MICS DIAs, and features.

SAS Data Libraries are DSORG=PS, which makes them eligible for alteration by space management utilities that free up unused space. Allowing this to happen will cause frequent system B37 abends. You must explicitly exclude all SAS data libraries used in CA MICS from these space management programs.

To avoid problems, either allow adequate secondary space for your CA MICS data files or exclude them from the utilities.

2.2.2.3 Tape Usage Requirements

The number of tapes that are required for the normal operation of CA MICS is a function of numerous user-supplied parameters. The CA MICS files stored on magnetic tape are:

1 - Database Backup Files
2 - Archive Audit Files
3 - Archive Weekly History Files
4 - Archive Monthly History Files
5 - Optional Incremental Update Tape Database

Each of the above categories represents a number of tape files organized as generation data groups and is stored with multiple files on a single tape volume (e.g., all of the backup files are stored on the same tape volume). As a general rule, assume that one tape will be required for each generation data group that is defined for each of the categories (for the number of GDG entries defined see section 2.3.3.2.1, JCL Option Definitions).

For example, specifying 53 GDG entries for the Archive Audit Files, 10 entries for the Database Backup Files, 7 entries for the Archive Weekly History Files, 5 entries for the Archive Monthly History Files, and 10 entries for the monthly backups will require a minimum of 85 tapes. Additional tape volumes will be required if you choose to exploit incremental update facilities with the option of keeping the incremental update database on tape between incremental update executions.

The Archive Audit Files generally require 1 tape volume for each generation and are created by the CA MICS WEEKLY job.

The Archive Weekly History Files normally require 1 to 2 tape volumes per generation and are updated by the CA MICS WEEKLY job. Due to the cumulative nature of these files, they will continue to grow in size (as you accumulate more data in the CA MICS database) until they reach the CUTOFF size specified by parameters in prefix.MICS.PARMS(DBMODEL).

The Archive Monthly History Files normally require 1 to 2 tape volumes per generation and are updated by the CA MICS MONTHLY job. Due to the cumulative nature of these files, they will continue to grow in size (as you accumulate more data in the CA MICS database) until they reach the CUTOFF size specified by parameters in prefix.MICS.PARMS(DBMODEL).

The Database Backup Files generally require 1 tape volume per generation and are created by the CA MICS BACKUP job, which may be run daily, every other day, or weekly. The Backup files are also created by the MONTHLY job. The size of the files will increase as your CA MICS database fills up.

If activated, the optional incremental update tape database feature generally requires 1 tape volume per generation for each product for which you requested this option.

Accounting and Chargeback has additional tape requirements. These are discussed in the Accounting and Chargeback User Guide.

Use of individual product options and exits for storing detail level transaction data files on tape will require additional tape volumes. See the individual product guides for information on product specific tape usage.

2.2.3 SAS Library Maintenance

You license the SAS Program Product from the SAS Institute and the Institute maintains the product. CA MICS uses the SAS Program Product and is very sensitive to changes made to it.

We recommend following the maintenance procedure outlined in the System Requirements section of the CA MICS System Administrator Guide to ensure the successful operation of CA MICS. This section also describes the special considerations and recommended maintenance for your CA MICS copy of SAS.

2.3 Installation Planning and Parameter Specification

The CA MICS installation process is designed to meet the needs of a very diverse set of installations. As such, it has evolved to contain a great many options, but no one piece of it is difficult. The degree of complexity will mainly be determined by the presence or absence of special standards or processing requirements for systems run in your shop.

The CA MICS installation process is similar in some respects to an OS/390 system generation and is performed in a series of separate stages:

Database Complex Definition

Component Installation

Database Allocation

Component Activation

The information which the CA MICS installation process needs to tailor the system must be specified by you in the various members of the partitioned data sets named "MICS.PARMS." The sections that follow explain each specification which you must make so that the CA MICS installation/generation process can customize the system to meet your site's requirements.

```
*************************************************************
*                                                           *
*  CAUTION:  CA MICS does NOT support "dynamic" parameter   *
*            libraries.  Nothing is accomplished by merely  *
*            changing a member of MICS.PARMS.  You MUST     *
*            run the appropriate CA MICS generation         *
*            process for the change to become effective.    *
*                                                           *
*************************************************************
```

One exception to this rule is the parameters specified in prefix.MICS.PARMS(EXECDEF). Changes to this member take effect the next time CA MICS is run. This member is discussed in Section 2.3.5.

Another exception to this rule is the account code derivation exit routines in sharedprefix.MICS.PARMS. Changes to these members take effect the next time CA MICS is run. These routines are discussed in section 2.3.1.7 and in the various product guides.

The parameters you must supply to guide the CA MICS installation and generation process may be broken down into four categories:

Database Complex parameters--those which provide information that applies to the entire database complex rather than to a particular component or database unit.

Operational parameters--those which guide the processing of the CA MICS system. For example, the parameters which tell CA MICS how many account fields you want in the Batch Job file and how to give them values are of this type.

JCL generation parameters--those which tell CA MICS how to tailor its batch jobs, macros, and TSO CLISTs to run properly at your shop. The jobs involved in the installation process itself must be tailored along with those used to actually run and use CA MICS. For example, the parameters that tell CA MICS how to build a job card that will be valid in your shop are of this type.

Database configuration and space parameters--those which tell CA MICS the number of CYCLEs of each type of file that should be kept in the online database, and also how much of what data should be kept on tape. For example, the parameters that tell CA MICS how many cycles to keep of the TSOTSO file in the YEARS timespan are of this type.

The parameters in the above categories feed the JCLGEN, CYCLEGEN, and parameter generator processes.

Note that not all the parameters which guide CA MICS execution are documented in this manual. The CA MICS Management Objective Reports and Exception Reports are tailored to your needs by direct modification of their modules in prefix.MICS.USER.SOURCE. For more information about tailoring CA MICS reports, consult the CA MICS Standard Reports Guide and the individual product guides.

The remainder of this section presents the following topics:

1 - Database Complex Planning and Parameters
2 - CA MICS Operational Planning and Parameters
3 - CA MICS JCL Planning and Parameters
4 - Database Space Modeling Facility
5 - Dynamic Execution Options (EXECDEF)

6 - Dynamic Allocation Parameter Overrides (//PARMOVRD)

2.3.1 Database Complex Planning and Parameters

This section discusses the concept of a CA MICS database complex, and describes how to specify the parameters which apply to the entire database complex.

The following topics are covered in the remainder of this section.

1 - Database Complex Overview
2 - Data Sets Included in an Installed CA MICS System
3 - CA MICS Parameter Generation Overview
4 - Database Complex Definition
5 - Notes on Coding CA MICS Parameters
6 - Database Complex JCL Definitions (JCLDEFC)
7 - Database Complex Account Code Parameters
8 - Complex Option Definitions (CPLXDEF)
9 - Complex-Level Computing System Parameters (CPLXSID)
10 - Complex-Level Time Zone Definitions (CPLXZONE)

2.3.1.1 Database Complex Overview

The purpose of this discussion is to provide an understanding of the relationship of a CA MICS database complex to the specification of the parameters that are the subject of this chapter. We will define database related terms and give some simple examples. A more comprehensive discussion can be found in the System Modification Guide.

A complete and independent CA MICS system contains one and only one database complex. A database complex consists of one or more databases.

There are four types of databases:

```
        UNIT DATABASE
        PRIMARY DATABASE
        TEST DATABASE
        SPECIAL DATABASE
```

A unit database consists of one or more CA MICS products. For example, a unit database could contain the Batch and Operations, Hardware and SCP, and CICS analyzers.

The simplest database complex would contain a single unit database containing CA MICS platform. You must provide the parameters (defined in this guide and in the parameters chapter of each guide) to create each unit database. A unit database requires a complete set of programs to update the files it contains. The input data sources (for example SMF, RMF, CICS) must be processed to update this unit database.

One of the unit databases defined in a database complex is called the primary database. This should normally be the first production database installed in the complex. The primary database takes responsibility for backing up the complex level data libraries, such as tables.

Certain CA MICS libraries (those preceded by the sharedprefix) are made available to all other databases in the database complex and exist in only one place in the complex. All other libraries (those preceded by prefix or tapeprefix) are applicable to a single database.

Consider the case where you want to separate an IMS database from one containing SMF, RMF, and CA TSO/MON PM. The database complex would have a unit database containing the Batch and Operations, Hardware and SCP, and TSO analyzers. This would be the primary database. A second unit database would be defined to contain the IMS Analyzer. Each unit database would have a separate set of parameter definitions and a separate generation process performed on those parameters. The daily update of each unit database would be performed independently. Therefore, any problem with IMS log data would not affect the update of SMF, RMF, and TSO files.

Now suppose you added a second IMS system and wanted to keep it in a different database than the first one (so each could be updated independently) rather than have them in the same database (which is also permissible). You could define another unit database for the second IMS system.

CA MICS allows considerable flexibility in defining a unit database. A component can have only certain timespans active and can also deactivate files and/or selected data elements in files. This is called file tailoring and is explained in the System Modification Guide. The only point to make here is that file tailoring is a global function that takes place at the database complex level.

Consider our example with two unit databases for IMS. They must contain the same timespans, information areas, files and data elements. However, each IMS unit database could have a different number of cycles for each file. Cycles are defined at a level local to a unit database. Of course, the IMS Analyzer contains one information area, but two unit databases could contain the Hardware and SCP Analyzer (which contains two information areas) and the principle would still apply.

You can create a test database in order to test changes to your CA MICS system. While there can be certain technical differences from a unit database (described in the System Modification Guide), for our purposes here we consider it as a specially marked unit database used for testing purposes.

A special database is used to create files for special study purposes. Unlike the unit and test databases, the special database does not use the database complex definitions of timespans, information areas, or files. Therefore, a special database can contain selected files only.

In summary, a database complex can have one primary database and several other units, test and special databases. The primary and unit databases can be viewed as production databases. Each database is generated separately from its own set of parameters, but shares certain libraries and definitions at the database complex level.

2.3.1.2 Data Sets Included in an Installed CA MICS System

CA MICS appears to the operating system as a collection of data sets and a few entries in certain system data sets (e.g., SYS1.PROCLIB). The data sets which make up CA MICS can be divided into four usage types: 1) shared data sets used by multiple CA MICS databases; 2) data sets used by one CA MICS database; 3) data sets which are archival (tape resident) and are used by one CA MICS database; and 4) MVS system data sets.

For a more complete discussion of the data sets which make up a running CA MICS system, refer to section 4.3.10, Data Set Names.

2.3.1.3 Parameter Generation Overview

There are three types of parameter generation processes:

JCL: Custom CLISTs, MACROs, JCL jobs, and PROCs are created. Some are used in the remainder of the generation process, while others are used in the operation of the CA MICS system. JCLGEN0 through JCLGEN2, and JCLGENA through JCLGENG are the jobs used for this generation process. In addition, parameter tables for use in other processes, including the online functions, are created.

MACROs: Operational parameters are translated into tables and into SAS macros that will be executed by the CA MICS system. Some of the macros refer to SAS formats that are generated from parameters. BASPGEN and the generators for each component being installed (for example, SMFPGEN, RMFPGEN, CICPGEN, and so on) are used for this process. They are executed by the job ALLPGEN, which selects the appropriate generators for the components that are being installed.

DATABASE: Database definition parameters are translated into SAS macros that will be executed by the CA MICS system.

Figures 2-1 through 2-3 illustrate the JCL, MACROs, and DATABASE generation process flows.

Note: The consolidated UNITGEN process is a combination of JCL, MACRO, and DATABASE generation processing that consolidates most generation processes for a database unit into a single batch process. The UNITGEN process is not specifically included in figures 2-1 through 2-3, as it is actually a composite of the database unit-level generation jobs described in the following tables. See section 5.6 for more information on the UNITGEN process.

```
+------------------+-------------------+-------------------+
|                  |                   |                   |
|  INPUT           |      JOB          |       OUTPUT      |
+------------------+-------------------+-------------------+
|  sharedprefix.   |                   |  sharedprefix.    |
|  MICS.CNTL       |                   |  MICS.CNTL        |
|                  |                   |                   |
|  sharedprefix.   |    JCLGEN0        |  sharedprefix.    |
|  MICS.GENLIB     |    JCLGEN1        |  MICS.MACAUTOS    |
|                  |    JCLGEN2        |                   |
|  sharedprefix.   |                   |  sharedprefix.    |
|  MICS.PARMS      |                   |  MICS.CLIST       |
|                  |                   |                   |
|  sharedprefix.   |                   |  sharedprefix.    |
|  MICS.PROTOLIB   |                   |  MICS.GENLIB      |
|                  |                   |                   |
|                  |                   |  sharedprefix.    |
|                  |                   |  MICS.ISPTLIB     |
|                  |                   |                   |
|                  |                   |  sharedprefix.    |
|                  |                   |  MICS.MCOLIB      |
+------------------+-------------------+-------------------+
|                  |                   |                   |
|  prefix.         |                   |  prefix.          |
|  MICS.PARMS      |     JCLGENA       |  MICS.CNTL        |
|                  |     JCLGENB       |                   |
|  sharedprefix.   |     JCLGENC       |  prefix.          |
|  MICS.GENLIB     |     JCLGEND       |  MICS.PARMS       |
|                  |     JCLGENE       |                   |
|  sharedprefix.   |     JCLGENF       |  prefix.          |
|  MICS.PROTOLIB   |     JCLGENG       |  MICS.MUOLIB      |
|                  |                   |                   |
|  prefix.         |                   |  prefix.          |
|  MICS.CNTL       |                   |  MICS.USER.SOURCE |
|                  |                   |                   |
|                  |                   |  sharedprefix.    |
|                  |                   |  MICS.GENLIB      |
|                  |                   |                   |
|                  |                   |  sharedprefix.    |
|                  |                   |  MICS.ISPTLIB     |
|                  |                   |                   |
|                  |                   |  sharedprefix.    |
|                  |                   |  MICS.MACAUTOS    |
|                  |                   |                   |
|                  |                   |  sharedprefix.    |
|                  |                   |  MICS.SOURCE      |
+------------------+-------------------+-------------------+
```

Figure 2-1. JCL Generation Process Flow

```
+------------------+-------------------+-------------------+
|  INPUT           |       JOB         |       OUTPUT      |
+------------------+-------------------+-------------------+
|  prefix.         |                   |  prefix.          |
|  MICS.CNTL       |                   |  MICS.CNTL        |
|                  |                   |                   |
|                  |                   |  prefix.          |
|  prefix.         |     ALLPGEN       |  MICS.MUOLIB      |
|  MICS.PARMS      |     cccPGEN       |                   |
|                  |                   |  prefix.          |
|                  |                   |  MICS.PARMS       |
|                  |                   |                   |
|                  |                   |  prefix.          |
|                  |                   |  MICS.USER.SOURCE |
+------------------+-------------------+-------------------+
```

Figure 2-2. Macro Code Generation Process Flow

```
+------------------+-------------------+-------------------+
|  INPUT           |       JOB         |       OUTPUT      |
+------------------+-------------------+-------------------+
|  prefix.         |                   |                   |
|  MICS.CNTL       |                   |                   |
|                  |                   |  Database Space   |
|                  |     DBSIZE        |  Estimator Report |
|  prefix.         |                   |                   |
|  MICS.PARMS      |                   |                   |
|  (DBMODEL)       |                   |                   |
+------------------+-------------------+-------------------+
|                  |                   |                   |
|  prefix.         |                   |                   |
|  MICS.CNTL       |                   |  prefix.          |
|                  |     CYCLEGEN      |  MICS.USER.SOURCE |
|  prefix.         |                   |                   |
|  MICS.PARMS      |                   |                   |
|  (DBMODEL)       |                   |                   |
+------------------+-------------------+-------------------+
```

Figure 2-3. Database Generation Process Flow

2.3.1.4 Database Complex Definition

The CA MICS Database may contain one or more database units. This section discusses the things you should consider in deciding how many database units you should establish, and which components should be in each one.

The remainder of this section presents the following topics.

1 - Planning Multiple Database Units
2 - Multi-System/Site Database Planning
3 - Database Complex Definition Worksheet

2.3.1.4.1 Planning Multiple Database Units

The CA MICS Database may contain one or more database units. The number of database units and the components in each is defined by the user when CA MICS is installed. This section discusses the basic considerations which will affect the number of database units you should install. An example is also presented to illustrate the considerations.

The key to understanding most of the considerations given below is to remember that each of the database units has its own update jobs which are run independently of the update jobs of the other database units.

The basic factors affecting the number of database units you should create are as follows:

1. Installation Size. Small installations or installations which only have a few components installed are more likely to have a single production database unit.

2. Component Data Source. Components which use the same input data source should be kept in the same database unit. The best example of this is the SMF,RMF,TSO, and CIC components which may all receive their input from the SMF tapes.

3. Input Data Availability. Components which obtain input data from different data sources may be put in different database units to avoid the operational problems of getting the different data sources together at the same location and time for a database update.

4. Input volume. Components which have high volumes of input data may be separated from others in order to break the total update time into several smaller jobs.

5. Flexibility and Reliability. Having several database units increases the flexibility in performing update and other operations on the database. In addition, if a problem is encountered in the database, only part of the database is affected.

6. Overhead. There are some increases in overhead for each additional database unit. These include library space for all of the prefix.MICS libraries, additional SAS overhead to repeat the SAS initialization and compilation for each update, and increased operational tasks of managing the execution of additional update job streams.

7. Testing Capability. We recommend that every site install a test database unit. The test unit will provide you with a controlled environment for developing CA MICS additions and modifications without endangering the integrity of your production databases.

An Example

To demonstrate some of the points above, consider the following hypothetical example for a company with a large data center and a smaller center located in a physically separate site. The two centers have the following configurations:

Site A (Main Site):

System	Workload
A001	TSO, batch
A002	TSO, batch, IMS
A003	CMS
A004	IMS
A005	TSO, batch, CICS

Components to be installed:

SMF, RMF, TSO, CIC, IMS, SRL, and MTI field developed applications for integrating NSM Performance Trend data for distributed systems.

Site B (Small Site):

System	Workload
B001	TSO, batch

Components to be installed:

SMF, RMF, TSO, CIC, SRL

Notes:

1. System A001 is the primary TSO machine.

2. System A004 is a dedicated IMS machine processing 5,000,000 transactions per day for a major IMS application. All other IMS applications run on system

A002 which processes 1,000,000 transactions per day.

3. CMF is used as the input source for CICS.

A total of seven database units is appropriate for site A, and 1 database unit for site B. The database units for site A contain the following components:

unit 1 - SMF, RMF, TSO (which all use SMF tapes as input.)

unit 2 - CIC (separated because of data volume)

unit 3 - IMS (for system 4; separated because of data volume)

unit 4 - IMS (for system 2; all other IMS work)

unit 5 - SRL (separated because the input data source is different)

unit 6 - MTI Field Developed Applications

unit 7 - a test database to develop and test user modifications.

The CA MICS database for site B processes a much smaller volume of input data, and so all components are located in one database unit. A test database unit might also be installed at this site if user modifications are to be developed here. If software development and control is done from the main site then the test database unit there would suffice.

2.3.1.4.2 Multi-System/Site Database Planning

The CA MICS Database may contain data from one or more computing systems. If you have multiple OS/390 systems or multiple distributed systems, then some thought should be given to the best CA MICS database configuration for your installation's requirements. If your installation only has one OS/390 System for which data will be maintained by CA MICS, then the issues discussed in this section may not be pertinent.

To CA MICS, a "computing system" is identified by a unique SYSID -- One SYSID, one system; two SYSIDs, two systems. For example, OS/390 multiprocessors (MPs) operating as a single LPAR are single systems to CA MICS because they are associated to a single SMF SYSID. On the other hand, "loosely" coupled systems, as one finds in an OS/390, JES3 complexes, or SYSPLEX JES2 Multi-Access Spool environments, are multiple systems to CA MICS (note, however, that the data from all member systems of a loosely-coupled complex must go into the same CA MICS database).

You should do the following when planning your database(s) and its(their) configuration:

1. Ensure that unique computing system identifications (SMF "SYSID"s) are being used on all the systems to be included. You must NEVER be running the same SYSID on two systems at once.

2. If you intend to put data from more than one system in a database, ensure that you have in place an operational process to bring the measurement data together from the different processors. If this has not yet been worked out, it may be most unwise to demand that it be implemented at the same time as, and for the purpose of supporting, CA MICS. As an alternative, use separate database units to store the data from each of the systems. It is also possible to create multiple database complexes if the systems do not share proclibs. However, this increases overhead needed for CA MICS system library space, and to perform CA MICS installation and maintenance tasks and so is not recommended.

3. Recognize that all machines in a loosely-coupled (as defined above) complex must have their data stored in a single database. This is because in such environments, a job may be read in on one CPU,

executed on another, and printed on a third. SMF records for such a job will be cut on all three systems, and the proper operation of CA MICS requires that all the data for the job be brought together for processing.

The concept of a commonly defined and maintained database available at a single site for a data processing department having multiple data centers is appealing for the purposes of corporate capacity planning, organizational control, auditing, etc. CA MICS enables a multi-site organization to take a giant stride closer to achieving the "headquarters database".

There is more than one way of implementing such a centralized database. We believe that it would be a mistake for most organizations to attempt to transport raw measurement data to the central point in order to feed CA MICS. Rather, CA MICS should be installed at each installation using the guidelines presented above. Periodically, say monthly, the headquarters or central office could easily extract the desired data files (if it is done monthly, the CA MICS MONTHS files would be the ones of interest) from each CA MICS database, and construct a single headquarters database containing the selected data for all computing systems.

Our suggestions for achieving this objective are as follows:

1. Ensure that unique and identifiable computing system identifications (SMF "SYSID"s) are being used on all systems to be included.

2. Agree upon standards governing the data files that will be maintained and agree on retention periods, so that each system can be defined to incorporate these decisions.

3. Define the data collection interval, which should be no less than weekly, but no greater than quarterly.

4. Define how the data will be transmitted at the defined collection intervals. Teleprocessing is, of course, attractive, but the most practical way may be simply writing the required CA MICS files on a tape in SAS format, mailing it to headquarters, and having headquarters personnel load the data into the headquarters database.

5. Use separate CA MICS database units to store the data collected from other sites.

We expect that there will be many organization-specific considerations involved in establishing a headquarters database and reporting structure. In designing your approach to a headquarters database, please feel free to contact the CA MICS Product Support Group for assistance.

2.3.1.4.3 Database Complex Definition Worksheet

The CA MICS Database may contain one or more database units. This section provides a worksheet on which you can define the database units for the complex which you are planning to install. This worksheet is not used by CA MICS in the installation process, but is meant as a reference for you. Some of the information contained on this worksheet is repeated in other parameter members, and is included here for convenience. The definitions on this worksheet are made when the complex is first installed, and then reviewed and modified any time the complex is changed.

The worksheet for defining the Database Complex is contained in Figure 2-4. One set of definition statements should be completed for each database unit that you plan to install. The statements are explained below. Most of the information provided in these statements is also used in a parameter member. When this is the case, a reference to the section discussing the parameter member is listed in parentheses after the description. More detailed discussions of the information requested may be found in the referenced section.

Sharedprefix: The name used to form the data set names for data sets shared by all database units in the complex. The SHAREDPREFIX must be different from all PREFIXes and TAPEPREFIXes. (2.3.3.2.1.1)

Database Name: A 1 to 8 character name used to identify the database unit during installation. (section 2.3.3.2.1.1)

Database Identifier: A 1 character (alphanumeric) ID used to form a unique name for JCL procedures, CLISTs, and macro names which are generated for each database unit. (2.3.3.2.1.1)

Database Type: PRIMARY, UNIT, or TEST. The first unit installed should be a TEST unit. The first production unit installed should be designated the PRIMARY unit. Other production units are given a type of UNIT. TEST is treated differently by the JCL generation process and by CA MICS Accounting and Chargeback. (2.3.3.2.1.1)

Components: The list of components to be installed in this unit. The list is based on the products your site is licensed for. Note that the Base Component must always be installed. (2.3.3.2.1.1)

ORGSYSIDs: The original (SMF) sysid of the input data. (2.3.2.2)

LOGSYSIDs: A logical system ID. It is mapped back to the ORGSYSID. Most CA MICS files are organized with this ID as the highest sort key. (2.3.2.2)

Prefix: The name used to form the data set names for data sets associated with a database unit. The PREFIX must be different from all other PREFIXes, and SHAREDPREFIXes. (2.3.3.2.1.1)

Tapeprefix: The name used to form the data set names for all of the tape data sets associated with a database unit. The TAPEPREFIX must be different from all other TAPEPREFIXes, and SHAREDPREFIXes. (2.3.3.2.1.1)

Archiving: Indicate the types of archiving you wish performed for each database unit. (2.3.3.2.1.3)

```
+--------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET: Database Complex Definition          |
|                                                                          |
|                                                                          |
| Reference Sections: 2.3.1.4                                              |
+--------------------------------------------------------------------------+
|                                                                          |
|  Sharedprefix:     ______________ (Maximum of 14 characters)             |
|                                                                          |
|                                                                          |
|                                                                          |
|       Database Name:   ________ (max of 8 characters)                    |
|       Database Identifier  _    (A - Z, 0 - 9, blank not allowed)        |
|       Database Type:   _______ (PRIMARY, UNIT, SPECIAL, OR TEST)         |
|       Components:  ___  ___  ___  ___  ___  ___  ___  ___                |
|       ORGSYSIDs:   ____ ____ ____ ____ ____ ____ ____ ____               |
|       LOGSYSIDs:   ____ ____ ____ ____ ____ ____ ____ ____               |
|       Prefix:     ______________ (Maximum of 14 characters)              |
|       Tapeprefix: ______________ (Maximum of 14 characters)              |
|       Archiving:  Audit Archiving:           ___ (YES/NO)                |
|                   Weekly History Archiving:  ___ (YES/NO)                |
|                   Monthly History Archiving: ___ (YES/NO)                |
|                                                                          |
|                                                                          |
|       Database Name:   ________ (max of 8 characters)                    |
|       Database Identifier  _    (A - Z, 0 - 9, blank not allowed)        |
|       Database Type:   _______ (PRIMARY, UNIT, SPECIAL, OR TEST)         |
|       Components:  ___  ___  ___  ___  ___  ___  ___  ___                |
|       ORGSYSIDs:   ____ ____ ____ ____ ____ ____ ____ ____               |
|       LOGSYSIDs:   ____ ____ ____ ____ ____ ____ ____ ____               |
|       Prefix:     ______________ (Maximum of 14 characters)              |
|       Tapeprefix: ______________ (Maximum of 14 characters)              |
|       Archiving:  Audit Archiving:           ___ (YES/NO)                |
|                   Weekly History Archiving:  ___ (YES/NO)                |
|                   Monthly History Archiving: ___ (YES/NO)                |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
+--------------------------------------------------------------------------+
| ....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+--------------------------------------------------------------------------+
```

Figure 2-4. Database Complex Definition Worksheet

2.3.1.5 Notes on Coding CA MICS Parameters

The prefix.MICS.PARMS library contains a number of members which specify the options to control various parts of the total CA MICS system. Members either contain control statements or are exit routines coded in SAS.

The format of control statements is free form, but positional, meaning that any number of blanks may come before and after any parameter, but all parameters must be specified, and specified in the order shown. There is no explicit provision for the continuation of control statements. Our approach has been to allow the repetition of control statements when more parameters must be specified than can fit on a single line. Comments are supported in all of the PARMS members. If the first parameter is '*', the parameter line is ignored. For those parameter statements which begin with a keyword, the keyword MAY NOT BE ABBREVIATED--it must be coded exactly as shown in the documentation.

Some PARMS members require you to specify your installation-specific requirements by coding an exit in SAS. This method of tailoring affords you great flexibility because the SAS language is powerful. You should, however, always follow these guidelines:

> Validate the fields your exits use to make their processing decisions. While CA MICS does some basic validation, it cannot catch the kinds of installation-specific errors you can (e.g., an invalid account number on a JOB statement). When one of your exits detects invalid input data, it should never pass it on into the database (e.g., pass through an invalid account number so it becomes part of the key of a record). Doing so increases the size of the database and decreases its usefulness for analysis and reporting. Rather, your exits should take some sort of standard default action when they detect invalid input data (e.g., assign a job with an invalid account code to an overhead category).
>
> Exit code is inserted directly into CA MICS processing logic, so you should ensure the code inserted will not do anything to cause subsequent CA MICS logic to operate abnormally. To help you ensure the validity of your exit code, the following guidelines are presented:
>
> **** DO the following: ****

```
DO:  Use the member supplied in the distributed version
     of the MICS.PARMS library as the starting point for
     your exit.

DO:  Liberally comment all code.

DO:  Examine input record length to be sure INPUT
     requests can be satisfied from the current record.

DO:  Reference input field locations with pointer logic
     where possible.

DO:  Preserve data element integrity by type (numeric,
     time stamp, character variable) and range (having
     the value agree with the FORMAT and LENGTH
     specifications).

DO:  Define new retained data elements by specifying
     the RETAIN statement after LENGTH and FORMAT
     statements.

DO:  Test all exit logic under the CA MICS-required SAS
     release.

DO:  Call the CA MICS Product Support Group if you have
     ANY questions.

        ****  D O N ' T  do the following:    ****


DON'T:  Use columns 73-80 for SAS statements. CA MICS
        only uses 1-72.

DON'T:  Code these SAS statements in your exit:  RETURN,
        DELETE, or INPUT without a trailing '@'.

DON'T:  Code INPUT statements that could cause SAS
        to perform an automatic input statement skip.
        Use the RECLEN variable to be sure the record is
        big enough to contain data at any intended input
        offset.

DON'T:  Forget the CA MICS naming standards, especially
        for DDnames, file names, and data element file
        prefixes.

DON'T:  Assume that any data element is RETAINed or
        not RETAINed.
```

DON'T: Redefine any currently-existing CA MICS data element unless the documentation of the specific exit in question says it is allowed.

DON'T: Assume that the SAS automatic variable _N_ contains the actual input observation number.

DON'T: Assume that setting the SAS automatic variable _ERROR_ will cause an input observation to be ignored or CA MICS processing to terminate.

DON'T: Use SAS language features not supported by the SAS release specified for use by the CA MICS update process.

When specifying a PARMS member, you should always do so by modifying the example member from the MICS.PARMS shipped with the system rather than coding one from scratch. Doing this will spare you annoying syntax errors.

2.3.1.6 Database Complex JCL Definitions (JCLDEFC)

JCLDEFC contains parameters used to generate complex-level JCL. JCLDEF is used for all other JCL generation. The structure of the two members is very similar, but JCLDEFC contains fewer parameters.

There are a few required parameters and many optional ones that you specify in sharedprefix.MICS.PARMS(JCLDEFC) to guide the JCLGEN process. This member contains one statement for each parameter defined.

The format of the statements is free-form but positional. Keywords must NOT be abbreviated. Any line starting with an asterisk is considered a comment. Only parameters that contain embedded or trailing blanks require beginning and ending quotes. A sample member is illustrated in Figure 2-5 and the JCL Definitions worksheet is illustrated in Figure 2-6.

Use the JCLDEFC member in the sharedprefix.MICS.PARMS library shipped with the CA MICS system as a starting point for specifying your options. These options are described in the following sections:

```
    1 - Database Complex Control Definitions
    2 - Database Complex Library Definitions
    3 - Database Complex JCL Definitions
    4 - Database Complex Tape Specifications
    5 - Database Complex Extended JOB Statement Definitions
    6 - Database Complex User Defined Symbols
    7 - JCLGEN Parameter Sharing
```

Note: Many of the parameter values defined in the JCLDEFC member are available as internal JCLGEN variables. For more information about JCLGEN variables, see section 2.3.3.3.3.1, Generation Control Language Variables.

The parameters in the following sample of JCLDEFC are grouped to match the parameter detail descriptions that follow.

```
 *
 * SAMPLE SHAREDPREFIX.MICS.PARMS(JCLDEFC)
 *
  COMPLEXPARMS   YES
 *
 *     CONTROL DEFINITIONS
 *
  SHAREDPREFIX   CPM981.CMN  MICSLEVEL
  TSHAREDPREF    CPM981.CMNT MICSLEVEL
 *
 * HFSROOT is case sensitive
 *
  HFSROOT       '/MICS/CPM981.CMN'
 *
 *     LIBRARY DEFINITIONS
 *
  SASVER          9.20
  SASNAME         SAS
  SASLIB         'CPM981.CMN.MICS.LIBRARY'
  SASAUTOS       'CPM981.CMN.MICS.AUTOLIB'
  SASHELP        'CPM981.CMN.MICS.SASHELP'
  SASMSG         'CPM981.CMN.MICS.SASMSG'
 *SASCONFIG      'CPM981.CMN.MICS.CONFIG(BATCH)'
  SASCONFIG1     'CPM981.CMN.MICS.CONFIG(BATCH)'
  SASCONFIG2     'CPM981.CMN.MICS.CONFIG(COMMON)'
  SASCONFIG3     'CPM981.CMN.MICS.CONFIG(ENW0)'
  SASCONFIG4     'CPM981.CMN.MICS.CONFIG(SITE)'
 *
  PROCLIB        'SYS1.PROCLIB'
 *
  ISPLLIB1       'ISP.SISPLOAD'
  ISPLLIB2       'ISP.SISPLPA'
  ISPMLIB1       'ISP.SISPMENU'
  ISPMLIB2       'ISP.ISRMLIB'
  ISPTLIB1       'ISP.SISPTENU'
  ISPTLIB2       'ISP.ISRTLIB'
 *
 *STEPLIB1       'DPM981.DB1.STEPLIB1'
 *STEPLIB2       'DPM981.DB1.STEPLIB2'
 *SORTLIB        'SYS1.SORTLIB'
 *
 *SYSTEMMODEL   'DPM981.DB1.MODEL'
 *
 *CAILIB         'SYS1.CA90S.LOAD'
 *
  OBJACCESS       COPY
  OBJWAIT         10
 *
 *SORTUTILITY   CASORT
 *
 *     JCL DEFINITIONS
 *
```

```
 ASSEMBLER      HL
 AUTOSUBMIT     YES
 COLIBSPACE     6144 1000 500
 DASDUNIT       SYSDA
 DASDSMSPARMS   STORCLAS=MICSDATA
 DOCUMENT       A
 DYNAMWAIT      15
*INTRDRCLASS
 JES            2
 LOGS           *
 MICSLOG        *
 RENUMBER       YES
 REPORTS        *
 SORTCYL        6
 SORTUNIT       SYSDA
 SORTWKNO       6
 SORTSMSPARMS   STORCLAS=WORK
 WORKSPACE      TRK 500 250
 WORKUNIT       SYSDA
 WORKSMSPARMS   STORCLAS=MICSTEMP
*WORKVOLSER
*
*     TAPE SPECIFICATIONS
*
 ACCTMGDG       85
*DENSITY
 TAPEPARM       UNIT=3480 EXPDT=1999/000 COMP
*TAPEPARM       VOLCOUNT=nn NULLUNIT=unitname NULLSER=volser
*TAPESMSPARM
 TMS            YES
*
*     EXTENDED JOB STATEMENT DEFINITIONS
*
 ACCT          'CA MICS'
 CLASS          A
 GROUP          XPM981
*JCLLIB         NO
 JOBNAME        DPM
*JOB7380
 MSGCLASS       A
 MSGLEVEL       1,1
 NOTIFY         DPM981
*PASSWORD
*PERFORM
```

```
 PROG          'CA MICS'
*PRTY
*REGION
 USER           &SYSUID
*
*
*     USER DEFINED SYMBOLS
*
*USERDEF USR_____  __________________________________________
```

Figure 2-5. Sample JCLDEFC Member

```
                          +----------------------------------------------------------------------------+
                          | INSTALLATION PREPARATION WORKSHEET: Database Complex JCL Definitions      |
                          | PARMS Library Member is JCLDEFC                                            |
                          | Reference Section:  2.3.1.6                                                |
                          +----------------------------------------------------------------------------+
Control Definitions       |*COMPLEXPARMS   _______________  (YES or NO)                                |
                          | SHAREDPREFIX   _______________  (MICSLEVEL or NOMICSLEVEL)                 |
                          | TSHAREDPREF    _______________  (MICSLEVEL or NOMICSLEVEL)                 |
                          |*HFSROOT     '____________________________________________'                 |
                          |                                                                            |
Library Definitions       |*ISPLLIB1    '____________________________________________'                 |
                          |*ISPLLIB2    '____________________________________________'                 |
                          |*ISPMLIB1    '____________________________________________'                 |
                          |*ISPMLIB2    '____________________________________________'                 |
                          |*ISPTLIB1    '____________________________________________'                 |
                          |*ISPTLIB2    '____________________________________________'                 |
                          |*PROCLIB     '____________________________________________'                 |
                          |*SASAUTOS    '____________________________________________'                 |
                          |*SASCONFIG1  '____________________________________________'                 |
                          |*SASCONFIG2  '____________________________________________'                 |
                          |*SASCONFIG3  '____________________________________________'                 |
                          |*SASCONFIG4  '____________________________________________'                 |
                          |*SASHELP     '____________________________________________'  (optional)     |
                          |*SASLIB      '____________________________________________'                 |
                          |*SASMSG      '____________________________________________'                 |
                          |*SASNAME      __________                                                    |
                          |*SASVER       __________                                                    |
                          |*SORTLIB     '____________________________________________'                 |
                          |*STEPLIB1    '____________________________________________'  (optional)     |
                          |*STEPLIB2    '____________________________________________'  (optional)     |
                          | CAILIB      '____________________________________________'                 |
                          |*OBJACCESS   '____________________________________________'                 |
                          |*OBJWAIT      ____          (# minutes to wait for unavailable data sets)   |
                          |*USEROBJLIB  '____________________________________________'                 |
                          |                                                                          |
JCL Definitions           |*ASSEMBLER    __            (H for IEV90, HL for ASMA90, Default is H)    |
                          | AUTOSUBMIT   ___           (YES or NO)                                   |
                          |*COLIBSPACE  '____________________________________________'               |
                          | DASDUNIT     ________      (generic name)                                |
                          | DASDSMSPARMS ________      (SMS data set allocation parameters)          |
                          |*DIRSTATS     ____          (YES or NO)                                   |
                          | DOCUMENT    '______ '      (SYSOUT class parameters - upper/lower case)  |
                          | DYNAMWAIT    ____          (# minutes to wait for unavailable data sets) |
                          | INTRDRCLASS  ___           (class parameter)                             |
                          | JES          _             (2 or 3)                                      |
                          | LOGS         __________    (SYSOUT class parameters - class,program,form)|
                          |*MICSDMDELIM   _            (default ~)                                   |
                          | MICSLOG      __________    (SYSOUT class parameters - MICSLOG output)    |
                          | RENUMBER     ___           (YES or NO)                                   |
                          | REPORTS      __________    (SYSOUT class parameters - class,program,form)|
                          | SORTCYL      __            (up to 999 cylinders)                         |
                          | SORTUNIT     ________      (generic name)                                |
                          | SORTSMSPARMS ________      (SMS data set allocation parameters)          |
                          |*SORTUTILITY  _________     (sort product)                                |
                          |*SORTWKNO     ________      (number of sortwk data sets - 0 to 99)        |
                          |*STEPCPULIMIT ____          (YES or NO)                                   |
                          | WORKSPACE    ___  ___  ___ (CYL or TRK, primary, secondary)              |
                          | WORKUNIT     _______       (generic name)                                |
                          | WORKVOLSER   _______       (volume name)                                 |
                          | WORKSMSPARMS ________      (SMS data set allocation parameters)          |
```

```
Tape Specifications       | TAPEPARM                   (one or more TAPEPARM parameters)             |
                          |          UNIT=________    (generic name)                                 |
                          |         EXPDT=_______     (YYDDD or YYYY/DDD)                            |
                          |       NOEXPDT             (omit both EXPDT= and RETPD= from DD stmts.)   |
                          |        NOCOMP vs. COMP    (no tape data compaction -- IDRC=NOCOMP        |
                          |                            tape data compaction -- IDRC=COMP)            |
                          |       NOSTACK vs. STACK   (non-stacked tapes = 1 file per tape volume;   |
                          |                            default STACK = multiple files per tape vol. )|
                          |         NOAFF vs. AFF     (no UNIT=AFF on DD stmts. = 1 unit per file    |
                          |                            default AFF = specify UNIT=AFF on DD stmts.)  |
                          |      VOLCOUNT=___         (default 5)                                    |
                          |      NULLUNIT=________    (generic name)                                 |
                          |       NULLSER=______      (dummy volume serial number)                   |
                          |         MODEL='____________________________________________'             |
                          | TAPESMSPARM   ________     (SMS data set allocation parameters)          |
                          | ACCTMGDG      __           (default 85)                                  |
                          | DENSITY       _            (3 for 1600 bpi 4 for 6250 bpi or 3480)       |
                          |*SYSTEMMODEL  '____________________________________________'              |
                          | TMS           _______      (tape management system, YES/name/NONE/LOCAL) |
                          |                                                                          |
Extended Job Statement    | ACCT         '____________________________________________'              |
Definitions               | CLASS         _________    (job class)                                   |
                          | GROUP         ________     (RACF GROUP)                                  |
                          |*JCLLIB        ___          (YES or NO)                                   |
                          | JOBNAME       ________                                                   |
                          | JOB7380       ________                                                   |
                          | MSGCLASS      _            (SYSOUT class)                                |
                          | MSGLEVEL      ______       (Default 2,1)                                 |
                          | NOTIFY        _______      (userid or &SYSUID)                           |
                          | PASSWORD      ________     (RACF PASSWORD)                               |
                          | PERFORM       __           (job performance group, 1 to 999)             |
                          | PROG         '____________________'                                      |
                          | PRTY          __           (job priority, 1 to 15)                       |
                          | REGION        ____         (nnnn -number of K bytes, nnnnK, or nnnM)     |
                          | USER          _______      (RACF USER)                                   |
                          |                                                                          |
                          |                                                                          |
User Defined Symbols      | USERDEF       USR_____  ________________________________________________ |
                          +--------------------------------------------------------------------------+
                                 Parameters marked with a * are eligible for parameter sharing
```

Figure 2-6. Database Complex JCL Definition Worksheet

2.3.1.6.1 Database Complex Control Definitions

The following control statements define the database complex, including the data set name qualifiers for complex-level DASD and tape data sets, and the root sub-directory name for CA MICS Hierarchical File System (HFS) files.

COMPLEXPARMS:

This optional statement controls the default mode for parameter sharing. 'YES' indicates that parameter sharing will be active for all units; 'NO' deactivates parameter sharing. If used, this must be the first statement in JCLDEFC.

See Section 2.3.1.6.7 for more information on JCLGEN parameter cascading. In addition, Section 5.5 provides an optional checklist that shows how to activate JCLGEN parameter sharing for all units in the CA MICS Complex.

SHAREDPREFIX:

This statement defines the prefix used to uniquely identify CA MICS data sets that are common to all CA MICS database units. The SHAREDPREFIX should be different from all of the database unit's PREFIXes and TAPEPREFIXes. SHAREDPREFIX can consist of more than one data set name node (qualifier), but cannot exceed 14 characters in length. The SHAREDPREFIX is used to complete the data set names for the following data sets:

```
sharedprefix.MICS.ASM
sharedprefix.MICS.BIN
sharedprefix.MICS.CLIST
sharedprefix.MICS.CNTL
sharedprefix.MICS.DIC.TEXT
sharedprefix.MICS.DOC.TEXT
sharedprefix.MICS.DTF.INDEX
sharedprefix.MICS.GENLIB
sharedprefix.MICS.HOLD.PARMS
sharedprefix.MICS.HOLD.USER.SOURCE
sharedprefix.MICS.INC.TEXT
sharedprefix.MICS.INCLLIB
sharedprefix.MICS.ISPHLIB
sharedprefix.MICS.ISPMLIB
sharedprefix.MICS.ISPPLIB
sharedprefix.MICS.ISPSLIB
sharedprefix.MICS.ISPTLIB
```

```
          sharedprefix.MICS.LIBLOAD.CNTL
          sharedprefix.MICS.LOAD
          sharedprefix.MICS.LOCALMOD.CNTL
          sharedprefix.MICS.MACAUTOS
          sharedprefix.MICS.MACLIB
          sharedprefix.MICS.MCOLIB
          sharedprefix.MICS.MWFPDS.DATA
          sharedprefix.MICS.MWFSAS.DATA
          sharedprefix.MICS.OBJ
          sharedprefix.MICS.PARMS
          sharedprefix.MICS.PROTOLIB
          sharedprefix.MICS.SASFLS
          sharedprefix.MICS.SCREENS
          sharedprefix.MICS.SOURCE
          sharedprefix.MICS.TABLES
          sharedprefix.MICS.TEST.INCLLIB
          sharedprefix.MICS.TEST.SOURCE
          sharedprefix.MICS.USER.SOURCE
```

TSHAREDPREF:

The TSHAREDPREF statement is optional. It is only used at the complex level if you have CA MICS Accounting and Chargeback installed and you want to maintain a complex level audit tape.

The TSHAREDPREF statement is used to uniquely identify tape data sets created by CA MICS. The prefix can consist of more than one node (qualifier), but cannot exceed 14 characters in length. If this statement is not specified or left blank, the tape data sets will have the same prefix defined on the SHAREDPREFIX statement.

Note that many generation data group (GDG) data sets will be cataloged under this index. If the prefix is a TSO user ID, it may impair that user's use of the LISTCAT command due to the voluminous output.

The TSHAREDPREF statement is used to form the name of the tsharedpref.MICS.MONTHC data set.

HFSROOT:

The name of the CA MICS Root Directory. This is the name of the Hierarchical File System (HFS) sub-directory to be used as the default location for storing HFS files for the CA MICS Platform, CA MICS Data Integration Applications, and CA MICS Management Applications.
Within the CA MICS Root Directory, subdirectories will

contain files for specific CA MICS products and applications.

Note: The name of the root directory must be no longer than 64 characters. If it is, DOC2HTML will abend.

HFSROOT defaults to

```
/MICS/&sharedprefix
```

where &sharedprefix is the value specified on the SHAREDPREFIX statement. Note, the MICSLEVEL/NOMICSLEVEL option does not apply to HFSROOT -- .MICS is never appended to the &sharedprefix value when generating the default HFSROOT string.

See Section 5.7 for more information on the CA MICS Root Directory and HFS file usage in CA MICS.

Generation of CA MICS Data Set Names

You can specify whether or not ".MICS." follows the sharedprefix and tsharedpref when data set names are generated. If you want ".MICS." appended to the data set name, specify the keyword "MICSLEVEL". If you do NOT want ".MICS." appended to the data set name, specify the keyword "NOMICSLEVEL". Place the keyword "MICSLEVEL" or "NOMICSLEVEL" after, but on the same line as, the definition of SHAREDPREFIX and TSHAREDPREF. When "NOMICSLEVEL" is specified, the SHAREDPREFIX and TSHAREDPREF length limit increases from 14 to 19. If no keyword is specified, the default MICSLEVEL is used so that ".MICS." follows the sharedprefix and tsharedpref. Figure 2-7 shows examples of a data set name generated with the same sharedprefix and tsharedpref with both "MICSLEVEL" and "NOMICSLEVEL" specified.

JCLDEF Keyword Value	Data Set Names Formed
SHAREDPREFIX EYCS MICSLEVEL	EYCS.MICS.SOURCE
SHAREDPREFIX EYCS NOMICSLEVEL	EYCS.SOURCE
TSHAREDPREF EYCS MICSLEVEL	EYCS.MICS.ACCT.MONTHC
TSHAREDPREF EYCS NOMICSLEVEL	EYCS.ACCT.MONTHC

As noted earlier, and illustrated below, the

MICSLEVEL/NOMICSLEVEL option does not apply to the derivation of the default HFSROOT value.

JCLDEF Keyword Value	Default HFSROOT Value
SHAREDPREFIX EYCS MICSLEVEL	/MICS/EYCS
SHAREDPREFIX EYCS NOMICSLEVEL	/MICS/EYCS

Figure 2-7. NOMICSLEVEL Data Set Name Formation

Note that more comprehensive control over CA MICS system data set names is available. See Section 2.3.3.3.2.3, "CA MICS User Names Table (JCLNAMES)" for more information.

2.3.1.6.2 Database Complex Library Definitions

Library definitions supply the fully-qualified data set names and other information needed to define the procedure, SAS, sort, and ISPF libraries associated with the database complex.

JCLGEN Parameter Sharing allows commonly used JCLGEN parameters to be specified in sharedprefix.MICS.PARMS(JCLDEFC), where they then become default parameters for prefix.MICS.PARMS(JCLDEF) at the unit level.

Use of this facility can greatly reduce the amount of coding in the JCLDEF and JCLDEFC members.

JCLGEN parameters eligible for sharing can be organized into the following groups:

```
    SAS Data Set Definitions:
    ------------------------
    SASNAME     - SAS program name
    SASCONFIGx  - SAS configuration member
    SASAUTOS    - SAS auto include library
    SASMSG      - SAS message data set
    SASLIB      - SAS executable library
    SASVER      - SAS version
    SASHELP     - SAS help library

    ISPF Data Set Definitions:
    -------------------------
    ISPTLIBx    - ISPF table data set
    ISPLLIBx    - ISPF load library
    ISPMLIBx    - ISPF message data set
    ISPPLIBx    - ISPF panel data set
    ISPSLIBx    - ISPF skeleton data set

    Miscellaneous Definitions:
    -------------------------
    ASSEMBLER   - IBM Assembler level
    COLIBSPACE  - Size of the temporary MCOLIB
    COMPLEXPARMS- Allow/disallow JCLGEN parameter sharing
    DIRSTATS    - ISPF stats for generated members
    HFSROOT     - Name of the CA MICS HFS path
    JCLLIB      - Include JCLLIB statement with job card
    MICSDMDELIM - Delimiter character for MICSDM/MICSDU procs
    OBJACCESS   - Access mode for MCOLIB and MUOLIB
    OBJWAIT     - Wait time for MCOLIB and MUOLIB allocation
    PROCLIB     - Location of CA MICS PROCs
```

SORTLIB	- SORT utility library
SORTUTILITY	- Installation sort utility
SORTWKNO	- Number of sort work data set allocations
STEPCPULIMIT	- Limit CPU time for any step to 60 minutes
STEPLIBx	- User-defined STEPLIBs
SYSTEMMODEL	- Name of model data for GDG creation
USEROBJLIB	- User object definition library

JCLGEN Parameter Sharing is always active, but it must be enabled by specifying the following in sharedprefix.MICS.PARMS(JCLDEFC) or prefix.MICS.PARMS(JCLDEF):

```
COMPLEXPARMS YES
```

When specified in JCLDEFC, the value of COMPLEXPARMS is the default for all units. If specified in JCLDEF, COMPLEXPARMS overrides the COMPLEXPARMS action specified in JCLDEFC.

Section 5.5 provides an optional checklist that shows how to activate JCLGEN parameter sharing for all units in the CA MICS complex.

Notes:

o For the ISPF data set and the STEPLIB definitions, specifying the first definition (that is, ISPTLIB1) will automatically blank out the second definition (that is, ISPTLIB2). Therefore, you must always specify these definitions in numeric order.

o COMPLEXPARMS must be the very first statement in JCLDEFC and JCLDEF. If it is not, an error message is generated.

The ISPxLIBn definitions are used to supply the data set names of your site's IBM ISPF and ISPF/PDF product libraries. The TSO LISTALC command can be used to assist you in determining the correct names to specify for the ISPxLIBn definitions that follow:

ISPLLIB1-2:

The names of your site's IBM ISPF Program Product load libraries. Both parameters are optional and need not be specified if the ISPF load libraries are in your linklist. If ISPF and PDF are combined into one library at your site, you do not need to specify ISPLLIB2.

ISPMLIB1-2:

The names of your IBM ISPF Program Product messages library. ISPMLIB1 is required. If ISPF and PDF are combined into one library, you do not need to specify ISPMLIB2.

ISPTLIB1-2:

The names of your IBM ISPF Program Product table libraries. ISPTLIB1 is required. If ISPF and PDF are combined into one library at your site, you do not need to specify ISPTLIB2.

PROCLIB:

The name of the system PROCLIB that the CA MICS procedures should be added to. JCLGEN creates a job in MICS.CNTL that, when run, adds the CA MICS procedures to the PROCLIB specified instead of modifying it directly. The System Administrator may need authorization to update PROCLIB. The default value for PROCLIB is SYS1.PROCLIB.

SASAUTOS:

The name of the SAS macro autocall library. Sometimes called MACAUTOs.

SASCONFIG, SASCONFIG1-4:

The full SAS data set and member name of the file containing the starting SAS configuration statements. Normally they are contained in member BATCH under SAS Versions 8 and BATW0 under SAS version 9.1.3. Specify SASCONFIG for these SAS versions. SAS version 9.2 and above normally use more than one member. Specify the full data set and member names with SASCONFIG1-4. SASCONFIG1 will override SASCONFIG if both are specified. CA MICS job streams add to or override these statements, using either sharedprefix.MICS.PARMS members MABATCH, ICFBATCH, or MATSO. MABATCH affects batch PROCS except MICSRPTS; ICFBATCH affects MICF batch execution; and MATSO affects MICF foreground, MSAS, and MICF Production and User Reporting (the MICSRPTS PROC).

SASHELP:

The name of the SAS help library.

SASLIB:

The name of the SAS Program Product load library that holds the version of SAS used for CA MICS database update processing. Consult the CA MICS System Administrator Guide for information about the levels of SAS currently supported for updating the CA MICS database.

SASMSG:

The name of the SAS message library.

SASNAME:

The program name of the SAS Program Product in the load library specified in SASLIB or in your LPA libraries. SASXAL is the default value for SASNAME.

Note: If you are using SAS Version 9 or above, you must use a value of SAS.

SASVER:

The version of the SAS system at your site. This is a required field, with no default.

SORTLIB:

The name of the library containing your sort product. This parameter should only be specified when your sort product cannot be loaded through the normal LINKLST process. When this parameter is specified, JCLGEN will add a SORTLIB DD statement to all CA MICS JCL procedures.

STEPLIB1-2:

These optional parameters can be used to add the name of any library to the STEPLIB DD of all CA MICS JCL procedures. STEPLIB1 is included only if specified and, if specified, is included before the SAS program library. STEPLIB2 is included only if specified and always after the SAS program library.

OBJACCESS:

Note: This parameter applies to both complex-level (sharedprefix.MICS.MCOLIB) and unit-level (prefix.MICS.MUOLIB) data sets.

The type of access that operational jobs will have to the sharedprefix.MICS.MCOLIB and prefix.MICS.MUOLIB data sets.

The OBJACCESS parameter is followed by one of the following values:

SHARE
COPY (this is the default value)

Coding a value of SHARE will cause the operational jobs to dynamically allocate the sharedprefix.MICS.MCOLIB and prefix.MICS.MUOLIB data sets with DISP=SHR and with a ddname of MCOLIB or MUOLIB, respectively.

Coding a value of COPY will cause the operational jobs to allocate temporary sharedprefix.MICS.MCOLIB and prefix.MICS.MUOLIB data sets, into which the entire contents of USERFMT1, MICSFMTS, and USERFMT2 format catalogs will be copied.

The OBJACCESS parameter does not affect CA MICS generation jobs, which include cccCGEN, JCLGENx, cccPGEN, and MAGRUN.

When you use the MICSCOM JCL procedure from a unit-level reporting job, the OBJACCESS value specified in sharedprefix.MICS.PARMS(JCLDEFC) will be used instead of the value specified in prefix.MICS.PARMS(JCLDEF).

OBJWAIT:

Note: This parameter applies to both complex-level (sharedprefix.MICS.MCOLIB) and unit-level (prefix.MICS.MUOLIB) data sets.

This optional parameter specifies the amount of time, in minutes, dynamic allocation of sharedprefix.MICS.MCOLIB and prefix.MICS.MUOLIB will wait if the data sets are not immediately available. The default is 5 minutes.

If data set allocation fails because another batch job or online user is already using a data set, CA MICS jobs will wait 15 seconds and then try the allocation again. This allocation will be attempted repeatedly every 15 seconds until, by default, a total of 5 minutes has elapsed. After 5 minutes, the operational job will abort.

If the default value is too short, you may want to use the OBJWAIT parameter to increase the maximum number of minutes the jobs will wait for the data sets to become available.

USEROBJLIB:

The name of the SAS data library that contains user-written informats and formats. User-written informats and formats are created by PROC FORMAT and stored in this SAS data library. You must store your informats and formats in the FORMATS catalog used by CA MICS jobs that implement standard CA MICS JCL procedures.

Your SAS data library is last in the search order for catalogs that contain user-written informats and formats.

When you use the MICSCOM JCL procedure from a unit-level reporting job, the USEROBJLIB value specified in sharedprefix.MICS.PARMS(JCLDEFC) will be used instead of the value specified in prefix.MICS.PARMS(JCLDEF).

CAILIB:

The name of the CA Common Services Library for z/OS.
This is used for CA Examine Auditing integration.
Refer to Section 4.11 in the PIOM for more information.

2.3.1.6.3 Database Complex JCL Definitions

The JCL definitions let you control how JCLGEN tailors the generated JCL in terms of SYSOUT designations, JES usage, DASD and sort generic device names, temporary sort space amounts, sort utility initialization parameters, SAS WORK file allocations, submit internal reader privileges, and generated job member renumbering. The following parameter statements are available:

ASSEMBLER

This option defines which IBM assembler will be used by the MICSASM JCL procedure. The values that can be coded include H for Assembler H (IEV90), and HL for the High Level Assembler (ASMA90). If this option is omitted, it defaults to H.

AUTOSUBMIT

If this option is set to YES, some of the CA MICS jobs will submit jobs through the internal reader. If this option is set to NO, the functions of the jobs that would have been submitted to the internal reader will be included in the jobs that would have submitted them. An example of this is the DAILY job. If this option is YES, then the CA MICS Management Objectives and Exception Reports will not run as part of the DAILY job. Rather, DAILY will submit another job, DAILYRPT, to the internal reader, to create these reports.

The CA MICS jobs that optionally use the internal reader are DAILY, WEEKLY, SCHEDULE, BACKUP, and MONTHLY. Specify either YES or NO to indicate whether the DAILY, WEEKLY, BACKUP, and MONTHLY jobs should submit jobs into the internal reader. If NO is specified, the jobs that would have been submitted become steps in the job that would have submitted them. The SCHEDULE job always uses the internal reader, regardless of this option.

The default value for AUTOSUBMIT is YES.

Note: The MICF inquiry production interface operates from the DAILYRPT, WEEKRPT, and MONTHRPT jobs, but not directly in DAILY, WEEKLY, or MONTHLY. If AUTOSUBMIT is NO, you must schedule MICF inquiry production reporting with your site's own scheduling facilities. For more information about MICF production reporting, see section 5.1.6 of the MICF Reference Guide.

COLIBSPACE

The type of space, primary quantity, and secondary quantity to use for temporary MCOLIB file allocations when operating in OBJACCESS COPY mode.

If the COLIBSPACE parameter is coded, the type of space and primary quantity are required, and the secondary quantity is optional.

The values for type of space are TRK, CYL, or block length (nnnnn). The primary quantity should be a numeric value greater than 1. The secondary quantity should be a numeric value or left blank.

Specify the COLIBSPACE parameter as follows:

```
COLIBSPACE type primary_quantity secondary_quantity
```

If the COLIBSPACE parameter is omitted, the following values will be used:

```
COLIBSPACE 6144 1000 500
```

The type of space value will be saved in a symbol named COLIBSPACE. The primary quantity value will be available in symbol COLIBSPPRIM, while the secondary quantity will be available in COLIBSPSEC.

DASDUNIT

The generic unit name to be used for work data set allocations. The default is SYSDA. DASDUNIT will be used in place of VCATUNIT for allocating the model data set if VCATUNIT is not specified.

DASDSMSPARMS

Optional parameter for specifying System Managed Storage (SMS) keywords (for example, STORCLAS=storclas) to be used for CA MICS data set allocations. Specify one or more SMS parameters according to JCL language syntax, separated by commas. There is no default and continuation is not allowed.

DIRSTATS

Specify either YES or NO. Create ISPF member statistics for all modules generated by JCLGEN. The default value is NO.

WARNING! Activating DIRSTATS can consume up to three times the current number of directory blocks. If a PDS runs out of directory blocks, then the JCLGEN job will abend. Prior to activating, the following CA MICS data sets should be tested to verify they have sufficient directory blocks.

```
sharedprefix.MICS.MACAUTOS
sharedprefix.MICS.CNTL
sharedprefix.MICS.PARMS
sharedprefix.MICS.SOURCE

prefix.MICS.PARMS
prefix.MICS.CNTL
prefix.MICS.USER.SOURCE
```

DOCUMENT

The SYSOUT class parameters to use for printing uppercase/lowercase documentation at the site. In the CA MICS generated JCL, the parentheses are generated, so DO NOT supply them yourself. Class A is the default.

Note: If you want to supply additional information to control documentation output, code this information here by ending the parameter with a trailing blank. This causes the generated right parenthesis to be treated as a comment. For example, you might code the following:

```
DOCUMENT 'A),CHARS=GT12,FCB=8A4 '
```

DYNAMWAIT

This optional parameter specifies the amount of time, in minutes, dynamic allocation processes in CA MICS operational jobs will wait for an unavailable data set. The default is 15 minutes.

Internal Step Restart, Incremental Update, Stand-Alone Archive Tape, and other CA MICS 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, CA MICS operational jobs will wait 15 seconds and then try the allocation again. This allocation will be attempted repeatedly every 15 seconds until, by default, a total of 15 minutes has elapsed. After 15 minutes, the operational job will abort.

If data set contention in your site does cause frequent operational 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 operational jobs will wait for the data set to become available.

On the other hand, if your site standards require that operational jobs fail immediately if required data sets are unavailable, specify the following:

```
DYNAMWAIT 0
```

Note: You can override the DYNAMWAIT parameter:

o At the unit level in prefix.MICS.PARMS(JCLDEF).

o At the product level in prefix.MICS.PARMS(cccOPS). For more information, see the individual DIA guides.

o At execution-time for DAILY and INCRccc jobs using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see section 2.3.6.

INTRDRCLASS

The SYSOUT class used by CA MICS for the internal reader. The default class is A. Normally, this definition is not required since the INTRDRCLASS is the default MSGCLASS for jobs submitted through the internal reader.

JES

Specifies whether the CA MICS operational jobs will be run on a JES2 or JES3 system. Specify 2 or 3. The default JES value is 2.

LOGS

The SYSOUT class parameters for log-type outputs. Class A is the default.

MICSLOG

The SYSOUT class parameters for MICSLOG output. Class specified for LOGS is the default.

MICSDMDELIM

Delimiter character for the MICSDM/MICSDU PROC. This value cannot be alphanumeric or a special character (A-Z, 0-9, @#$). The tilde (~) is the default. This value should only be changed if the tilde cannot be used.

RENUMBER

Specify either YES or NO to indicate whether or not the generated job should be renumbered. The default is NO. This option is forced to NO if JOB7380, discussed in Section 2.3.3.2.1.6, has been specified.

REPORTS

The SYSOUT class parameters for report outputs. Class A is the default.

SORTCYL

The number of cylinders to allocate for each sort work DD statement used in CA MICS procedures. The default value for SORTCYL is 6. The maximum value for SORTCYL is 999.

SORTUNIT

The generic unit name to be used for sort work data set DASD allocations. The default is SYSDA.

SORTWKNO

The number of unique sort work data set allocations. The default is 6. The range of acceptable values is 0 to 99.

SORTSMSPARMS

Optional parameter for specifying System Managed Storage (SMS) keywords (for example, STORCLAS=storclas) to be used for sort work data set allocations. Specify one or more SMS parameters according to JCL language syntax, separated by commas. There is no default and continuation is not allowed.

SORTUTILITY

The name of the sort utility to use. Specify CASORT, DFSORT, SYNCSORT, or OTHER. The default is SYNCSORT.

If CASORT is specified the following JCL is included:

```
//SORTCNTL DD DISP=SHR,
//            DSN=sharedprefix.MICS.PARMS(CASORT)
```

If DFSORT is specified the following JCL is included:

```
//SORTCNTL DD DISP=SHR,
//            DSN=sharedprefix.MICS.PARMS(DFSORT)
```

If SYNCSORT is specified the following JCL is included:

```
//$ORTPARM DD DISP=SHR,
//            DSN=sharedprefix.MICS.PARMS(SYNCSORT)
```

You can modify the appropriate member in sharedprefix.MICS.PARMS to accommodate any unique requirements for sort initialization parameters.

No statements are generated if OTHER is specified.

(If you want to concatenate your load libraries either before or after the CA MICS and SAS load libraries, see Section 4.4.7.)

STEPCPULIMIT

Specify YES or NO. If YES, then JCLGEN will add TIME=60 to the EXEC statement of each PROC. This is done to prevent "runaway" steps.

Note: This option will apply to select PROCs. The default is YES.

UOLIBSPACE

The type of space, primary quantity, and secondary quantity to use for temporary MUOLIB file allocations when operating in OBJACCESS COPY mode.

If the UOLIBSPACE parameter is coded, the type of space and primary quantity are required, and the secondary quantity is optional.

The values for type of space are TRK, CYL, or block length (nnnnn). The primary quantity should be a numeric value greater than 1. The secondary quantity should be a numeric value or left blank.

Specify the UOLIBSPACE parameter as follows:

```
UOLIBSPACE type primary_quantity secondary_quantity
```

If the UOLIBSPACE parameter is omitted, the following values are used:

```
UOLIBSPACE 6144 200 100
```

The type of space value will be saved in a symbol named UOLIBSPACE. The primary quantity value will be available in symbol UOLIBSPPRIM while the secondary quantity will be available in UOLIBSPSEC.

WORKSPACE

The type of space, primary quantity, and secondary quantity to use for SAS WORK file allocations.

If the WORKSPACE parameter is coded, the type of space and primary quantity are required, and the secondary quantity is optional.

The values for type of space are TRK or CYL. The primary quantity should be a numeric value greater than 1. The secondary quantity should be a numeric value or should be left blank.

Specify the WORKSPACE parameter as follows:

```
WORKSPACE type primary_quantity secondary_quantity
```

If the WORKSPACE parameter is omitted, the following values are used:

```
WORKSPACE TRK 500 250
```

The type of space value will be saved in a symbol named WORKSPACE. The primary quantity value will be available in symbol WORKSPPRIM while the secondary quantity will be available in WORKSPSEC.

WORKUNIT

The generic unit name to be used for SAS WORK file allocations. The default is the value specified for the DASDUNIT keyword.

WORKVOLSER

The volume serial number to be used for SAS WORK file allocations.

WORKSMSPARMS

Optional parameter for specifying System Managed Storage (SMS) keywords (for example, STORCLAS=storclas) to be used for SMS WORK file allocations. Specify one or more SMS parameters according to JCL language syntax, separated by commas. There is no default and continuation is not allowed.

2.3.1.6.4 Database Complex Tape Specifications

Tape specifications let you control the number of generation data group entries maintained for the accounting complex-level audit tape, volume count, tape volume density, retention and expiration dates, as well as the VSAM or ICF catalog used to maintain the tape catalog information.

The accounting complex-level audit tape processes use generation data groups. The accounting processes are described in the Accounting and Chargeback Option User Guide.

The number specified for ACCTMGDG is used for two purposes. First, it defines the number of entries in the respective GDG (BLDG ENTRIES=n). Second, when multiplied by the specified retention factor, it provides the JCL LABEL retention period (RETPD=days) that is used in creating the respective GDG files. The ACCTMGDG files are created monthly; therefore, the retention period specified by CA MICS is the number of GDG entries you specify multiplied by 31 (days). However, any value specified in the JCLDEF EXPDT parameter overrides the calculated RETPD.

ACCTMGDG

This is an optional statement if the Accounting and Chargeback Option is being installed; otherwise it is ignored if specified. If you want to maintain complex-level audit files, specify the number of generations of monthly audit files that are to be maintained in the monthly GDG. The default value is 85, and the allowable range is 0 to 999. If you do not want to maintain complex-level audit files, set the value to 0.

TAPEPARM

Use the TAPEPARM statement to specify parameters that define default processing options for CA MICS tape data sets.

The TAPEPARM statement overrides and replaces many of the old-style individual tape data set statements in order to consolidate tape data set definitions into a single, easy-to-locate statement. The individual old-style statements that are replaced are noted in the detailed descriptions of the TAPEPARM parameters that follow.

You can, optionally, override individual TAPEPARM parameter specifications using statements that apply to specific types of CA MICS tape data sets, such as the CA MICS backup, monthly backup, and archive tape files. As a result, you have the option of establishing a common set of defaults that you override, as required, for specific tape applications.

The general form of the TAPEPARM statement is the keyword TAPEPARM followed by one or more TAPEPARM parameters. TAPEPARM parameters include both keyword parameters and keyword=value specifications. You can specify the TAPEPARM statement as many times as you want in order to specify all of the tape data set definitions you require.

The following summarizes the TAPEPARM parameters:

- Each parameter is described in detail below.

- While the following syntax sample shows only one or two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character-maximum line length). You may also choose to specify only one parameter per line. Either approach is acceptable.

```
TAPEPARM UNIT=unitname VOLCOUNT=n
TAPEPARM EXPDT=expdt / NOEXPDT
TAPEPARM COMP/NOCOMP
TAPEPARM NULLUNIT=unitname NULLSER=nulser
TAPEPARM STACK/NOSTACK  AFF/NOAFF
TAPEPARM MODEL='modeldsn' / MODEL=NOMODEL
TAPEPARM DISK / TAPE  SPACE=spacedef
TAPEPARM VOLSER=volser
```

The following is a sample TAPEPARM specification:

```
TAPEPARM UNIT=CART VOLCOUNT=10 EXPDT=1999/000 COMP
TAPEPARM NULLUNIT=CART NULLSER=NULSER
TAPEPARM MODEL='CPM981.CMN.MICS.MODEL'
```

The following describes the TAPEPARM parameters:

- UNIT=unitname

  Defaults to UNIT=3400-5.

  The generic unit name for tape file allocation. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

  Replaces and overrides the old-style TAPEUNIT statement.

  Note: The ASIS sub-parameter is not required when using the TAPEPARM statement UNIT parameter.

- VOLCOUNT=n

  No default. The OS/390 default volume count of 5 will apply.

  The volume count JCL parameter for CA MICS tape data sets. Specify a decimal number from 1 to 255.

  Replaces and overrides the old-style VOLCOUNT statement.

- EXPDT=expdt

  No default. If no expiration date is specified, a default retention period will be derived as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

  Specify an expiration date definition which replaces the derived default retention period (RETPD) specification(s). This facility is specifically intended for use by installations with tape management systems which require specific EXPDT definitions.

  Note that for some tape management systems, certain EXPDT values will make the tape available for scratching immediately. Such values are NOT desirable for CA MICS tapes.

  The EXPDT=expdt parameter is optional, and, if omitted, CA MICS will derive a RETPD=nnn specification as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

  Note: If you specify an EXPDT=expdt, then you must also specify a value for the TMS statement (e.g., TMS NONE).

  Replaces and overrides the old-style EXPDT statement.

  Note: The EXPDT=expdt and NOEXPDT parameters are mutually exclusive.

- NOEXPDT

  Specify NOEXPDT if you want CA MICS to omit both the EXPDT=date and RETPD=days JCL parameters from generated JCL statements for new tape data set creation. This option is provided for those installations where the tape management system or SMS specifications automatically control tape expiration dates and/or retention periods.

  Note: If you specify the NOEXPDT parameter, you must also specify a value for the TMS statement (e.g., TMS NONE).
  Replaces and overrides the old-style EXPDT statement.

  Note: The NOEXPDT and EXPDT=expdt parameters are mutually exclusive.
- COMP / NOCOMP

  No default. The OS/390 default for the selected tape unit type will apply.

  This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) for a tape device enabled for hardware compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified, no TRTCH subparameter will be generated.

  Replaces and overrides the old-style IDRC statement.
- NULLUNIT=unitname

  Defaults to the tape unit value (UNIT=unitname).

  The generic unit name to be used for tape file allocation for the NULLSER. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

  Replaces and overrides the old-style NULLUNIT statement.

- NULLSER=nulser

  Defaults to nulser.

  The value to be used to signify a dummy tape volume to CA MICS. This value is used for pre-allocating dummy tape files when the system is first run. This technique allows CA MICS to run without error even though certain input files do not exist. A trap in the macro routine recognizes the NULLSER value and generates the appropriate SET statements. If your installation standards prevent using the default value (NULSER), then you must change this value. An example is in the case of automatic tape loaders (ATLs).

  For example, if you are adding a new file to history archive in an existing unit, and the existing history archive tapes are in an ATL, you may need to change your NULLSER value to an actual volume serial number contained in the ATL. This prevents an additional tape device (outside the ATL) from being allocated, as well as the MOUNT for the volume serial NULSER. This is imperative for JES3 customers with ATLs because the job will not continue until the mount is satisfied.

  Replaces and overrides the old-style NULLSER statement.
- STACK / NOSTACK

  Defaults to STACK.

  Specify STACK (the default) to "stack" multiple CA MICS tape files on a single tape volume. This is the normal mode of operation where a minimum number of tape volumes are used and tape mounts are minimized. CA MICS JCL statements are generated with the VOL=REF=ddname construct and unit affinity is enforced (that is, STACK also means AFF).

  Specify NOSTACK to place exactly one CA MICS tape file on a tape volume. With this option the generated CA MICS JCL statements will NOT contain the VOL=REF=ddname construct. You may find this option useful if your implementation of a virtual tape system prohibits "stacked" tape volumes, or if you are directing CA MICS tape files to a DASD pool backed with an archival facility.

o AFF / NOAFF

Defaults to AFF.

Specify AFF to use a single tape unit for all related tape files. This is the normal mode of operation where a minimum number of tape units are used and tape mounts are minimized. CA MICS JCL statements are generated with the UNIT=AFF=ddname construct. AFF is required for the STACK option.

Specify NOAFF to enable allocation of different tape units for each CA MICS tape data set. With this option the generated CA MICS JCL statements will NOT contain the UNIT=AFF=ddname construct. You may find this option useful with certain virtual tape system implementations, or where you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Note, if you specify NOAFF, then NOSTACK is automatically set.

o MODEL='modeldsn' / MODEL=NOMODEL

No default.

The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted, a CA MICS generation job will allocate a default model GDG data set.

If NOMODEL is specified as the model dataset name, the use of the model dataset in the DCB specification will be bypassed.
Replaces and overrides the old-style SYSTEMMODEL statement.

Note: The MODEL='modeldsn' and MODEL=NOMODEL
      parameters are mutually exclusive.

- DISK / TAPE

  Defaults to TAPE.

  Specify DISK if you want to write the dataset to a disk instead of a tape. Specify TAPE to write the dataset to a tape.

  When you specify DISK, make sure that you have both a UNIT parameter with a generic unit name of a disk device and a SPACE parameter defining disk space. You can also define a VOLSER parameter to direct the data to a specific disk.

- SPACE=spacedef

  No default.

  Defines the space allocation parameter for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

- VOLSER=volser

  No default.

  Defines the volume serial number of a disk for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

TAPESMSPARM

Optional parameter for specifying System Managed Storage (SMS) keywords (for example, STORCLAS=storclas) to be used as the default for CA MICS tape data set allocations. Specify one or more SMS parameters according to JCL language syntax. There is no default, and continuation is not allowed.

ACCTMTAPEPARM

Use the optional ACCTMTAPEPARM statement to specify parameters defining processing options for Accounting and Chargeback Monthly Audit tape data sets. When using the Monthly Audit tapes, the ACCTMTAPESMSPARM parameter specifications override the tape data set defaults established by the TAPEPARM statement.

The primary reason to specify the ACCTMTAPEPARM statement is to override the default number of monthly audit GDG entries to be defined (the GDG= parameter). However, your data center may want to direct the Accounting and Chargeback Monthly Audit to a unique tape volume pool, or to a tape device type such as a virtual tape unit.

Note: Directing the Monthly Audit data set to disk with the TAPEPARM DISK parameter is currently not supported.

The general form of the ACCTMTAPEPARM statement is the keyword ACCTMTAPEPARM followed by one or more ACCTMTAPEPARM parameters. ACCTMTAPEPARM parameters include both keyword parameters and keyword=value specifications. You can specify the ACCTMTAPEPARM statement as many times as you want in order to specify all of the tape data set definitions you require.
If you specify a parameter multiple times, the last instance defines the value in effect.

The following summarizes the ACCTMTAPEPARM parameters:

- Each parameter is described in detail next.

- While the following example shows only one or two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character maximum parameter line length). You may also choose to specify only one parameter per line.

Either approach is acceptable.

```
ACCTMTAPEPARM GDG=nnn UNIT=unitname VOLCOUNT=n
ACCTMTAPEPARM EXPDT=expdt / RETPD=nnn / NOEXPDT
ACCTMTAPEPARM COMP/NOCOMP
ACCTMTAPEPARM NULLUNIT=unitname NULLSER=nulser
ACCTMTAPEPARM STACK/NOSTACK AFF/NOAFF
ACCTMTAPEPARM MODEL='modeldsn' / MODEL=NOMODEL
```

The following is a sample ACCTMTAPEPARM statement:

```
ACCTMTAPEPARM GDG=90 VOLCOUNT=7
```

In this example, the TAPEPARM parameter values are used for most of the ACCTMTAPEPARM parameters.

The following describes each of the ACCTMTAPEPARM parameters:

o GDG=nnn

Defaults to 85.

The number of entries to be created in the GDG for each CA MICS Accounting and Chargeback Monthly Audit data set.

Overrides and replaces the ACCTMGDG statement.

o UNIT=unitname

Defaults to the value specified on the TAPEPARM statement or to UNIT=3400-5 if UNIT=unitname is not specified on the TAPEPARM statement.

The generic unit name for tape file allocation. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

Overrides the TAPEPARM statement UNIT=unitname value and/or the old-style TAPEUNIT statement.

o VOLCOUNT=n

Defaults to the value specified on the TAPEPARM statement. If no VOLCOUNT is specified, then the OS/390 default volume count of 5 will apply.

The volume count JCL parameter for CA MICS tape data sets. Specify a decimal number from 1 to 255.

Overrides the TAPEPARM statement VOLCOUNT=nnn parameter or the old-style VOLCOUNT statement.

- EXPDT=expdt

  Defaults to the value specified on the TAPEPARM statement. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

  Specify an expiration date definition which replaces the derived default retention period (RETPD) specification(s). This facility is specifically intended for use by installations with tape management systems which require specific EXPDT definitions.

  Note that for some tape management systems, certain EXPDT values will make the tape available for scratching immediately. Such values are NOT desirable for CA MICS tapes.

  The EXPDT=expdt parameter is optional, and, if omitted, CA MICS will derive a RETPD=nnn specification as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

  Note: If you specify an EXPDT=expdt, then you must also specify a value for the TMS statement (e.g., TMS NONE).

  Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameter, or the old-style EXPDT statement.

  Note: The EXPDT=expdt, RETPD=nnn, and NOEXPDT parameters are mutually exclusive.

- NOEXPDT

  Defaults to the value specified on the TAPEPARM

statement. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

Specify NOEXPDT if you want CA MICS to omit both the EXPDT=date and RETPD=days JCL parameters from generated JCL statements for new tape data set creation. This option is provided for those installations where the tape management system or SMS specifications automatically control tape expiration dates and/or retention periods.

Note: If you specify the NOEXPDT parameter, you must also specify a value for the TMS statement (e.g., TMS NONE).

Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameters, or the old-style EXPDT statement.

Note: The NOEXPDT, EXPDT=expdt, and RETPD=nnn parameters are mutually exclusive.

o COMP / NOCOMP

Defaults to the value specified on the TAPEPARM statement. If neither COMP nor NOCOMP is specified, then the OS/390 default for the selected tape unit type will apply.

This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) for a tape device enabled for hardware compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified, no TRTCH subparameter will be generated.

Overrides the TAPEPARM statement COMP/NOCOMP parameter or the old-style IDRC statement.

o NULLUNIT=unitname

Defaults to the value specified on the TAPEPARM statement or to the tape unit value.

The generic unit name to be used for tape file allocation for the NULLSER. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

Overrides the TAPEPARM statement NULLUNIT=unitname parameter or the old-style NULLUNIT statement.

- NULLSER=nulser

  Defaults to the value specified on the TAPEPARM statement or to NULLSER.

  The value to be used to signify a dummy tape volume to CA MICS. This value is used for pre-allocating dummy tape files when the system is first run. This technique allows CA MICS to run without error even though certain input files do not exist. A trap in the macro routine recognizes the NULLSER value and generates the appropriate SET statements. If your installation standards prevent using the default value (NULSER), then you must change this value. An example is in the case of automatic tape loaders (ATLs).

  For example, if you are adding a new file to history archive in an existing unit, and the existing history archive tapes are in an ATL, you may need to change your NULLSER value to an actual volume serial number contained in the ATL. This prevents an additional tape device (outside the ATL) from being allocated, as well as the MOUNT for the volume serial NULSER. This is imperative for JES3 customers with ATLs because the job will not continue until the mount is satisfied.

  Overrides the TAPEPARM statement NULLSER=nulser parameter or the old-style NULLSER statement.

- STACK / NOSTACK

  Defaults to the value specified on the TAPEPARM statement or to STACK.

  Specify STACK (the default) to "stack" multiple CA MICS tape files on a single tape volume. This is the normal mode of operation where a minimum number of tape volumes are used and tape mounts are minimized. CA MICS JCL statements are generated with the VOL=REF=ddname construct and unit affinity is enforced (that is, STACK also means AFF).

Specify NOSTACK to place exactly one CA MICS tape file on a tape volume. With this option the generated CA MICS JCL statements will NOT contain the VOL=REF=ddname construct. You may find this option useful if your implementation of a virtual tape system prohibits "stacked" tape volumes, or if you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Overrides the TAPEPARM statement STACK/NOSTACK parameter.

o AFF / NOAFF

Defaults to the value specified on the TAPEPARM statement or to AFF.

Specify AFF to use a single tape unit for all related tape files. This is the normal mode of operation where a minimum number of tape units are used and tape mounts are minimized. CA MICS JCL statements are generated with the UNIT=AFF=ddname construct. AFF is required for the STACK option.

Specify NOAFF to enable allocation of different tape units for each CA MICS tape data set. With this option the generated CA MICS JCL statements will NOT contain the UNIT=AFF=ddname construct. You may find this option useful with certain virtual tape system implementations, or where you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Note, if you specify NOAFF, then NOSTACK is automatically set.

Overrides the TAPEPARM statement AFF/NOAFF parameter.

o MODEL='modeldsn' / MODEL=NOMODEL

Defaults to the value specified on the TAPEPARM statement.

The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted, a CA MICS generation job will allocate a default model GDG data set.

If NOMODEL is specified as the model dataset name, the use of the model dataset in the DCB specification will be bypassed.

Overrides the TAPEPARM statement MODEL='modeldsn' parameter or the old-style SYSTEMMODEL statement.

ACCTMTAPESMSPARM

Defaults to the values specified on the TAPESMSPARM statement.

Optional parameter for specifying System Managed Storage (SMS) keywords (for example, STORCLAS=storclas) for CA archive monthly history tape data set allocations. Specify one or more SMS parameters according to JCL language syntax separated by commas. Continuation is not allowed.

Overrides the TAPESMSPARM statement.

TAPEUNIT

Replaced by the TAPEPARM statement UNIT=unitname parameter.

The generic unit name to be used for tape file allocation. The values allowed for this parameter are 2400, 2400-1, 2400-2, 2400-3, 2400-4, 3400-2, 3400-3, 3400-4, 3400-5, 3400-6, 3400-9, and 3480. The default value is 3400-5. Specify the parameter ASIS if a non-device-dependent name (for example, TAPE) is used at your data center. A sample TAPEUNIT statement is:

```
TAPEUNIT TAPE ASIS
```

VOLCOUNT

Replaced by the TAPEPARM statement VOLCOUNT=nnn parameter.

The volume count is to be used for the backup and archive tape data sets. It is a decimal number from 1 to 255. If the VOLCOUNT keyword is omitted or is null, the default volume count of 5 will be used.

IDRC

Replaced by the TAPEPARM statement COMP/NOCOMP parameter.

This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) on a tape device enabled for compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified or left blank, no TRTCH subparameter will be generated.

TMS

This parameter identifies the tape management system in use at your data center. It is required to allow CA MICS to test for EXPDT values that could cause CA MICS tapes to be accidentally scratched. Valid values are YES, the name of a tape management system, or NONE.

If you specify the TAPEPARM statement EXPDT=expdt or NOEXPDT parameter, or the old-style EXPDT statement, then you must also specify a value for TMS (for example, TMS NONE).

EXPDT

Replaced by the TAPEPARM statement EXPDT=expdt or NOEXPDT parameters.

Specify an expiration date definition that replaces the standard retention period (RETPD) definitions for the tape files used in backup, archive audit, archive weekly history, and archive monthly history. This facility is specifically directed for use by data centers that use a tape management system that requires specific EXPDT definitions. Note that for some tape management systems, certain EXPDT values will make the tape available for scratching immediately. Such values are not desirable for CA MICS tapes. This statement is an option, and if omitted, will result in the standard retention period (calculated in the manner described in the previous section) being used on the tape creation DDs. However, if you do specify an EXPDT value, you must also specify a value for the TMS parameter (for example, TMS NONE).

SYSTEMMODEL

Replaced by the TAPEPARM statement MODEL='modeldsn'/ MODEL=NOMODEL parameter.

The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted (or left blank), a CA MICS generation job will allocate a model GDG data set.

If NOMODEL is specified as the model data set name, the use of the model data set in the DCB specification will be bypassed.

DENSITY

This old-style statement is rarely used and is retained for compatibility with older tape devices.

The tape density to be used for the backup and archive tape data sets (3 for 1600 bpi or 4 for 6250 bpi).

TAPEUNIT

Replaced by the TAPEPARM statement UNIT=unitname parameter.

The generic unit name to be used for tape file allocation. The values allowed for this parameter are 2400, 2400-1, 2400-2, 2400-3, 2400-4, 3400-2, 3400-3, 3400-4, 3400-5, 3400-6, 3400-9, and 3480. The default value is 3400-5. Specify the parameter ASIS if a non-device-dependent name (for example, TAPE) is used at your data center. A sample TAPEUNIT statement is:

```
TAPEUNIT TAPE ASIS
```

VOLCOUNT

Replaced by the TAPEPARM statement VOLCOUNT=nnn parameter.

The volume count is to be used for the backup and archive tape data sets. It is a decimal number from 1 to 255. If the VOLCOUNT keyword is omitted or is null, the default volume count of 5 will be used.

IDRC

Replaced by the TAPEPARM statement COMP/NOCOMP parameter.

This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) on a tape device enabled for compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified or left blank, no TRTCH subparameter will be generated.

TMS

This parameter identifies the tape management system in use at your data center. It is required to allow CA MICS to test for EXPDT values that could cause CA MICS tapes to be accidentally scratched. Valid values are YES, the name of a tape management system, or NONE.

If you specify the TAPEPARM statement EXPDT=expdt or NOEXPDT parameter, or the old-style EXPDT statement, then you must also specify a value for TMS (for example, TMS NONE).

EXPDT

Replaced by the TAPEPARM statement EXPDT=expdt or NOEXPDT parameters.

Specify an expiration date definition that replaces the standard retention period (RETPD) definitions for the tape files used in backup, archive audit, archive weekly history, and archive monthly history. This facility is specifically directed for use by data centers that use a tape management system that requires specific EXPDT definitions. Note that for some tape management systems, certain EXPDT values will make the tape available for scratching immediately. Such values are not desirable for CA MICS tapes. This statement is an option, and if omitted, will result in the standard retention period (calculated in the manner described in the previous section) being used on the tape creation DDs. However, if you do specify an EXPDT value, you must also specify a value for the TMS parameter (for example, TMS NONE).

SYSTEMMODEL

Replaced by the TAPEPARM statement MODEL='modeldsn'/ MODEL=NOMODEL parameter.

The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted (or left blank), a CA MICS generation job will allocate a model GDG data set.

If NOMODEL is specified as the model data set name, the use of the model data set in the DCB specification will be bypassed.

DENSITY

This old-style statement is rarely used and is retained for compatibility with older tape devices.

The tape density to be used for the backup and archive tape data sets (3 for 1600 bpi or 4 for 6250 bpi).

The following parameters are used to specify the catalog that contains the high-level qualifier in the TSHAREDPREF. Either CVOLUNIT and CVOLSER must be specified and the VCAT parameters omitted (or left blank), or the VCAT parameters must be specified and the CVOL parameters omitted. In addition to building the GDG entries in the catalog specified by these parameters, the CA MICS generation jobs will allocate a model GDG data set on the volume containing the catalog if SYSTEMMODEL has not been specified.

CVOLUNIT

The generic unit name for the device on which the control volume catalog for the CA MICS data set resides. This parameter is mutually exclusive with VCATNAME.

CVOLSER

The volume serial of the volume on which the control volume catalog for the CA MICS data set resides. This parameter is mutually exclusive with VCATNAME.

VCATNAME

The name of the VSAM or ICF catalog in which the CA MICS data sets are to be cataloged, or SYSTEM. The SYSTEM keyword requires that the CA MICS tape prefix be an alias in the master catalog, or that you have update authority against the master catalog. SYSTEM is the default. This parameter is mutually exclusive with CVOLSER.

VCATSER

The volume serial number of the VSAM or ICF catalog in which the CA MICS data sets are to be cataloged. This parameter is mutually exclusive with CVOLSER. This parameter is required if VCATNAME has a value other than SYSTEM. If SYSTEM has been specified, then the value coded here will only be used in allocating the model data set.

VCATUNIT

The generic unit name of the VSAM or ICF catalog in which the CA MICS data sets are to be cataloged. This parameter is required if VCATNAME has a value other than SYSTEM. If SYSTEM has been specified then the value of this parameter will only be used in allocating the model data set. If no value is provided for VCATUNIT then DASDUNIT will be used instead.

VCATOWNER

The owner of the VSAM or ICF catalog in which the CA MICS data sets are to be cataloged. This parameter is optional and mutually exclusive with CVOLSER. This parameter is ignored if VCATNAME has a value of SYSTEM.

VCATPASSWORD

The password of the VSAM or ICF catalog in which the CA MICS data sets are to be cataloged. This password must provide update access, and is only required if needed. It is mutually exclusive with CVOLSER. This parameter is ignored if VCATNAME has a value of SYSTEM.

2.3.1.6.5 Database Complex Extended Job Card Definitions

Extended JOB statement parameters let you individually define each of the supported JOB statement options as listed below. Keep in mind that the JOB statement may be supplied at the beginning of the USERJCL PARMS member described in section 2.3.3.2.2, Job and JES Definitions. If you choose to have CA MICS construct JOB statements, specify the following parameters:

ACCT

Standard accounting field(s) required by the installation on JOB statements. A maximum of 53 characters may be specified. The value specified in this parameter is enclosed in parenthesis on the JOB statement. For example, if you specify the following definition:

```
ACCT 'MICS,ACCOUNT'
```

a JOB statement is generated with the following:

```
//jobname JOB (MICS,ACCOUNT),
```

CLASS

For JES2, the single character job class to be specified on the JOB statement. For JES3, the one to eight character job class to be specified on the //*MAIN statement.

GROUP

The RACF GROUP parameter to be specified on the JOB statement. The default is not to specify GROUP on the JOB statement.

JCLLIB

Specify YES to include, with the generated JOB statement, a JCLLIB statement for the CA MICS PROCLIB data set. The generated JCLLIB statement is of the form:

```
//MICS JCLLIB ORDER=(proclib_dsn)
```

Specify NO (the default) to omit the JCLLIB statement.

Warning: Do not use this option if you have a JCLLIB statement specified in USERJCL.

JOBNAME

The user may specify a single job name to be used for every job created by CA MICS (this is strongly discouraged) or may provide a job name prefix that is used by the JCLGEN process to build the job name.

The CA MICS JCL generation process will provide standard job names for each job that it creates (e.g., the daily CA MICS update stream's job name is DAILY). There are times, however, when the standard job names generated by CA MICS may not conform to an installation's job naming convention. We strongly recommend using the standard job names generated by CA MICS if at all possible. However, if it is necessary to have job names different from those generated by CA MICS, use the JOBNAME statement to specify your changes. If the CA MICS standard job names are satisfactory, either omit the JOBNAME statement from JCLDEFC or code it with no value following the JOBNAME keyword.

If you want to qualify all CA MICS job names with a special prefix, define a prefix of 1 to 7 characters for the value for the JOBNAME statement. The JCL generator uses the prefix value to concatenate the prefix with the standard CA MICS job name truncating the result to an eight character job name. For example, the JOBNAME value ABC would result in the standard CA MICS job name MONTHLY to now have the job name ABCMONTH. The new name is built using the job name prefix and the first n characters of the standard CA MICS job name up to the allowable eight character job name. The standard CA MICS JOBNAMEs may be found in sharedprefix.MICS.PARMS members beginning with JCL$I.

```
     JOBNAME ABC
```

If you want the same job name for all CA MICS jobs, specify a full eight-character name in quotes (trailing blanks are acceptable). Again, this job name option is strongly discouraged.

```
     JOBNAME 'ABCD    '
```

JOB7380

A few installations require job parameters to be entered in columns 73 through 80 of the first statement of the (possibly continued) JOB statement. This can be done with this parameter. Use of this parameter is not recommended unless required by your standards.

MSGCLASS

The job output message class to be specified on the JOB statement. The default MSGCLASS is A. Valid values are A-Z and 0-9.

MSGLEVEL

The job output message level to be specified on the JOB statement. The default MSGLEVEL is 2,1.

NOTIFY

The TSO user ID to be used for the JOB statement NOTIFY parameter, which sends a message to the specified user ID upon batch job completion. If you specify the value,

```
     &SYSUID
```

the JOB statement will be generated with NOTIFY=&SYSUID, which will direct the job completion message to the userid of the user who submitted the batch job.

PASSWORD

The RACF PASSWORD parameter to be used on the JOB statement. The default is to not specify a PASSWORD on the JOB statement.

PERFORM

The performance group to be specified on the JOB statement. The valid range of values is a numeric from 1 to 999. The default is to not specify a value for the performance group.

PROG

The 1- to 20-character programmer name required on the JOB statement by the installation. The value specified in this parameter is enclosed in quotes on the generated JOB statement.

PRTY

The job input priority to be specified on the JOB statement. For JES2 systems, the maximum priority that may be specified is 15. For JES3 systems, the maximum priority that may be specified is 14. The default is not to specify a PRTY value on the JOB statement.

```
REGION

    The job region size to be used on all JOB statements. The
    value can be specified in one of the following forms:

    nnnn  - region size in units of 1024 bytes (Kilobytes)
    nnnnK - region size in units of 1024 bytes (Kilobytes)
    nnnM  - region size in units of 1048576 bytes (Megabytes)

    Note - 0K and 0M are valid specifications.
         - leading zeros are not required.

    We recommend that you not specify this parameter unless
    it is required by installation standards.  All CA MICS
    PROCs have appropriate step region sizes specified.
USER

    The RACF USER parameter to be specified on the JOB
    statement.
```

CA MICS supports an option to hard-code a JOB statement in MICS.PARMS(USERJCL) instead of having CA MICS generate one from the parameters described above. All the parameters listed above may be used as symbolic variables whose values may be referenced, along with those of certain other CA MICS variables, in JCL that you supply in the USERJCL member in MICS.PARMS. For more information, see section 2.3.3.2.2, Job and JES Definitions (USERJCL).

2.3.1.6.6 Database Complex User Defined Symbols

This parameter allows the user to define a symbol and a value that will be substituted for the symbol by JCLGEN. This can be used in a user modification to PROTOLIB to provide user-defined symbolic elements. Several USERDEF symbols may be defined. The only restriction is that the symbol on each USERDEF statement must be different from those on all other symbols and different from all symbols currently used by JCLGEN. A list of CA MICS-defined symbols can be found in MICS.SOURCE(JCLPARM). Also, the value must end before column 73 of the record. For example,

```
   USERDEF DIGITS 0123456789
```

would define a symbol DIGITS. Anywhere this symbol appeared in a PROTOLIB member, JCLGEN would substitute the string 0123456789 in its place.

Note that any user symbols which appear in a PROTOLIB member for a complex-level job must be defined in sharedprefix.MICS.PARMS(JCLDEFC).

2.3.1.6.7 JCLGEN Parameter Sharing

JCLGEN Parameter Sharing allows commonly used JCLGEN parameters to be specified in sharedprefix.MICS.PARMS(JCLDEFC), where they then become default parameters for prefix.MICS.PARMS(JCLDEF) at the unit level.

Use of this facility can greatly reduce the amount of coding in the JCLDEF and JCLDEFC members.

JCLGEN parameters eligible for sharing can be organized into the following groups:

```
    SAS Data Set Definitions:
    ------------------------
    SASNAME     - SAS program name
    SASCONFIGx  - SAS configuration member
    SASAUTOS    - SAS auto include library
    SASMSG      - SAS message data set
    SASLIB      - SAS executable library
    SASVER      - SAS version
    SASHELP     - SAS help library

    ISPF Data Set Definitions:
    -------------------------
    ISPTLIBx    - ISPF table data set
    ISPLLIBx    - ISPF load library
    ISPMLIBx    - ISPF message data set
    ISPPLIBx    - ISPF panel data set
    ISPSLIBx    - ISPF skeleton data set

    Miscellaneous Definitions:
    -------------------------
    ASSEMBLER   - IBM Assembler level
    COLIBSPACE  - Size of the temporary MCOLIB
    COMPLEXPARMS- Allow/disallow JCLGEN parameter sharing
    DIRSTATS    - ISPF stats for generated members
    HFSROOT     - Name of the CA MICS HFS path
    JCLLIB      - Include JCLLIB statement with job card
    MICSDMDELIM - Delimiter character for MICSDM/MICSDU procs
    OBJACCESS   - Access mode for MCOLIB and MUOLIB
    OBJWAIT     - Wait time for MCOLIB and MUOLIB allocation
    PROCLIB     - Location of CA MICS PROCs
    SORTLIB     - SORT utility library
    SORTUTILITY - Installation sort utility
    SORTWKNO    - Number of sort work data set allocations
    STEPCPULIMIT- Limit CPU time for any step to 60 minutes
    STEPLIBx    - User-defined STEPLIBs
    SYSTEMMODEL - Name of model data for GDG creation
    USEROBJLIB  - User object definition library
```

JCLGEN Parameter Sharing is always active, but it must be enabled by specifying the following in sharedprefix.MICS.PARMS(JCLDEFC) or prefix.MICS.PARMS(JCLDEF):

```
COMPLEXPARMS YES
```

When specified in JCLDEFC, the value of COMPLEXPARMS is the default for all units. If specified in JCLDEF, COMPLEXPARMS overrides the COMPLEXPARMS action specified in JCLDEFC.

Section 5.5 provides an optional checklist that shows how to activate JCLGEN parameter sharing for all units in the CA MICS complex.

Notes:

- For the ISPF data set and the STEPLIB definitions, specifying the first definition (that is, ISPTLIB1) will automatically blank out the second definition (that is, ISPTLIB2). Therefore, you must always specify these definitions in numeric order.

- COMPLEXPARMS must be the very first statement in JCLDEFC and JCLDEF. If it is not, an error message is generated.

2.3.1.7 Database Complex Account Code Parameters

Some CA MICS products have account code fields as part of the key by which their files are organized. This section describes how to define an account code structure (typically cccACCT) and derive account code values (typically cccACRT). As complex-level parameters, these definitions hold for every unit database in the database complex. Specifics about account codes for data integration applications are documented in each data integration guide.

When defining the account code structure, it is important for the CA MICS System Administrator to understand the installation's accounting conventions. If these conventions are not well documented, consider spending time to document the system in use.

Note: If you are unfamiliar with CA MICS file keys and summarization, you might want to review Section 2.3.2.1 before continuing.

DEFINING AN ACCOUNT CODE STRUCTURE TO CA MICS (cccACCT)

Once you have documented the accounting structure in use, you must decide how you want it expressed in the CA MICS Database. You must settle on the number of account code fields (CA MICS supports from 1 to 9) and what will be placed in each. Because the account code fields are part of the key for each file in which they exist, there will be at least one record for each unique combination of values (and perhaps more than one because there are other fields in the keys). The trade-off you must make is between keeping a fine level of detail via the account code fields and therefore having a large database, and keeping a small level of detail and perhaps not meeting your reporting or analysis requirements.

Once you have completed the account code worksheets in the data integration guides, code the contents of the account code structure parameters as follows:

- o Statement format:

```
level length 'descriptive title'
or
level mask length 'descriptive title'
```

- o Code a separate statement for each account code field.

- o Blank statements are permitted. Comments are coded by beginning the statement with an asterisk (*).

- o Account levels are provided in order, starting with "1".

- o Up to nine levels are permitted; gaps between the numbers are NOT permitted.

- o The format of a statement is free-form but positional, meaning that the keywords on the statement must appear in the order shown but are not restricted to being keyed in specific columns. All keywords except mask are required.

The meaning of each keyword on the account code statement is:

- o level - the account level

  The level of importance of each element. Level 1 is the most important (major field) and the highest level number is the least important (minor field). From 1 to 9 levels can be specified. An account code data element will eventually be built for each account code level that is defined. For instance, if the batch accounting structure has five levels, then the Batch information area files in the CA MICS Database would have the following data elements:

  ```
  ACCTNO1 ACCTNO2 ACCTNO3 ACCTNO4 ACCTNO5
  ```

  CA MICS keeps all files in which these variables appear sorted with ACCTNO1 being the highest key and ACCTNO9 the lowest.

- o length - the field length

  The length of the specified element. The length can range from 1 to 30. Make the field long enough to hold meaningful values, but remember that the account field(s) appear in every database record for the associated product's file(s). Thus, the length of account fields can impact total database size.

- o 'descriptive title'

  The title describing the account code field. The length of the title is limited to 40 characters; it must be enclosed in single quote marks.

- mask - the optional timespan mask

  Account code levels can be deactivated in the DAYS, WEEKS, MONTHS, and YEARS timespans. For example, the Detail timespan can have account code levels 1, 2, and 3 while the Months timespan has only level 1. This capability is a powerful tool for ensuring adequate data granularity while controlling database size. You can define account codes to carry very detailed identification information at the DETAIL and DAYS timespans, while dropping account codes of lesser value in MONTHS-level summary data.

  Setting account code levels by timespan is accomplished by specifying the timespan mask keyword T(......), as follows:

  ```
    level T(......) length 'descriptive title'
  ```

  where each "." position in the timespan mask keyword represents a timespan in the order DETAIL, DAYS, WEEKS, MONTHS, YEARS, and TABLES. For each timespan, a "Y" indicates that the account level should be active and an "N" indicates that the account level should be inactive.

  If this keyword is not coded, the assumed value is T(YYYYYY) for each level.

  If you code this keyword, the following rules apply:

  - Detail must always be "Y". Account codes cannot be deactivated in this timespan.

  - If Days is "N", then WEEKS, MONTHS, and YEARS must also be "N". If the account code level is inactive in the DAYS timespan, it cannot be active in higher timespans.

  - If Months is "N", then Years must be "N". If the account code level is inactive in the Months timespan, it cannot be active in the Years timespan.

DEFINING ACCOUNT CODE DERIVATIONS TO CA MICS (cccACRT)

To assign values to the account code fields, you must define a corresponding account code derivation exit. Account code derivation exits are written in SAS language and are stored in the sharedprefix.MICS.PARMS library in members named cccACRT.

Exit code is included directly into the processing of the CA MICS update jobs, so it is very important that such routines be coded correctly. Testing and maintaining the accuracy of these routines is your responsibility.

Refer to the data integration guides for examples of cccACRT exits.

2.3.1.8 Complex Option Definition (CPLXDEF)

This section explains how to specify the complex-level database tailoring options and environmental parameters. These parameters apply to all components installed in the CA MICS database complex and are specified by parameters in sharedprefix.MICS.PARMS(CPLXDEF).

Use the CPLXDEF member to define the following:

o Global database tailoring options: the essential data element option, the derived data element option, and the four-digit year data element option.

o Printed report and SAS log title text.

o Default values for unit database years timespan option, weeks timespan option, the first day of the week, the first week of the year, and fiscal calendar.

sharedprefix.MICS.PARMS(CPLXDEF) contains two required statements:

o ESSENTIAL
o DERIVED

It contains seven optional statements:

o REPORT
o LOG
o YEAR4
o YEARS TIMESPAN
o WEEKS TIMESPAN
o WEEKSTART
o 13MONTHYEAR

Each of these statements is described in detail below.

THE ESSENTIAL STATEMENT

The ESSENTIAL statement is used to select the default essential data element option. The format of the ESSENTIAL statement is:

```
ESSENTIAL   ALL/ONLY/INITIAL
```

where only one value (ALL, ONLY, or INITIAL) is specified. The effect of coding each of these values is described next.

ALL All data elements defined in the component's cccGENIN member on sharedprefix.MICS.GENLIB are stored on the disk or are available through the derived element macros.

ONLY Only the data elements essential for file key structures, component standard reports, and CA MICS analytic component feeds are stored on disk. All other data elements are dropped from CA MICS processing and are not available without reprocessing raw DETAIL data. When ONLY is specified, the DERIVED option determines whether essential computed elements are stored on disk or are available through the derived element macros.

INITIAL Like the ONLY option, only essential data elements are stored on disk. When INITIAL is specified, the DERIVED statement is ignored and the DERIVED option is automatically forced to COMP. This means the essential computed elements are available only through the derived element macros. Additionally, the WEEKS, MONTHS, and YEARS timespans of all user and application unit identified files are disabled.

ESSENTIAL element options take effect after the CPLXDEF member is changed, a complex-level JCLGEN job (for example, JCLGEN0, JCLGENUC) or CPLXGEN job has been run, and a component generation job (for example, RMFCGEN) has been run. Running such generations to change the content of the database usually requires JCL generations and other installation modifications for each database unit in which regenerated components have been installed.

The distributed value for the ESSENTIAL option in CPLXDEF is ONLY. This uses the least DASD and CPU resources to support a meaningful subset of the data available from each component.

When changing from INITIAL to ONLY or ALL, several components will have previously inactive files activated in timespans (WEEKS, MONTHS or YEARS). To change from INITIAL to any other ESSENTIAL option, it is necessary to reinstall the component in all database units in which the component has been installed. Use the checklist in Section 3.8.2, Adding a Component to an Existing Database Unit, to reinstall the necessary components.

THE DERIVED STATEMENT

The DERIVED statement is used to select the default derived data element option. The format of the DERIVED statement is:

```
DERIVED   DISK/COMP
```

where only one value (DISK or COMP) is specified. The effect of coding each of these values is described separately below.

DISK Computed data elements, defined as type 'C' or 'CN' data elements in component generation control members named sharedprefix.MICS.GENLIB(cccGENIN), are stored on disk (auxiliary storage) in the SAS files for the CA MICS database. When a program reads the SAS file on which such data elements reside, the data elements are immediately available for use.

COMP Computed data elements, defined as type 'C' or 'CN' data elements in component generation control members named sharedprefix.MICS.GENLIB(cccGENIN), are not stored on disk, but are computed for each use (such as in a user-written inquiry program). CA MICS macros are available for performing the calculations necessary to compute the derived data elements.

DERIVED element options take effect after the CPLXDEF member is changed and a complex-level JCLGEN (for example, JCLGEN0 or JCLGENUC) or CPLXGEN job has been run.

Note that setting the ESSENTIAL INITIAL option means that the DERIVED statement is automatically forced to COMP. Only essential computed elements are available through the derived element macros. A DERIVED statement must always be coded, even in the case where it is ignored.

Note that both the ESSENTIAL and DERIVED settings can be overridden at the component and file level using the COMPOPT and FOPT statements available in each components CCCGENIN member. See the System Modification Guide section 4.2.4 for more information.

The REPORT statement defines the CA MICS complex name to use in the title for complex-level reports. It is an optional statement. If used, the name text must be separated from the keyword by one or more blanks. The maximum length of the name string is 40 characters. Do not specify quotes around the name string. For example, the following statement results in all complex-level reports having the title, CA MICS Production Complex:

```
REPORT  CA MICS Production Complex
```

THE LOG STATEMENT

The LOG statement defines the title to write at the beginning of the SAS log for complex-level jobs. It is an optional statement. If used, the title text must be separated from the keyword by one or more blanks. The maximum length of the text string is 40 characters. Do not specify quotes around the text string. For example, the following statement results in all complex-level SAS logs having the title, CA MICS Production Complex:

```
LOG  CA MICS Production Complex
```

YEAR4 STATEMENT

The YEAR4 statement is used to generate macros that enable the display/processing of data beyond the calendar year 2000 by adding a four-digit variable, YEAR4, to all CA MICS files.

The YEAR4 data element coexists with the two-digit YEAR data element if YEAR4 is activated. You can use the YEAR4 data element instead of YEAR in your user-written reports for code clarity. For example, you can select data between 1998 and 2008 as follows:

```
IF 1998 < YEAR4 < 2008;
```

The YEAR4 data element is shipped inactive and is not defined as a normal CA MICS data element in sharedprefix.MICS.GENLIB. As a result, you cannot use the standard method of file tailoring to turn YEAR4 on or off. You must activate or deactivate this data element using the YEAR4 statement listed next.

To activate the YEAR4 data element, specify the following:

```
YEAR4
```

To deactivate the YEAR4 data element, omit the YEAR4 statement.

Since the YEAR4 statement affects the operation of the entire CA MICS complex, the YEAR4 data element will reside either in every CA MICS file or no file at all.

After you modify CPLXDEF, you must run the corresponding CPLXGEN job to generate the necessary keep, length, format, and label specifications that are used by CA MICS operational jobs.

Note: The CPLXDEF YEAR4/NOYEAR4 option activates or deactivates the YEAR4 data element in all CA MICS files; however, until you run the cccCGEN job for each data integration product (including BASCGEN) installed in the CA MICS complex, YEAR4:

- Does not appear in DIA guide file element lists or in their data dictionaries.
- Is not available to MICF inquiries or to the Document Access Dictionary Browse option.

Using this method allows you to implement your YEAR4 specification in the database as quickly as possible, while delaying cccCGEN processing until a convenient time.

Note: The following parameters provide global default values for all unit databases in the CA MICS complex. When you specify unit database parameters, you have the option to use the global default or you can override the complex-level parameter values and specify unique definitions for a single unit database.

In either case, parameter specifications are implemented by running the individual unit database BASPGEN jobs. Changes to CPLXDEF are not effective in an individual unit database until you execute the BASPGEN job for that unit database.

YEARS TIMESPAN ACTIVE/INACTIVE:

The files in the YEARS timespan are updated as part of the MONTHLY processing job if the YEARS timespan is active at your site.

The default unit database parameters are distributed as:

YEARS TIMESPAN INACTIVE

This saves both DASD space and processing time by not updating the YEARS files.

If the CA MICS administrator decides that the YEARS timespan should be used to update the YEARS file, either delete the YEARS TIMESPAN parameter line or specify ACTIVE for this option. The ACTIVE value will be effective for any any database that does not override the complex-level default by explicitly specifying the YEARS TIMESPAN option.

Note that CA MICS still maintains a prefix.MICS.YEARS data set when the YEARS timespan is inactive. The files in this data set contain zero SAS observations and are not updated during MONTHLY processing. In this mode, the YEARS timespan requires only a minimal amount of DASD space.

If the YEARS timespan is changed from ACTIVE to INACTIVE after data is processed by CA MICS, any SAS observations in the year-to-date files are dropped during the MONTHLY process following the parameter change. Data in files other than the year-to-date files are not affected by changing the parameter to INACTIVE from ACTIVE.

WEEKS TIMESPAN ACTIVE/INACTIVE:

The files in the WEEKS timespan are updated as part of the DAILY processing job if the WEEKS timespan is active at your site.

By default, the unit database parameters are distributed without the WEEKS TIMESPAN statement.

If the CA MICS administrator decides that the WEEKS timespan is not used for reporting purposes and is therefore unneeded, it can be deactivated for the entire complex by inserting the WEEKS TIMESPAN parameter option as shown below:

```
WEEKS TIMESPAN INACTIVE
```

The INACTIVE parameter will be effective for any unit database that does not override the complex-level default by explicitly specifying the WEEKS TIMESPAN option.

Note that CA MICS still maintains a prefix.MICS.WEEKS data set when the WEEKS timespan is inactive. The files in this data set contain zero SAS observations and are not updated during DAILY processing. In this mode, the WEEKS timespan requires only a minimal amount of DASD space.

If the WEEKS timespan is changed from ACTIVE to INACTIVE after data is processed by CA MICS, any SAS observations in the week-to-date files are dropped during the DAILY process following the parameter change. Data in files other than the week-to-date files are not affected by changing the parameter to INACTIVE from ACTIVE.

WEEKSTART Option (SUN MON TUE THU FRI SAT):

This option specifies the first day of the week, whose value can be changed from SUN to one of the following: MON, TUE, WED, THU, FRI, or SAT. Only use the WEEKSTART option if you intend to change the default value (SUN). WEEKSTART is used to compute the value of WEEK from a given date.

If you choose to specify the WEEKSTART parameter, you are encouraged to make your specification here, at the complex-level, and then allow the parameter to default in each unit database. By doing so, your data will be maintained in a consistent manner across all units.

Thirteen Month Fiscal Year Option (13MONTHYEAR):

The standard definitions used by CA MICS for DAY, WEEK, MONTH, and YEAR are based on a calendar year beginning January 1 and having 12 months, with the number of days per month varying from 28 to 31 depending on the month. Data is summarized in the CA MICS database according to these standard definitions. If your organization operates on the standard 12-month year and follows the common North American understanding of the first week of a new year, DO NOT code this option.

The default for this option is NO. Only use this option if you intend to override the default. Before using it, we strongly recommend that you review Section 4.7.2 in the System Modification Guide for information about adjusting algorithms and year-end boundary considerations.

13MONTHYEAR has been provided to offer non-standard calendar-related options. In particular, it enables CA MICS to summarize and store data for an organization that operates on a 13-month fiscal year. It also allows a year to have up to 380 days.

Note: The CA MICS Accounting and Chargeback component includes facilities for defining a unique accounting calendar separate from the global CA MICS calendar definitions. This capability is designed to help you meet your requirements for chargeback and accounting, relative to your company's fiscal calendar, while continuing to use the standard 12-month calendar for the majority of your CA MICS information. For more information on altering the global CA MICS calendar, see the CA MICS Accounting and Chargeback guides.

If you choose to define a fiscal calendar for your CA MICS environment, you are encouraged to make your specification here, at the complex-level, and then allow the parameter to default in each unit database. In this way, your data will be maintained in a consistent manner across all units.

This statement can be formatted in any of the following ways:

```
13MONTHYEAR NO
13MONTHYEAR NO  #DWMY=memname
13MONTHYEAR YES ddmonyy ddmonyy ...  ddmonyy
13MONTHYEAR YES #DWMY=memname ddmonyy ddmonyy ...
```

Note: Only one 13MONTHYEAR control statement is allowed. Subsequent 13MONTHYEAR statements replace prior 13MONTHYEAR statements.

where:

13MONTHYEAR is the option keyword.

NO is specified if you use a standard 12-month year beginning on January 1 and ending on December 31. The default value shipped with CA MICS is NO. It causes the SOURCE member #DWMY12 to be used.

YES is specified only if your organization operates on a 13-month year.

memname is a member that contains date macros. It MUST begin with the characters #DWMY. For example: #DWMYUSR.

The ddmonyy keyword defines the year's starting date. Valid values are:

```
dd   01 to 31, a day of the month
mon  JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC, a
     three-letter month abbreviation
yy   two digits, representing the year
```

Dates must be in ascending order and a year cannot contain more than 380 days. You cannot code two dates using the same year value (that is, 01JAN08 and 28DEC08). You can specify a maximum of eight dates (that is, the 13MONTHYEAR statement can define no more than eight years).

Here is an example:

```
13MONTHYEAR YES 01OCT07 30SEP08
```

This is interpreted as fiscal year 2007 begins on October 1, 2007 and ends on September 29, 2008. Fiscal 2008 begins on September 30, 2008 and ends on September 29, 2009.

The CPLXDEF member is distributed with the following options specified:

```
ESSENTIAL  ONLY
DERIVED    COMP
LOG
REPORT
WEEKS TIMESPAN ACTIVE
YEARS TIMESPAN ACTIVE
WEEKSTART SUN
13MONTHYEAR NO
```

This means the following:

- Only essential data elements are kept in the database.

- Essential computed data elements are calculated using derived element macros.

- YEAR4 is NOT kept in the database.

- The unit database definition defaults are

  - The YEARS timespan is active.

  - The WEEKS timespan is active.

  - Weeks start on Sunday.

  - CA MICS operates under the standard 12-month Gregorian calendar.

Figure 2-8 provides a worksheet for collecting the specifications used in the CPLXDEF member.

```
+--------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET: Complex Option Definitions           |
|                                                                          |
| PARMS Library Member is CPLXDEF                                          |
| Reference Section: 2.3.1.8                                               |
+--------------------------------------------------------------------------+
|                                                                          |
|                                                                          |
| ESSENTIAL OPTION:                                                        |
| ESSENTIAL _____________ (ALL, ONLY, or INITIAL)                          |
|                                                                          |
|                                                                          |
| DERIVED OPTION:                                                          |
| DERIVED   _____________ (DISK or COMP)                                   |
|                                                                          |
|                                                                          |
| Title for complex-level report (optional):                               |
| REPORT    _____________                                                  |
|                                                                          |
|                                                                          |
| Title for complex-level log (optional):                                  |
| LOG       _____________                                                  |
|                                                                          |
| Add 4-digit year variable to all CA MICS files (optional):               |
| YEAR4     _____________                                                  |
|                                                                          |
| YEARS TIMESPAN OPTION:                                                   |
| YEARS TIMESPAN ________ (ACTIVE or INACTIVE)                             |
|                                                                          |
| WEEKS TIMESPAN OPTION:                                                   |
| WEEKS TIMESPAN ________ (ACTIVE or INACTIVE)                             |
|                                                                          |
|                                                                          |
| WEEKSTART OPTION:  (default is SUN)                                      |
| WEEKSTART ________ (MON TUE WED THU FRI SAT)                             |
|                                                                          |
|                                                                          |
| 13MONTHYEAR OPTION:  (default is NO)                                     |
| 13MONTHYEAR NO  #DWMY=________                                           |
|             YES ________ ________ ________                               |
|                                                                          |
+--------------------------------------------------------------------------+
| ....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+--------------------------------------------------------------------------+
```

Figure 2-8. Complex Option Definitions Worksheet

2.3.1.9 Complex-Level Computing System Parm (CPLXSID)

Data in the CA MICS database are associated with and summarized according to the computing system that produced the raw data used to create the observations in the CA MICS database.

Computing systems are identified by their SMF system identifiers (SYSIDs), which are generally unique to specific systems. For example, SYSID=P390 might be the Model 3090 Production system. The SYSID parameter lets you map SMF system identifiers (CA MICS data element ORGSYSID), which will probably change over time, to the business work that takes place on your systems (CA MICS data element SYSID).

You code information about the systems used at your installation in sharedprefix.MICS.PARMS(CPLXSID) and/or prefix.MICS.PARMS(SYSID).

- o sharedprefix.MICS.PARMS(CPLXSID) provides global, complex-level control parameter and system definition defaults. Use CPLXSID to define systems common to multiple database units.

- o prefix.MICS.PARMS(SYSID), the unit-level parameters, override complex-level default parameter values from sharedprefix.MICS.PARMS(CPLXSID). Use the SYSID member for system definitions specific to a single unit database.

Use the worksheet in Figure 2-9A to collect the data needed to code this parameter member.

There are two keyword parameters:

- UNDEFINEDSYSID ABORT/EXCLUDE
- SYSIDCHECK ABORT/AUTODEFINE

For each system, code one line containing seven parameter values:

- SMF SYSID (CA MICS data element ORGSYSID)
- logical SYSID (CA MICS data element SYSID)
- CPU model
- system control program name
- RMF conversion factor
- adjustment factor
- system name

Lines beginning with '*' are comments and are ignored. Blank statements are allowed.

Both the keyword and SYSID definition parameters are described in greater detail later in this section. You can begin your customization by modifying the sample CPLXSID member in sharedprefix.MICS.PARMS(CPLXSID).

Example System Definition Entry

```
P390 PROD '3090-200' OS/390 . 14.6 'TSO PRODUCTION SYSTEM'
```

This sample entry specifies an SMF system of P390, which is translated to the logical system PROD. The processor is a 3090-200 running OS/390, the RMF CPU conversion factor is left as a missing value because the RMF data provides this value, the Adjustment Factor is 14.6 (per internal processor), and the reporting name to be displayed for this system is TSO PRODUCTION SYSTEM. (The values used in this example are for illustration purposes only. DO NOT assume that they are correct for your 3090-200 processor).

Note: The Adjustment Factor may have a missing value of PERIOD, '.' not BLANK. BLANK values will result with error messages.

Considerations for Coding CPLXSID

By default, CA MICS terminates with a user abend if input data comes from an SMF system that is not defined in the unit-level parameter library or provided through the complex-level CPLXSID parameter. You modify this default behavior through control parameters

To change complex-level CPLXSID parameter information after CA MICS is in use, first change the CPLXSID member in sharedprefix.MICS.PARMS and then execute the job CPLXGEN in the sharedprefix.MICS.CNTL library.

Note: CPLXSID parameter specifications are ignored unless prefix.MICS.PARMS(SYSID) specifies:

```
COMPLEXSYSIDPARM USE
```

Review prefix.MICS.PARMS(SYSID) in each unit database. Add COMPLEXSYSIDPARM USE to the SYSID parameter member and run BASPGEN in each unit database where you want

```
          CPLXSID specifications to apply.

     Considerations for Analyzers
     ----------------------------

     See the individual product guides for specific considerations
     related to specific CA MICS data integration applications.
     See the CA MICS System Modification Guide for considerations
     related to MTI components and Field Developed Applications.
```

```
+---------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET:  Complex-Level System Identifiers    |
|                                                                           |
| PARMS Library Member is CPLXSID                                           |
| Reference Sections:  2.3.1.9                                              |
+---------------------------------------------------------------------------+
|                                                                           |
| UNDEFINEDSYSID  ________________________ (ABORT or EXCLUDE)               |
| SYSIDCHECK  ________________________ (ABORT or AUTODEFINE)                |
|                                                                           |
|                                                                           |
| SMF Logical CPU  RMF Adjust.                                              |
|SYSID SYSID Model SCP Factor Factor 'System Name'                          |
+----------------------------------------------------------------------     |
| P390 PROD '3090-200' OS/390 .  14.6 'SAMPLE TSO PRODUCTION SYSTEM'        |
|                                                                           |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                           |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                           |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                           |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                           |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                           |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                           |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                           |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                           |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                           |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                           |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                           |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                           |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                           |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                           |
|                                                                           |
+---------------------------------------------------------------------------+
| ....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+---------------------------------------------------------------------------+
```

Figure 2-9A. Complex-Level System Description Worksheet

Control Parameters:

UNDEFINEDSYSID

Defaults to ABORT if keyword is not specified.

Specify ABORT to terminate database update processing with a U310 abend if data comes from an SMF system that is not defined either at the unit-level in prefix.MICS.PARMS(SYSID), or at the complex-level in sharedprefix.MICS.PARMS(CPLXSID).

Specify EXCLUDE to automatically drop data that comes from an SMF system that is not defined.

- This option not only keeps data from unexpected systems out of your CA MICS database, but also avoids parameter changes, CPLXGEN or BASPGEN executions, and database update job restarts to address data from unexpected sources such as test systems.

- Use the EXCLUDE option for "permanent" changes.

- To address temporary or one-time situations, specify

  EXCLUDEORGSYSIDS UNDEFINEDSYSID

  in prefix.MICS.PARMS(EXECDEF) to override an UNDEFINEDSYSID ABORT specification.

SYSIDCHECK

Defaults to ABORT if keyword is not specified.

Specify ABORT to terminate cccPGEN processing if ORGSYSID or SYSID values appear in a CA MICS product parameter member (e.g., CICOPS, SMFOPS, PRFGP, etc.) but are not defined either at the unit-level in prefix.MICS.PARMS(SYSID), or at the complex-level in sharedprefix.MICS.PARMS(CPLXSID).

Specify AUTODEFINE to have BASPGEN or ALLPGEN automatically create new prefix.MICS.PARMS(SYSID) entries for previously undefined ORGSYSID and/or SYSID values found in CA MICS product parameter members (e.g., CICOPS, SMFOPS, PRFGP, etc.). With this option,

- BASPGEN, ALLPGEN, and product cccPGEN jobs (e.g., SMFPGEN, CICPGEN, etc.) continue to issue MICSLOG warning messages when ORGSYSID or SYSID values appear in CA MICS product parameters without corresponding entries in prefix.MICS.PARMS(SYSID) or sharedprefix.MICS.PARMS(CPLXSID).

- Product cccPGEN jobs (e.g., SMFPGEN, CICPGEN, etc.) issue MICSLOG warning messages instructing you to run BASPGEN in order to create required entries in prefix.MICS.PARMS(SYSID).

- BASPGEN and ALLPGEN save the original, unmodified SYSID member as prefix.MICS.PARMS(SYSIDnnn), where nnn is a generated, unique identifier, and then append new entries to prefix.MICS.PARMS(SYSID). You can delete the SYSIDnnn member at any time as it is created only to give you a "before image" of automatic changes.

- Autodefined SYSID entries are created as follows.

    - SMF SYSID (ORGSYSID) is set to the new ORGSYSID/SYSID value from the CA MICS product parameter member.

    - Logical SYSID (SYSID) is set to match ORGSYSID.

    - CPU model is set to UNKNOWN.

    - System control program name is set to UNKNOWN.

    - RMF conversion factor is set to missing value ('.').

    - Adjustment factor is set to missing value ('.').

    - System name is set to the string "SYSTEM " followed by the ORGSYSID value.

- You can correct/revise autodefined SYSID entries by editing the new prefix.MICS.PARMS(SYSID) and running BASPGEN.

SMF SYSID: the Original or Input System Identifier

First, identify all of the SYSIDs currently being used and that you expect to appear in each unit database. (These are contained in SMF records and are referred to as original system IDs.) Then ensure that the SYSIDs are unique.

If the SYSIDs are not unique, work with your site's system programmers to assign unique SYSIDs to each system.

Logical SYSID: the Logical System Identifier

The logical system identifier is the key under which almost all measurement data in the CA MICS database is stored. This value, which is kept in the CA MICS common data element SYSID, is the highest key in almost all CA MICS files.

You can specify that the original and logical system identifiers be identical. If, however, you have a current system identifier that may change, we urge you to implement a unique logical system identifier.

*** Warning ***

> You cannot assign two concurrently used SYSIDs to the same logical SYSID. If a logical SYSID is specified more than once, the CA MICS generation process will flag the condition as an error. Contact CA Technical Support for advice on how to handle a situation where both the original machine and its successor run side-by-side for a transition period.

CPU Model

Some data sources do not identify the CPU model on which the data was produced. To ensure that this information is available in your CA MICS database, you provide a CPU model identifier in the SYSID parameter.

Code this parameter as 1-20 characters in quotation marks (e.g., '3090-400J') with no blanks inside the quotes.

While CA MICS does not process data based on this value, it is available to a SAS program written to run in the CA MICS environment via a SAS macro.

SCP: the System Control Program Level

The SCP keyword provides the level of MVS currently being run on the system described by this parameter statement.

Code this parameter as 1-10 characters with no imbedded blanks (e.g., OS/390).

While CA MICS does not process data based on this value, it is available to a SAS program written to run in the CA MICS environment via a SAS macro.

RMF Factor: the RMF Service-Units-to-CPU-Time Conversion
            Factor (Optional)

The RMF factor keyword asks for the value that CA MICS should use to convert TCB and SRB service units, reported by RMF, into CPU times.

The releases of RMF supported by CA MICS provide this value, so code a period (translated as a SAS missing value) for this keyword. Your "missing value" will be overridden by the value found in the Workload Activity Record.

Note: Any value other than '.' will be ignored during processing.

Adjustment Factor (Optional)

The CA MICS database lets you retain information about CPU consumption in units of time. This information is retained in elements whose names take the form fffxxxTM, where fff is the file identifier, xxx is file-specific, and TM stands for time.

Over time, the validity of time as a measure of CPU use becomes questionable due to hardware environment changes. (For example, one second of ES/9000 time is not the same as one second of 3033 time.)

To compensate for this difference, CA MICS lets you retain the number of instructions executed. This information is retained in elements whose names take the form fffxxxNI, where fff is the file identifier, xxx is file-specific, and NI stands for number of instructions.

The number of instructions executed is computed from the measured CPU time and an adjustment factor you choose, such as a MIPS rating for an MVS system. The adjustment factor is optional. If you code a zero, the fffxxxNI elements will have a zero value.

In choosing an adjustment factor, consider these sources:

- Conduct a hardware monitor study to determine your processor's rating. This method yields the most reliable results, but is costly to implement.

- Ask the hardware manufacturer for a normalized rating, such as a MIPS rate. (If your site operates in a multiprocessor environment, ask for the rate for a single processor.)

- Contact a hardware monitor manufacturer and request normalized ratings.

- Ask colleagues if they have information they are willing to share.

- Review the RMF Service-Units-to-CPU-Time Conversion Factor MVS used to normalize CPU times across different processor models.

System Name: the Descriptive Name of the System

With this keyword you supply a 1-30 character name for the system identified on the parameter statement. This keyword must be enclosed in single quotes.

The value you specify is not carried in the database, but is available to user-written routines by means of a SAS format in prefix.MICS.MUOLIB named $SYSIDN. It is created when the CA MICS unit is generated. For example, a SAS data step could include the following statement:

```
  SYSTEMNM = PUT(LOGSYSID,$SYSIDN.);
```

where LOGSYSID is a 4 digit character variable containing a Logical System Identifier as described above (PROD is used in the worksheet).

2.3.1.10 Complex-Level Time Zone Definitions (CPLXZONE)

Defining time zones is important because summarizing data by zone lets you evaluate availability and load during different service periods, both in the short-term and the long-term. For example, in evaluating online service, attention is focused on the activity of the peak load zones, while little, if any, attention is directed at the non-peak zones.

You code information about the operating schedule or service periods (time zones) used at your data center in sharedprefix.MICS.PARMS(CPLXZONE) and/or prefix.MICS.PARMS(ZONE).

- sharedprefix.MICS.PARMS(CPLXZONE) provides a global, complex-level time zone definition.

- prefix.MICS.PARMS(ZONE), the unit-level parameters, override the complex-level definition from sharedprefix.MICS.PARMS(CPLXZONE).

It is your choice to use the complex-level ZONE defintion or to fully specify the ZONE definition at the unit-level. As a general rule, you should use the complex-level parameter, sharedprefix.MICS.PARMS(CPLXZONE), to establish a consistent time zone definition for all units. In this way, you can maintain your holiday and special shift definitions in a single parameter member, and implement changes with a single CA MICS parameter generation. In using this approach, the individual unit database prefix.MICS.PARMS(ZONE) members will contain the single parameter line,

```
       COMPLEXZONEPARM USE
```

Use the unit-level time ZONE parameters to override the complex-level time zone definitions to address special requirements for different time zone definitions for different units of work. For example one database unit may support a remote data center, or you may have unique shift differential requirements for online transaction workloads separate from standard TSO and batch workloads. In these situations, you isolate the special workloads to a unique unit database and fully specify a unique time zone definition in prefix.MICS.PARMS(ZONE) for that unit database. Note that the unit-level ZONE definition completely replaces and overrides the complex-level ZONE definition for this unit database. You control this choice with the COMPLEXZONEPARM USE/IGNORE parameter in prefix.MICS.PARMS(ZONE). See Section 2.3.2.3 for more information on the unit-level ZONE definition.

The remainder of this section describes how to specify your global complex-level time zone definition in sharedprefix.MICS.PARMS(CPLXZONE). After completing the time zone definition, execute sharedprefix.MICS.CNTL(CPLXGEN) to implement your specifications.

In sharedprefix.MICS.PARMS(CPLXZONE), you are asked to define up to nine service periods (time zones) for your data center. Time zones are defined once and apply to all Information Areas within a database. It is not possible to have a time zone for TSO and another time zone for RMF in the same database. These service periods, which are also sometimes referred to as "shifts," are applied over all batch and online work examined by CA MICS.

The time zone, which is identified by the data element ZONE in the database, is used for summarizing data in the WEEKS, MONTHS, and YEARS timespans (it becomes part of the key of the files, replacing DAY, HOUR, and End Time-Stamp). The ZONE data element also improves the user's ability to select or extract data easily from files in the DETAIL and DAYS timespans where it exists, but is not part of the key.

The number of zones defined impacts the projected space requirements of the database since the data element ZONE is used for summarization (is part of the key of the files) in the WEEKS, MONTHS, and YEARS timespans.   The more zones, the larger the online DASD space requirements for the database. We recommend that the number of standard zones defined be four or fewer, with one zone for holidays.

CA MICS supports two kinds of ZONEs:

Standard zones are periods that recur in each week. For example, a standard zone can be defined as "9 a.m. to 5 p.m. on Mondays through Fridays, except for holidays".

Holiday zones are periods that are precisely identified by date and time. For example, "9 a.m. to 5 p.m. on New Year's Day, January 1, 2003". In addition, some data centers have used holiday zones to identify "super peak" periods such as end-of-month closings.

In determining the ZONE into which a specific date/time combination found in a measurement record falls, CA MICS first checks the holiday zones for an exact match. If one is found, the corresponding zone number is used. If there is no match against a holiday zone, the standard zones are checked. A standard zone can always be assigned to any date/time because CA MICS requires you to assign every one of the 168 hours in a week to a standard zone (no overlapping is allowed).

A sample definition of time zones follows:

Time Zone	Time Included in Zone Definition	
Number	Days of the Week	Start-End Hours
1	Monday through Friday	8 a.m. - Noon
1	Monday through Friday	1 p.m. - 5 p.m.
2	Monday through Friday	6 a.m. - 8 a.m.
2	Monday through Friday	Noon - 1 p.m.
2	Monday through Friday	5 p.m. - 8 p.m.
3	Monday through Friday	Midnight - 6 a.m.
3	Monday through Friday	8 p.m. - 12 p.m.
3	Saturday and Sunday	All day
4	Holiday 12/25/93	All day
4	Holiday 01/01/94	All day

Figure 2-9B. Time Zone Definition Example

Figure 2-9B illustrates that time zones are simply periods or "windows" within a given week (standard zones) or specific calendar dates (holiday zones).

Figure 2-9C is a worksheet for collecting the data with which to complete sharedprefix.MICS.PARMS(CPLXZONE). It asks you to do the following:

1. Identify the number of standard zones to be defined. Up to nine are allowed. They need not be defined in sequential order, nor need the numbers be contiguous.
2. Define the hours and days of week that each standard zone will include.
3. Define the holiday dates, for as many years as is necessary (holiday zone definition for three years is suggested).
4. Define the hours and days of week that each holiday date will include. In most cases, this is 00 to 23.
5. Define a descriptive name to be used with each zone.

The first entries on the worksheet are completed with sample information. The entry at the top of the form specifies that standard zone 1 includes the hours from 08 (8 a.m.) to 15 (up to, but not including 4:00 p.m.) on weekdays (Monday through Friday). For many sites, this is the definition of "prime" shift, and, indeed, the reporting name assigned to this zone is PRIME TIME. The entry on the second part of the form defines New Year's Day as being part of the holiday zone, number 4.

```
+--------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET: Time Zone Specification              |
|--------------------------------------------------------------------------|
| PARMS Library Member is CPLXZONE                                         |
| Reference Sections: 2.3.2.3, 2.3.2.3.1, and 2.3.2.3.2                    |
+--------------------------------------------------------------------------+
|                                                                          |
|                                  STANDARD ZONES                          |
| ZONE START  END  START  END    SERVICE                                   |
|  NO   DAY   DAY   HOUR  HOUR   FACTOR      ZONE NAME                     |
+--------------------------------------------------------------------------|
|   1   MON   FRI    08    15     1          'PRIME-TIME'                  |
|   (sample)                                                               |
|   _   ___   ___    __    __    ___       '______________________________' |
|   _   ___   ___    __    __    ___       '______________________________' |
|   _   ___   ___    __    __    ___       '______________________________' |
|   _   ___   ___    __    __    ___       '______________________________' |
|   _   ___   ___    __    __    ___       '______________________________' |
|   _   ___   ___    __    __    ___       '______________________________' |
|   _   ___   ___    __    __    ___       '______________________________' |
|   _   ___   ___    __    __    ___       '______________________________' |
|   _   ___   ___    __    __    ___       '______________________________' |
|   _   ___   ___    __    __    ___       '______________________________' |
|   _   ___   ___    __    __    ___       '______________________________' |
+--------------------------------------------------------------------------|
|                                  HOLIDAY ZONES                           |
| ZONE                START  END  SERVICE                                  |
|  NO  YEAR MONTH DAY HOUR  HOUR  FACTOR   ZONE NAME                       |
+--------------------------------------------------------------------------|
|   4   94    01   01  00    23     1     'HOLIDAY SAMPLE'                 |
|   (sample)                                                               |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
+--------------------------------------------------------------------------+
| ....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+--------------------------------------------------------------------------+
```

Figure 2-9C. Time Zone Specification Worksheet

The following summarizes how to code the parameters required in sharedprefix.MICS.PARMS(CPLXZONE). The planning issues are discussed in more detail in the sections that follow.

If you are defining ZONE at the complex level:

- o The member contains one line per standard zone (ZONE) entry and one line per holiday zone entry (HOLIDAY). Comments may be coded by beginning the statement with an asterisk (*).

- o The format of the statements is free-form but positional. All parameters are required. The parameter placement for a standard zone is:

KEYWORD	(ZONE)
ZONE NUMBER	(1 through 9)
STARTING DAY	(SUN MON TUE WED THU FRI SAT)
ENDING DAY	(SUN MON TUE WED THU FRI SAT)
STARTING HOUR	(00 to 23)
ENDING HOUR	(00 to 23)
SERVICE FACTOR	(.1 to 99.9)
ZONE NAME	(1- to 40-character name in quotes)

- o The parameter placement for a holiday zone is:

KEYWORD	(HOLIDAY)
ZONE NUMBER	(1 through 9)
YEAR	(00 through 99 or %%)
MONTH	(01 through 12)
DAY	(01 through 31)
STARTING HOUR	(00 to 23)
ENDING HOUR	(00 to 23)
SERVICE FACTOR	(.1 to 99.9)
ZONE NAME	(1- to 35-character name in quotes)

- o For holidays that occur on the same calendar date regardless of the year, the characters "%%" are specified in place of a year value, thereby treating all occurrences of the date as the holiday regardless of the year. For example, New Year's Day is January 1 regardless of the year, so %% is coded in place of a year value.

One or more statements may be required to fully describe a zone. Remember the following points when specifying time zones.

- o The parameters for standard zones must account for all 168 hours of the week with no overlapping permitted.

- The ending day must follow the starting day within the order specified. (To CA MICS, a week begins on Sunday and ends on Saturday so you may not "wrap" over the weekend. The WEEKSTART parameter of prefix.MICS.PARMS(SITE) and sharedprefix.MICS.PARMS(CPLXDEF) does not affect ZONE specification. The only exception to this rule is that a weekend zone may be specified as starting on Saturday and ending on Sunday.)

- The ending hour must be greater than or equal to the starting hour (note that "15" means 3 p.m. when specified as a starting hour, but means 3:59:59.99 p.m. when specified as an ending hour).

- Day and/or time splits (e.g., a shift that spans midnight or a zone that includes Saturday and Sunday and other days) must be defined with multiple statements.

- The ending day and starting day are the same when the statement defines a single day.

You should modify the sharedprefix.MICS.PARMS(CPLXZONE) member distributed with CA MICS to reflect your data center's needs.

The following sections describe in detail the parameters mentioned above and the planning considerations involved in correctly specifying them:

1 - Standard Time Zones
2 - Holiday Time Zones

Note: If you determine that the standard CA MICS ZONE definition facilities cannot meet your data center's requirements, an exit lets you override and redefine the ZONE value assigned through the standard defintion facility. See the System Modification Guide, Section 4.3.2.1, for more information on the ZONE definition exit.

Important! Be careful when using this exit because changes in the CA MICS ZONE construct can have serious implications, including, but not limited to, total database corruption.

2.3.1.10.1 Standard Time Zones

Whether or not you have appropriately defined your standard time zones will determine whether you can meet your installation's analysis and reporting requirements when they call for using data from the WEEKS, MONTHS, and YEARS time-spans. This is because the time granularity of data in those levels of summarization is no finer than to the ZONE (HOUR within ZONE for WEEKS). For example, if you are asked for a long-term study of off-prime CPU utilization that treats weekdays and weekends separately and off-prime weekdays and weekends are not separate ZONEs in the database, you will not be able to do it from the CA MICS data readily at hand.

Standard zones may not overlap with other standard zones, but of necessity they will overlap with all holiday zones. The holiday zone always takes precedence. Note that a holiday zone number may be the same as a standard zone number, allowing, for example, holidays to be lumped in with your "graveyard" shift.

The format of a statement which defines a standard time zone is:

```
    ZONE n day1 day2 hour1 hour2 sf 'description of zone'
```

The following describes the parameters on this statement.

ZONE NUMBER

A number from 1 to 9, which identifies the time zone. Note that while this specification is numeric, the data element ZONE is a one-byte character field in the CA MICS database files.

There is no requirement that ZONE NUMBERs be defined consecutively (e.g., 1, 2, 5, and 7 would be acceptable), nor is there a requirement that they be defined in numeric order. It is recommended that the zones be defined in order of importance, with the most important defined as 1.

STARTING DAY

The name of the first day within the CA MICS standard week (SUN through SAT) which is to be included in this zone definition. Days are identified by the values SUN, MON, TUE, WED, THU, FRI, or SAT. When specifying a single day, that day's name will be given here and as the "ending day" as well.

ENDING DAY

The name of the last day within the CA MICS standard week (SUN through SAT) which is to be included in this zone definition. Days are identified by the values SUN, MON, TUE, WED, THU, FRI, or SAT. When specifying a single day, that day's name will be given here and as the "starting day" as well. Note that a range of days may not reverse the normal (MICS) order of the days from SUN through SAT. In other words, one could not specify the range FRI through SUN since the specification is not in the order of SUN through SAT. The only exception to this rule is that weekends may be specified as starting on Saturday and ending on Sunday.

BEGINNING HOUR OF THE DAY

The first hour in each of the days specified above which is to be included in the zone definition. Hours are specified as numbers from 00 to 23, and they are inclusive.

ENDING HOUR OF THE DAY

The last hour in each of the days specified above which is to be included in the zone definition. Hours are specified as numbers from 00 to 23, and they are inclusive. When only one hour is to be included, both beginning and ending hours must be specified, and they will be identical. To define a time zone of 8 am to 5 pm, you would code '08 16', which tells CA MICS to include all times between 8:00 am (inclusive) up through 4:59:59.99 pm (16:59:59.99).

SERVICE FACTOR (sf):

The service factor is specified to allow differential zone accounting by Accounting and Chargeback. It allows a discount or surcharge factor to be specified for particular zones to reflect the time of day that the service was provided. For more information on the service factor, see the documentation for Accounting and Chargeback. The allowable range of values for the service factor is 0.1 through 99.9. If multiple ZONE statements are required for one zone, then a '+' must be specified on all but the first statement for that zone. The first statement for each zone must always contain a service factor value. This parameter MUST be specified even if Accounting and Chargeback is not installed.

ZONE NAME

The 1 to 40 character name to use for reporting purposes when this Time Zone is displayed. The name may not contain percent signs or apostrophes. This parameter is required on the first ZONE statement for a given zone number. It is optional on subsequent statements for that same zone number and in any case is ignored.

2.3.1.10.2 Holiday Time Zones

Holiday zones allow you to single out specific dates and times and put them into zones of your choice. One reason to do this is to acknowledge that, from a reporting and analysis point of view, 10 a.m. to noon on New Years Day is a much different time period than 10 a.m. to noon on a normal weekday when you are looking at something like online system resource consumption.

"Holiday," as noted earlier, is a misnomer: CA MICS users use holiday time zones to single out specific dates and times that are not holidays, but are interesting from an analytic viewpoint (usually because they are very busy periods).

The format of a statement that defines a holiday time zone is:

```
  HOLIDAY n year month day hour1 hour2 sf 'zone description '
```

The following describes the parameters on this statement.

ZONE NUMBER (n)

A number from 1 to 9 identifying the time zone. This specification is numeric, however the data element ZONE is a one-byte character field in CA MICS database files. This number may be the same as one mentioned on a ZONE or another HOLIDAY statement. A typical reason to do this is to lump weekends and holidays together.

Zone numbers, should be defined with the prime (most important) zone as zone 1, the next as zone 2, with the holiday zone defined as N (the highest zone number used).

There is no requirement that ZONE NUMBERs be defined consecutively (so 1, 2, 5, and 7 would be acceptable) nor is there a requirement that they be defined in numeric order.

YEAR (year)

A two-digit value of the year in which the special day, being defined by this statement occurs, or a %% to bypass year checking (which is useful for holidays that occur on the same day regardless of the year).

For example, the HOLIDAY definition below illustrates how the %% definition is used to define New Years Day as a holiday.

```
HOLIDAY 4 %% 01 01 00 23 'HOLIDAY ZONE' *  NEW YEARS DAY
```

MONTH (month)

A number from 1 to 12 that specifies the month of the year in which the special day being defined by this statement occurs.

DAY (day)

A number from 1 to 31 that specifies the day of the month on which the special day being defined by this statement occurs.

BEGINNING HOUR OF THE DAY (hour1)

The first hour of the special day to be included in the zone definition. Hours are specified as numbers from 00 to 23, and they are inclusive.

ENDING HOUR OF THE DAY (hour2)

The last hour of the special day to be included in the zone definition. Hours are specified as numbers from 00 to 23, and they are inclusive.

When only one hour is to be included, both beginning and ending hours must be specified, and they will be identical.

To define a time zone of 8 a.m. to 5 p.m., you would code '08 16', which tells CA MICS to include all times between 8:00 a.m. (inclusive) up through 4:59:59.99 p.m. (16:59:59.99).

SERVICE FACTOR (sf)

The service factor is specified to allow differential accounting by CA MICS Accounting and Chargeback. It allows a discount or surcharge factor to be specified for particular zones to reflect the time of day that the service was provided. For more information on the service factor, see the CA MICS Accounting and Chargeback User Guide.

The allowable range of values for the service factor is

0.1 through 99.9. If multiple ZONE statements are required for one zone, then a plus sign (+) must be specified on all but the first statement for that zone. The first statement for each zone must always contain a numeric service factor value. This parameter must be specified even if Accounting and Chargeback is not installed.

ZONE NAME (zone description)

A one- to 40-character name that is used on reports when this Time Zone is displayed. The name cannot contain percent signs or apostrophes. This parameter is required on the first HOLIDAY (or ZONE) statement for a given zone number. On subsequent statements for the same zone number, it is ignored.

2.3.2 Database Unit Planning and Parameters

This section shows you how to specify the operational parameters required to install CA MICS. Operational parameters guide the processing of data in the CA MICS system. They are stored in the prefix.MICS.PARMS library associated with the database being run and are incorporated into the CA MICS system by the base and optional product generation processes (i.e., BASPGEN, SMFPGEN, RMFPGEN, etc.). Changes to parameters do not take effect until the appropriate generator is run, unlike SAS code changes (to exits, for example), which take effect immediately.

The following topics are covered in this section:

1 - An Introduction to the Concept of Summarization
2 - Computing System Parameters (SYSID)
3 - Time Zone Definitions (ZONE)
4 - Site Characteristics (SITE)
5 - Performance Group Names (PRFGP) and Service Classes
6 - Compression (COMPRESS)

For information about specific data integration product's parameters, refer to that product's planning, installation definitions, or parameters chapter in its guide.

2.3.2.1 An Introduction to the Concept of Summarization

A number of the members of the MICS.PARMS library and the parameters therein exist to give you control over the CA MICS process known as "summarization". Very simply stated, summarization is the process of taking a lot of data and reducing it to less data, without losing information required for analysis and reporting purposes. For a given CA MICS file, many records in the DETAIL timespan are consolidated into fewer records in the DAYS timespan. Then those fewer records are consolidated into yet fewer records in the WEEKS and MONTHS timespans which are consolidated into fewer still records in the YEARS timespan.

Summarization is a necessity given the volume of measurement data handled by the CA MICS system. No shop has the DASD space to keep DETAIL-level data online indefinitely, nor would it have the machine resources to process it if the data could be kept.

Summarization is done by "key". In each CA MICS timespan (DETAIL, DAYS, etc.), each CA MICS file (e.g., BATJOB, SCPPGA) is in sequence by the values of a number of its variables. Taken together, these variables are the key of the file.

For example, the BATJOB file in the DETAIL timespan is sequenced by the values of its SYSID, ACCTNOs (account number fields), JOBGROUP, JOB (the jobname), YEAR, MONTH, DAY, and ENDTS (ending time-stamp) variables. As in all CA MICS files, there will exist one record in the file for each unique combination of key variable values, and there will be no duplicate keys. Thus at the DETAIL level there must be one record for each job run on the system during the time covered by the file (the ENDTS variable separates executions of jobs with the same JOBname).

At the MONTHS level, the "key" of the BATJOB file is SYSID, ACCTNO(s) (the accounting fields), JOBGROUP, YEAR, MONTH, and ZONE (zone is a concept similar to "shift"). When a CYCLE in the MONTHS level of the BATJOB file is created, there will be one and only one record included for each unique combination of these variables. As CA MICS never creates a record unless data is actually encountered, at least one job's information will be in each record, and, of course, summarization would not be serving its purpose if, on the average, there were not a good number of jobs consolidated into each record at the MONTHS level.

Setting aside, for the moment, the problem of how information from many records is put into one, we may observe that the importance of certain key fields varies between the DETAIL and MONTHS timespans. At the DETAIL level, the account numbers and the JOBGROUP are not a significant part of the key because there would be the same number of records in the same order containing the same information even if these variables were not part of the key--one for each unique execution of a job. At the MONTHS level, on the other hand, the account numbers and the JOBGROUP are very important parts of the key. JOB(name) and ENDTS have disappeared, and therefore the account numbers and the JOBGROUP mainly determine what information will be present in each CYCLE of the file and in what order.

To give a concrete example of the use of these fields, suppose in your organization there are three programming groups, each with its own manager. Hearing of the capabilities of the new (CA MICS) system you have installed, you are asked to create an inquiry transaction which any manager may run at any time to show, for each of his/her programmers, how much it cost to run the batch jobs they submitted for each project on which they worked during that month so far. Further assume that at your site the programmers identify the project on whose behalf a job is run by the first two characters of the JOBname they use. It is clear that if the input to your report were CA MICS DETAIL level data, you would not much care what the accounting fields or JOBGROUP were set to: you have the JOBnames and programmer name fields for every single job run during the month. It is also clear that the managers would not much care for the response time of the inquiry transaction you gave them -- it would have to pass a record for every batch job run in the entire month!

This is where summarization comes to your aid if you have set up your CA MICS installation correctly. If, anticipating such a request, you set up CA MICS so that the project identifier and programmer identifier were carried in the account number fields, then you could supply your management with an inquiry which runs off of the (much smaller) MONTHS BATJOB file (CYCLE 00 for month-to-date).

How does CA MICS know to assign the right values to variables like the account numbers of the BATJOB file so you can meet your reporting requirements? It doesn't. You must tell how this is to be accomplished via parameters at CA MICS generation time. In the specific case of the batch job (BATJOB) file, the relevant parameters are stored in

MICS.PARMS(ACCOUNT) and MICS.PARMS(ACCTRTE), but there are similar processes which must go on for all CA MICS files which deal with individual units of work. For example, the corresponding members for the IMS component are IMSACCT and IMSACRT.

Remember as you go through the many parameters in this chapter which influence the CA MICS process called summarization that your task is to tell CA MICS a way to consolidate your data in such a way that you can meet your reporting and analytical requirements without keeping so much data that you cannot afford to have it online.

2.3.2.2 Computing System Parameters (SYSID)

Data in the CA MICS database are associated with and summarized according to the computing system that produced the raw data used to create the observations in the CA MICS database.

Computing systems are identified by their SMF system identifiers (SYSIDs), which are generally unique to specific systems. For example, SYSID=P390 might be the Model 3090 Production system. The SYSID parameter enables you to map SMF system identifiers (CA MICS data element ORGSYSID), which will probably change over time, to the business work that takes place on your systems (CA MICS data element SYSID).

You code information about the systems used at your installation in sharedprefix.MICS.PARMS(CPLXSID) and/or prefix.MICS.PARMS(SYSID). The unit-level parameter specifications in prefix.MICS.PARMS(SYSID) override, for this unit database, the complex-level parameter values from sharedprefix.MICS.PARMS(CPLXSID).

Use the worksheet in Figure 2-9 to collect the data needed to code this parameter member.

There are three keyword parameters:

- COMPLEXSYSIDPARM IGNORE/USE
- UNDEFINEDSYSID ABORT/EXCLUDE
- SYSIDCHECK ABORT/AUTODEFINE

You code one line containing seven parameter values for each SYSID:

- SMF SYSID (CA MICS data element ORGSYSID)
- logical SYSID (CA MICS data element SYSID)
- CPU model
- system control program name
- RMF conversion factor
- adjustment factor
- system name

Lines beginning with '*' are comments and are ignored. Blank statements are allowed.

Both the keyword and SYSID definition parameters are described in greater detail later in this section. You can begin your customization by modifying the sample SYSID member in prefix.MICS.PARMS(SYSID).

Example System Definition Entry

```
P390 PROD '3090-200' OS/390 . 14.6 'TSO PRODUCTION SYSTEM'
```

This sample entry specifies an SMF system of P390, which is translated to the logical system PROD. The processor is a 3090-200 running OS/390, the RMF CPU conversion factor is left as a missing value because the RMF data provides this value, the Adjustment Factor is 14.6 (per internal processor), and the reporting name to be displayed for this system is TSO PRODUCTION SYSTEM. (The values used in this example are for illustration purposes only. DO NOT assume that they are correct for your 3090-200 processor).

Considerations for Coding SYSID

By default, CA MICS terminates with a user abend if input data comes from an SMF system that is not defined in the unit-level parameter library. You modify this default behavior through control parameters.

To change unit-level SYSID parameter information after CA MICS is in use, first change the SYSID member in prefix.MICS.PARMS and then execute the job BASPGEN in the prefix.MICS.CNTL library.

Considerations for Analyzers

Refer to the individual product guides for specific considerations relative to specific CA MICS data integration applications. Refer to the CA MICS System Modification Guide for considerations relative to MTI components and Field Developed Applications.

```
+--------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET:  System Identifiers                  |
|                                                                          |
| PARMS Library Member is SYSID                                            |
| Reference Sections:  2.3.2.2                                             |
+--------------------------------------------------------------------------+
|                                                                          |
| COMPLEXSYSIDPARM ________________________ (IGNORE or USE)                |
| UNDEFINEDSYSID  ________________________ (ABORT or EXCLUDE)              |
| SYSIDCHECK  ________________________ (ABORT or AUTODEFINE)               |
|                                                                          |
|                                                                          |
| SMF Logical CPU  RMF Adjust.                                             |
|SYSID SYSID Model SCP Factor Factor 'System Name'                         |
+---------------------------------------------------------------------     |
| P390 PROD '3090-200' OS/390 . 14.6 'SAMPLE TSO PRODUCTION SYSTEM'        |
|                                                                          |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                          |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                          |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                          |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                          |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                          |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                          |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                          |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                          |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                          |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                          |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                          |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                          |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                          |
| ____ ____ ________ ___ _____ _____ '______________________________'|
|                                                                          |
|                                                                          |
+--------------------------------------------------------------------------+
| ....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+--------------------------------------------------------------------------+
```

Figure 2-9. System Description Worksheet

Control Parameters:

COMPLEXSYSIDPARM

Defaults to IGNORE if keyword is not specified.

Specify IGNORE to limit system definitions for this unit database to just those systems defined in prefix.MICS.PARMS(SYSID). With IGNORE, this unit database is completely independent of your global (complex-level) sharedprefix.MICS.PARMS(CPLXSID) specifications.

Specify USE to apply your global (complex-level) sharedprefix.MICS.PARMS(CPLXSID) specifications to this unit database.

- Use this option to define common systems once, in sharedprefix.MICS.PARMS(CPLXSID), and avoid the effort and risk of repeating the same information in multiple unit databases.

- This option also helps you to react more quickly to configuration changes by updating a single member and running a single complex-level generation job (CPLXGEN) rather than having to update a member and run a BASPGEN in EACH unit database.

- You can override complex-level system definitions or control parameter values for a specific unit database by coding the override in prefix.MICS.PARMS(SYSID) and running BASPGEN.

UNDEFINEDSYSID

Defaults to ABORT if keyword is not specified and you are not using the complex-level specification for the default value.

Specify ABORT to terminate database update processing with a U310 abend if data comes from an SMF system that is not defined either at the unit-level in prefix.MICS.PARMS(SYSID) or at the complex-level in sharedprefix.MICS.PARMS(CPLXSID).

Specify EXCLUDE to automatically drop data that comes from an SMF system that is not defined.

- With this option, you not only keep data from unexpected systems out of your CA MICS database, but you also avoid parameter changes, BASPGEN executions, and database update job restarts caused by data from unexpected sources such as test systems.

- Use the EXCLUDE option for "permanent" changes.

- To address temporary or one-time situations, specify

  ```
  EXCLUDEORGSYSIDS UNDEFINEDSYSID
  ```

  in prefix.MICS.PARMS(EXECDEF) to override an UNDEFINEDSYSID ABORT specification.

SYSIDCHECK

Defaults to ABORT if keyword is not specified and you are not using the complex-level specification for the default value.

Specify ABORT to terminate cccPGEN processing if ORGSYSID or SYSID values appear in a CA MICS product parameter member (e.g., CICOPS, SMFOPS, PRFGP, etc.) but are not defined either at the unit-level in prefix.MICS.PARMS(SYSID), or at the complex-level in sharedprefix.MICS.PARMS(CPLXSID).

Specify AUTODEFINE to have BASPGEN or ALLPGEN automatically create new prefix.MICS.PARMS(SYSID) entries for previously undefined ORGSYSID and/or SYSID values found in CA MICS product parameter members (e.g., CICOPS, SMFOPS, PRFGP, etc.). With this option, the following happens:

- BASPGEN, ALLPGEN, and product cccPGEN jobs (e.g., SMFPGEN, CICPGEN, etc.) continue to issue MICSLOG warning messages when ORGSYSID or SYSID values appear in CA MICS product parameters without corresponding entries in prefix.MICS.PARMS(SYSID) or sharedprefix.MICS.PARMS(CPLXSID).

- Product cccPGEN jobs (e.g., SMFPGEN, CICPGEN, etc.) issue MICSLOG warning messages instructing you to run BASPGEN in order to create required entries in prefix.MICS.PARMS(SYSID).

- BASPGEN and ALLPGEN save the original, unmodified SYSID member as prefix.MICS.PARMS(SYSIDnnn), where nnn is a generated, unique identifier, and then append new entries to prefix.MICS.PARMS(SYSID). You can delete the SYSIDnnn member at any time as it is created only to give you a "before image" of automatic changes.

- o Autodefined SYSID entries are created as follows.

  - SMF SYSID (ORGSYSID) is set to the new ORGSYSID/SYSID value from the CA MICS product parameter member.

  - Logical SYSID (SYSID) is set to match ORGSYSID.

  - CPU model is set to UNKNOWN.

  - System control program name is set to UNKNOWN.

  - RMF conversion factor is set to missing value ('.').

  - Adjustment factor is set to missing value ('.').

  - System name is set to the string "SYSTEM " followed by the ORGSYSID value.

- o You can correct/revise autodefined SYSID entries by editing the new prefix.MICS.PARMS(SYSID) and running BASPGEN.

SMF SYSID: the Original or Input System Identifier

First, identify all of the SYSIDs currently being used. (These are contained in SMF records and are referred to as original system IDs.) Then ensure that the SYSIDs are unique.

If the SYSIDs are not unique, work with your site's system programmers to assign unique SYSIDs to each system.

Logical SYSID: the Logical System Identifier

The logical system identifier is the key under which almost all measurement data in the CA MICS database is stored. This value, which is kept in the CA MICS common data element SYSID, is the highest key in almost all CA MICS files.

You can specify that the original and logical system identifiers be identical. If, however, you have a current system identifier that may change, we urge you to implement a unique logical system identifier.

*** Warning ***

You cannot assign two concurrently used SYSIDs to the same logical SYSID. If a logical SYSID is specified more than once, the CA MICS generation process will flag the condition as an error. Contact CA Technical Support for advice on how to handle a situation where both the original machine and its successor run side-by-side for a transition period.

CPU Model

Some data sources do not identify the CPU model on which the data was produced. To ensure that this information is available in your CA MICS database, you provide a CPU model identifier in the SYSID parameter.

Code this parameter as 1-20 characters in quotation marks (e.g., '3090-400J') with no blanks inside the quotes.

While CA MICS does not process data based on this value, it is available to a SAS program written to run in the CA MICS environment via a SAS macro.

SCP: the System Control Program Level

The SCP keyword provides the level of MVS currently being run on the system described by this parameter statement.

Code this parameter as 1-10 characters with no imbedded blanks (e.g., OS/390).

While CA MICS does not process data based on this value, it is available to a SAS program written to run in the CA MICS environment via a SAS macro.

RMF Factor: the RMF Service-Units-to-CPU-Time Conversion
            Factor (Optional)

The RMF factor keyword asks for the value that CA MICS should use to convert TCB and SRB service units, reported by RMF, into CPU times.

The releases of RMF supported by CA MICS provide this value, so code a period (translated as a SAS missing value) for this keyword. Your "missing value" will be overridden by the value found in the Workload Activity Record.

Note that any value other than '.' will be ignored during processing.

Adjustment Factor (Optional)

The CA MICS database enables you to retain information about CPU consumption in units of time. This information is retained in elements whose names take the form fffxxxTM, where fff is the file identifier, xxx is file-specific, and TM stands for time.

Over time, the validity of time as a measure of CPU use becomes questionable due to hardware environment changes. (For example, one second of ES/9000 time is not the same as one second of 3033 time.)

To compensate for this difference, CA MICS enables you to retain the number of instructions executed. This information is retained in elements whose names take the form fffxxxNI, where fff is the file identifier, xxx is file-specific, and NI stands for number of instructions.

The number of instructions executed is computed from the measured CPU time and an adjustment factor you choose, such as a MIPS rating for an MVS system. The adjustment factor is optional. If you code a zero, the fffxxxNI elements will have a zero value.

In choosing an adjustment factor, consider these sources:

- Conduct a hardware monitor study to determine your processor's rating. This method yields the most reliable results, but is costly to implement.

- Ask the hardware manufacturer for a normalized rating, such as a MIPS rate. (If your site operates in a multiprocessor environment, ask for the rate for a single processor.)

- Contact a hardware monitor manufacturer and request normalized ratings.

- Ask colleagues if they have information they are willing to share.

- Review the RMF Service-Units-to-CPU-Time Conversion Factor MVS uses to normalize cpu times across different processor models.

System Name: the Descriptive Name of the System

With this keyword you supply a 1-30 character name for the system identified on the parameter card. This keyword must be enclosed in single quotes.

The value you specify is used to identify the system in certain CA MICS standard reports. It is not carried in the database, but is available to user-written routines by means of a SAS macro created when the CA MICS system is generated.

2.3.2.3 Time Zone Definitions (ZONE)

The definition of time zones may be one of the most important made in terms of tailoring the database to your installation's requirements because the facility to summarize data by zone provides the ability to evaluate, both short-term and long-term, service, availability, and load according to the service periods established. For instance, in evaluating online service, management's attention is focused on the activity of the peak load zones, and little, if any, attention is directed at the non-peak zones.

You have the option to use the complex-level zone definition specified in sharedprefix.MICS.PARMS(CPLXZONE), or to define a unique zone definition for this unit database in prefix.MICS.PARMS(ZONE). As a general rule, you should use the complex-level parameter to establish a consistent time zone definition for all units. In this way, you can maintain your holiday and special shift definitions in a single parameter member, and implement changes with a single CA MICS parameter generation. In using this approach, the individual unit database prefix.MICS.PARMS(ZONE) members will contain the single parameter line,

```
COMPLEXZONEPARM USE
```

Use the unit-level time ZONE parameters to override the complex-level time zone definitions to address special requirements for different time zone definitions for different units of work. For example one database unit may support a remote data center, or you may have unique shift differential requirements for online transaction workloads separate from standard TSO and batch workloads. In these situations, you isolate the special workloads to a unique unit database and fully specify a unique time zone definition in prefix.MICS.PARMS(ZONE) for that unit database. Please note, the unit-level ZONE definition completely replaces and overrides the complex-level ZONE definition for this unit database. You control this choice with the COMPLEXZONEPARM USE/IGNORE parameter described in more detail below.

To use the complex-level time zone definition, specify,

```
         COMPLEXZONEPARM USE
```

as the only parameter in prefix.MICS.PARMS(ZONE). See section 2.3.1.10 for more information on complex-level time zone definition.

To override the complex-level time zone definition for this unit database, follow the instructions in the remainder of this section to specify a complete unit-level time zone definition in prefix.MICS.PARMS(ZONE). After completing the time zone definition, execute prefix.MICS.CNTL(BASPGEN) to implement your specifications.

In prefix.MICS.PARMS(ZONE), you are asked to define up to nine service periods (time zones) for your installation. Time zones are defined once and apply to all Information Areas within a database. It is not possible to have a time zone for TSO and another time zone for RMF in the same database. These service periods, which are also sometimes referred to as "shifts," are applied over all batch and online work examined by CA MICS.

The time zone, which is identified by the data element ZONE in the database, is used for summarizing data in the WEEKS, MONTHS, and YEARS timespans (it becomes part of the key of the files, replacing DAY, HOUR, and End Time-Stamp). The ZONE data element also improves the user's ability to select or extract data easily from files in the DETAIL and DAYS timespans where it exists but is not part of the key.

The number of zones defined impacts the projected space requirements of the database since the data element ZONE is used for summarization (is part of the key of the files) in the WEEKS, MONTHS, and YEARS timespans. The more zones, the larger the online DASD space requirements for the database. We recommend that the number of standard zones defined be four or fewer, with one zone for holidays.

CA MICS supports two kinds of ZONEs:

Standard zones are periods that recur in each week. For example, a standard zone can be defined as "9 a.m. to 5 p.m. on Mondays through Fridays, except for holidays".

Holiday zones are periods that are precisely identified by date and time. For example, "9 a.m. to 5 p.m. on New Years Day, January 1, 1993". In addition, some installations have used holiday zones to identify "super peak" periods such as end-of-month closings.

In determining the ZONE into which a specific date/time combination found in a measurement record falls, CA MICS first checks the holiday zones for an exact match. If one is found, the corresponding zone number is used. If there is no match against a holiday zone, the standard zones are checked. A standard zone can always be assigned to any date/time because CA MICS requires you to assign every one of the 168 hours in a week to a standard zone (no overlapping is allowed for standard zones).

A sample definition of time zones follows:

Time Zone	Time Included in Zone Definition	
Number	Days of the Week	Start-End Hours
1	Monday through Friday	8 a.m. - Noon
1	Monday through Friday	1 p.m. - 5 p.m.
2	Monday through Friday	6 a.m. - 8 a.m.
2	Monday through Friday	Noon - 1 p.m.
2	Monday through Friday	5 p.m. - 8 p.m.
3	Monday through Friday	Midnight - 6 a.m.
3	Monday through Friday	8 p.m. - 12 p.m.
3	Saturday and Sunday	All day
4	Holiday 12/25/93	All day
4	Holiday 01/01/94	All day

Figure 2-10. Time Zone Definition Example

Figure 2-10 illustrates that time zones are simply periods or "windows" within a given week (standard zones) or specific calendar dates (holiday zones).

Figure 2-11 is a worksheet for collecting the data with which to complete prefix.MICS.PARMS(ZONE). It asks you to do the following:

1. Identify the number of standard zones to be defined. Up to nine are allowed. They need not be defined in sequential order, nor need the numbers be contiguous.

2. Define the hours and days of week that each standard zone will include.
3. Define the holiday dates, for as many years as is necessary (holiday zone definition for three years is suggested).

4. Define the hours and days of week that each holiday date will include. In most cases, this is 00 to 23.

5. Define a descriptive name to be used with each zone.

The first entries on the worksheet are completed with sample information. The entry at the top of the form specifies that standard zone 1 includes the hours from 08 (8 a.m.) to 15 (up to, but not including 4:00 p.m.) on weekdays (Monday through Friday). For many sites, this is the definition of "prime" shift, and, indeed, the reporting name assigned to this zone is PRIME TIME. The entry on the second part of the form defines New Years Day as being part of the holiday zone, number 4.

```
+--------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET:  Time Zone Specification             |
|--------------------------------------------------------------------------|
|                                                                          |
| COMPLEXZONEPARM_____________________USE/IGNORE                           |
|                                                                          |
|--------------------------------------------------------------------------|
| PARMS Library Member is ZONE                                             |
| Reference Sections:  2.3.1.10, 2.3.2.3, 2.3.2.3.1, and 2.3.2.3.2         |
+--------------------------------------------------------------------------+
|                                                                          |
|                                 STANDARD ZONES                           |
| ZONE START  END  START  END    SERVICE                                   |
|  NO   DAY   DAY   HOUR  HOUR   FACTOR      ZONE NAME                     |
+-------------------------------------------------------------------------|
|   1   MON   FRI    08    15     1          'PRIME-TIME'                  |
|   (sample)                                                               |
|   _   ___   ___    __    __    ___      '______________________________' |
|   _   ___   ___    __    __    ___      '______________________________' |
|   _   ___   ___    __    __    ___      '______________________________' |
|   _   ___   ___    __    __    ___      '______________________________' |
|   _   ___   ___    __    __    ___      '______________________________' |
|   _   ___   ___    __    __    ___      '______________________________' |
|   _   ___   ___    __    __    ___      '______________________________' |
|   _   ___   ___    __    __    ___      '______________________________' |
|   _   ___   ___    __    __    ___      '______________________________' |
|   _   ___   ___    __    __    ___      '______________________________' |
|   _   ___   ___    __    __    ___      '______________________________' |
+-------------------------------------------------------------------------|
|                                 HOLIDAY ZONES                            |
| ZONE                START  END  SERVICE                                  |
|  NO  YEAR MONTH DAY HOUR  HOUR  FACTOR   ZONE NAME                       |
+-------------------------------------------------------------------------|
|   4   94    01   01  00    23     1     'HOLIDAY SAMPLE'                 |
|   (sample)                                                               |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
|   _   __    __   __  __    __    ___    '______________________________' |
+--------------------------------------------------------------------------+
| ....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+--------------------------------------------------------------------------+
```

Figure 2-11. Time Zone Specification Worksheet

The following shows you how to code the parameters required in prefix.MICS.PARMS(ZONE). The planning issues are discussed in more detail in the sections that follow.

For the COMPLEXZONEPARM USE/IGNORE option:

- If you defined your zones in sharedprefix.MICS.PARMS(CPLXZONE), then specify COMPLEXZONEPARM USE to apply this definition to this unit database. Remember to comment out all of the zone and holiday parameters if you specify COMPLEXZONEPARM USE.

- If you did not define your zones in sharedprefix.MICS.PARMS(CPLXZONE), or if you want to override the common zone definition with a unique zone definition for this unit database, omit the COMPLEXZONEPARM statement or specify COMPLEXZONEPARM IGNORE.

If you are defining ZONE at the unit level:

- The member contains one line per standard zone (ZONE) entry and one line per holiday zone entry (HOLIDAY). Comments may be coded by beginning the statement with an asterisk (*).

- The format of the statements is free-form but positional. All parameters are required. The parameter placement for a standard zone is:

KEYWORD	(ZONE)
ZONE NUMBER	(1 through 9)
STARTING DAY	(SUN MON TUE WED THU FRI SAT)
ENDING DAY	(SUN MON TUE WED THU FRI SAT)
STARTING HOUR	(00 to 23)
ENDING HOUR	(00 to 23)
SERVICE FACTOR	(.1 to 99.9)
ZONE NAME	(1- to 40-character name in quotes)

The parameter placement for a holiday zone is:

KEYWORD	(HOLIDAY)
ZONE NUMBER	(1 through 9)
YEAR	(00 through 99 or %%)
MONTH	(01 through 12)
DAY	(01 through 31)
STARTING HOUR	(00 to 23)
ENDING HOUR	(00 to 23)
SERVICE FACTOR	(.1 to 99.9)
ZONE NAME	(1- to 35-character name in quotes)

- For holidays that occur on the same calendar date regardless of the year, the characters "%%" are specified in place of a year value, thereby treating all occurrences of the date as the holiday regardless of the year. For example, New Years Day is January 1 regardless of the year, so %% is coded in place of a year value.

One or more statements may be required to fully describe a zone. Remember the following points when specifying time zones.

- The parameters for standard zones must account for all 168 hours of the week with no overlapping permitted.

- The ending day must follow the starting day within the order specified. (To CA MICS, a week begins on Sunday and ends on Saturday so you may not "wrap" over the weekend. The WEEKSTART parameter of prefix.MICS.PARMS(SITE) and sharedprefix.MICS.PARMS(CPLXDEF) does not affect ZONE specification. The only exception to this rule is that a weekend zone may be specified as starting on Saturday and ending on Sunday.)
- The ending hour must be greater than or equal to the starting hour (note that "15" means 3 p.m. when specified as a starting hour, but means 3:59:59.99 p.m. when specified as an ending hour).

- Day and/or time splits (e.g., a shift which spans midnight or a zone which includes Saturday and Sunday and other days) must be defined with multiple statements.

- The ending day and starting day are the same when the statement defines a single day.

You should modify the prefix.MICS.PARMS(ZONE) member that is generated by the COPYLIBS job to reflect your installation needs.

The following sections describe in detail the parameters mentioned above and the planning considerations involved in correctly specifying them:

1 - Standard Time Zones
2 - Holiday Time Zones

NOTE: If you determine that the standard CA MICS ZONE definition facilities can not meet your installations requirements, an exit is available where you can override and redefine the ZONE value assigned through the standard definition facility. See the System Modification Guide, Section 4.3.2.1, for more information on the ZONE definition exit. Please consider use of this exit very carefully as changes in the CA MICS ZONE construct can have serious implications, including, but not limited to, total database corruption.

2.3.2.3.1 Standard Time Zones

Whether or not you have appropriately defined your standard time zones will determine whether you can meet your installation's analysis and reporting requirements when they call for using data from the WEEKS, MONTHS, and YEARS timespans. This is because the time granularity of data in those levels of summarization is no finer than to the ZONE (HOUR within ZONE for WEEKS). For example, if you are asked for a long-term study of off-prime CPU utilization which treats weekdays and weekends separately, but they are not separate ZONEs in the database, you will not be able to do it from the CA MICS data readily at hand.

Standard zones may not overlap with other standard zones, but of necessity they will overlap with all holiday zones. The holiday zone always takes precedence. Note that a holiday zone number may be the same as a standard zone number, allowing, for example, holidays to be lumped in with your "graveyard" shift.

The format of a statement which defines a standard time zone is:

```
ZONE n day1 day2 hour1 hour2 sf 'description of zone'
```

The following describes the parameters on this statement.

ZONE NUMBER

A number from 1 to 9, which identifies the time zone. Note that while this specification is numeric, the data element ZONE is a one-byte character field in the files of the CA MICS database.

There is no requirement that ZONE NUMBERs be defined consecutively (e.g., 1, 2, 5, and 7 would be acceptable) nor is there a requirement that they be defined in numeric order. It is recommended that the zones be defined in order of importance, with the most important being defined as 1.

STARTING DAY

The name of the first day within the CA MICS standard week (SUN through SAT) which is to be included in this zone definition. Days are identified by the values SUN, MON, TUE, WED, THU, FRI, or SAT. When specifying a single day, that day's name will be given here and as the "ending day" as well.

ENDING DAY

The name of the last day within the CA MICS standard week (SUN through SAT) which is to be included in this zone definition. Days are identified by the values SUN, MON, TUE, WED, THU, FRI, or SAT. When specifying a single day, that day's name will be given here and as the "starting day" as well. Note that a range of days may not reverse the normal (CA MICS) order of the days from SUN through SAT. In other words, one could not specify the range FRI through SUN since the specification is not in the order of SUN through SAT. The only exception to this rule is that weekends may be specified as starting on Saturday and ending on Sunday.

BEGINNING HOUR OF THE DAY

The first hour in each of the days specified above which is to be included in the zone definition. Hours are specified as numbers from 00 to 23, and they are inclusive.

ENDING HOUR OF THE DAY

The last hour in each of the days specified above which is to be included in the zone definition. Hours are specified as numbers from 00 to 23, and they are inclusive. When only one hour is to be included, both beginning and ending hours must be specified, and they will be identical. To define a time zone of 8 am to 5 pm, you would code '08 16', which tells CA MICS to include all times between 8:00 am (inclusive) up through 4:59:59.99 pm (16:59:59.99).

SERVICE FACTOR (sf):

The service factor is specified to allow differential zone accounting by Accounting and Chargeback. It allows a discount or surcharge factor to be specified for particular zones to reflect the time of day that the service was provided. For more information on the service factor, see the documentation for Accounting and Chargeback. The allowable range of values for the service factor is 0.1 through 99.9. If multiple ZONE statements are required for one zone, then a '+' must be specified on all but the first statement for that zone.

The first statement for each zone must always contain a service factor value. This parameter must be specified even if Accounting and Chargeback is not installed.

ZONE NAME

The 1 to 40 character name that is to be used for reporting purposes when this Time Zone is displayed. The name may not contain percent signs or apostrophes. This parameter is required on the first ZONE statement for a given zone number. It is optional on subsequent statements for that same zone number and in any case is ignored.

2.3.2.3.2 Holiday Time Zones

Holiday zones allow you to single out specific dates and times and put them into zones of your choice. One reason to do this is to acknowledge that, from a reporting and analysis point of view, 10 a.m. to noon on New Years Day is a much different time period than 10 a.m. to noon on a normal weekday when you are looking at something like online system resource consumption.

"Holiday," as noted earlier, is a misnomer: CA MICS users use holiday time zones to single out specific dates and times that are not holidays, but are interesting from an analytic viewpoint (usually because they are very busy periods).

The format of a statement which defines a holiday time zone is:

```
  HOLIDAY n year month day hour1 hour2 sf 'zone description '
```

The following describes the parameters on this statement.

ZONE NUMBER (n)

A number from 1 to 9 that identifies the time zone. This specification is numeric, however the data element ZONE is a one-byte character field in CA MICS database files. This number may be the same as one mentioned on a ZONE or another HOLIDAY statement. A typical reason to do this is to lump weekends and holidays together.

In assigning zone numbers, they should be defined with the prime (most important) zone as zone 1, the next as zone 2, with the holiday zone defined as N (the highest zone number used).

There is no requirement that ZONE NUMBERs be defined consecutively (so 1, 2, 5, and 7 would be acceptable) nor is there a requirement that they be defined in numeric order.

YEAR (year)

A 2-digit value of the year in which the special day being defined by this statement occurs, or a %% to bypass the year checking (which is useful for holidays that occur on the same day regardless of the year).

For example, the HOLIDAY definition below illustrates how the %% definition is used to define New Years Day as a holiday.

```
HOLIDAY 4 %% 01 01 00 23  'HOLIDAY ZONE' * NEW YEARS DAY
```

MONTH (month)

A number from 1 to 12 that specifies the month of the year in which the special day being defined by this statement occurs.

DAY (day)

A number from 1 to 31 that specifies the day of the month on which the special day being defined by this statement occurs.

BEGINNING HOUR OF THE DAY (hour1)

The first hour in the day specified above which is to be included in the zone definition. Hours are specified as numbers from 00 to 23, and they are inclusive.

ENDING HOUR OF THE DAY (hour2)

The last hour in the day specified above which is to be included in the zone definition. Hours are specified as numbers from 00 to 23, and they are inclusive.

When only one hour is to be included, both beginning and ending hours must be specified, and they will be identical.

To define a time zone of 8 a.m. to 5 p.m., you would code '08 16', which tells CA MICS to include all times between 8:00 a.m. (inclusive) up through 4:59:59.99 p.m. (16:59:59.99).

SERVICE FACTOR (sf)

The service factor is specified to allow differential accounting by CA MICS Accounting and Chargeback. It allows a discount or surcharge factor to be specified for particular zones to reflect the time of day that the service was provided. For more information on the service factor, see the CA MICS Accounting and Chargeback User Guide.

The allowable range of values for the service factor is

0.1 through 99.9. If multiple ZONE statements are required for one zone, then a plus sign (+) must be specified on all but the first statement for that zone. The first statement for each zone must always contain a numeric service factor value. This parameter must be specified even if Accounting and Chargeback is not installed.

ZONE NAME (zone description)

A one- to 40-character name that is used on reports when this Time Zone is displayed. The name cannot contain percent signs or apostrophes. This parameter is required on the first HOLIDAY (or ZONE) statement for a given zone number. On subsequent statements for the same zone number, it is ignored.

2.3.2.4 Site Characteristics (SITE)

Miscellaneous data items describing your database unit are specified to CA MICS via the parameters in prefix.MICS.PARMS(SITE). The SITE designation means that the parameters here are specific to the database unit associated with the "prefix" rather than to all CA MICS databases that are installed at your site. The parameters specified include headers for CA MICS reports, special calendar definitions, the start of the week, and an option to control the activation of the YEARS timespan.

A subset of these parameters (YEARS TIMESPAN, WEEKSTART, and 13MONTHYEAR) default to the values you specified in sharedprefix.MICS.PARMS(CPLXDEF). You have the option to use the complex-level default, or to override the complex-level parameter specification to define unique options for this unit database. In either case, the parameter specification (either explicit or default) is implemented by running the BASPGEN job in this unit database. See Section 2.3.1.8 for more information on specifying this parameter at the complex level.

Figure 2-12 is a worksheet to help you collect the data needed to complete prefix.MICS.PARMS(SITE). As with most CA MICS PARMS members, the format of statements is free-form but positional. Comments are coded by beginning the statement with an asterisk (*). Blank statements are allowed.

We recommend that you modify the sample SITE member supplied with the distributed MICS.PARMS library rather than code it from scratch. Starting with an existing, correct specification can prevent syntax errors. A sample SITE member is shown below:

```
     *
     * SITE PARAMETER DEFINITIONS
     *
      NAME 'C A'
      YEARS TIMESPAN INACTIVE
      WEEKS TIMESPAN INACTIVE
      WEEKSTART SUN
      13MONTHYEAR NO
      CKPTCNT 100
      AUDITCYC 10
```

The following items are specified once for each database unit (the statement name is given in parenthesis):

Installation Name (NAME):

The 1- to 66-character name used to identify the site. This statement is required. Note that all percent signs (%) in the name will be translated to blanks to avoid conflicts with SAS macro definition conventions.

YEARS TIMESPAN Option (YEARS TIMESPAN):

The files in the YEARS timespan are updated as part of the MONTHLY processing job if the YEARS timespan is active at your site.

The default unit database parameters are distributed as:

YEARS TIMESPAN INACTIVE

This saves both DASD space and processing time by not updating the YEARS files.

Delete this parameter to use the complex-level default from sharedprefix.MICS.PARMS(CPLXDEF).

If you decide that the YEARS timespan should be used to update the YEARS file, specify ACTIVE for this option.

Note that CA MICS still maintains a prefix.MICS.YEARS data set when the YEARS timespan is inactive. The files in this data set contain zero SAS observations and are not updated during MONTHLY processing.

If the YEARS timespan is changed from ACTIVE to INACTIVE after data is processed by CA MICS, any SAS observations in the year-to-date files are dropped during the MONTHLY process following the parameter change. Data in files other than the year-to-date files are not affected by changing the parameter to INACTIVE from ACTIVE.

WEEKS TIMESPAN Option (WEEKS TIMESPAN):

The files in the WEEKS timespan are updated as part of the DAILY processing job if the WEEKS timespan is active at your site.

By default, the unit database parameters are distributed without the WEEKS TIMESPAN statement. Insert this parameter to override the complex-level default from sharedprefix.MICS.PARMS(CPLXDEF) to deactivate the WEEKS timespan.

Note that CA MICS still maintains a prefix.MICS.WEEKS data set when the WEEKS timespan is inactive. The files in this data set contain zero SAS observations and are not updated during DAILY processing.

If the WEEKS timespan is changed from ACTIVE to INACTIVE after data is processed by CA MICS, any SAS observations in the week-to-date files are dropped during the DAILY process following the parameter change. Data in files other than the week-to-date files are not affected by changing the parameter to INACTIVE from ACTIVE.

WEEKSTART Option (WEEKSTART):

This option specifies the first day of the week, whose value can be changed from SUN to one of the following: MON TUE WED THU FRI SAT. Only use the WEEKSTART option if you intend to override the complex-level default specified in sharedprefix.MICS.PARMS(CPLXDEF). WEEKSTART is used to compute the value of WEEK from a given date.

Before specifying this parameter, you should first consider the impact of having different week definitions in different unit databases. In general, you should define WEEKSTART for your site once in sharedprefix.MICS.PARMS(CPLXDEF) in order to have a consistent definition of WEEK across all units. Then, specify WEEKSTART in prefix.MICS.PARMS(SITE) only for those unit databases that require a unique week definition to address specific requirements; such as, those resulting from a unit database supporting a remote data center, or from a unique isolated workload.

Thirteen Month Fiscal Year Option (13MONTHYEAR):

The standard definitions used by CA MICS for DAY, WEEK, MONTH, and YEAR are based on a calendar year beginning January 1 and having 12 months, with the number of days per month varying from 28 to 31 depending on the month. Data is summarized in the CA MICS database according to these standard definitions. If your organization operates on the standard 12-month year and follows the common North American understanding of the first week of a new year, DO NOT code this option.

The default for this option is NO. Only use this option if you intend to override the default. Before doing so, we strongly recommend that you review Section 4.7.2 in the System Modification Guide for information about adjusting algorithms and year-end boundaries.

Note: If you defined your special fiscal calendar in sharedprefix.MICS.PARMS(CPLXDEF), then you can omit the 13MONTHYEAR parameter for this unit database. The complex-level default will be applied. However, if you want this unit database to operate under the standard 12 month calendar AND you defined a special fiscal calendar in CPLXDEF, specify the following:

```
13MONTHYEAR NO
```

to force this unit database to use the standard 12 month Gregorian calendar.

13MONTHYEAR has been provided to enable non-standard calendar-related options. In particular, it enables CA MICS to summarize and store data for an organization that operates on a 13-month fiscal year. It also allows a year to have up to 380 days.

Note: The CA MICS Accounting and Chargeback component includes facilities for defining a unique accounting calendar separate from the global CA MICS calendar definitions. This capability is designed to help you meet your requirements for chargeback and accounting relative to your company's fiscal calendar while continuing to use the standard 12-month calendar for the majority of your CA MICS information. See the CA MICS Accounting and Chargeback guides for more information before altering the global CA MICS calendar.

Before specifying this parameter, you should first consider the impact of having different calendar definitions in different unit databases. In general, you should define your fiscal calendar once in sharedprefix.MICS.PARMS(CPLXDEF) in order to have a consistent calendar definition across all units. Then, specify the 13MONTHYEAR parameter in prefix.MICS.PARMS(SITE) only for those unit databases that require a unique fiscal calendar to address specific requirements; such as, those resulting from a unit database supporting a remote data center, or from a unique isolated workload.

This statement can be formatted in any of the following ways:

```
13MONTHYEAR NO
13MONTHYEAR NO  #DWMY=memname
13MONTHYEAR YES ddmonyy ddmonyy ...  ddmonyy
13MONTHYEAR YES #DWMY=memname ddmonyy ddmonyy ...
```

Note: Only one 13MONTHYEAR control statement is allowed. Subsequent 13MONTHYEAR statements replace prior 13MONTHYEAR statements.

where:

13MONTHYEAR is the option keyword.

NO is specified if you use a standard 12-month year beginning on January 1 and ending on December 31. The default value shipped with CA MICS is NO. It causes the SOURCE member #DWMY12 to be used.

YES is specified only if your organization operates on a 13-month year.

memname is a member that contains date macros. It MUST begin with the characters #DWMY. For example: #DWMYUSR.

The ddmonyy keyword defines the year's starting date. Valid values are:

dd	01 to 31 to specify a day of the month
mon	JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC to specify the month
yy	two digits to specify the year

Dates must be in ascending order and a year cannot contain more than 380 days. You cannot code two dates using the same year value (that is, 01JAN98 and 28DEC98). You can specify a maximum of eight dates (that is, the 13MONTHYEAR statement can define no more than 8 years).

Here is an example:

```
13MONTHYEAR YES 01OCT97 30SEP98
```

is interpreted as fiscal year 1997, which begins on

October 1, 1997 and ends on September 29, 1998. Fiscal year 1998 begins on September 30, 1998 and ends on September 29, 1999.

#DWMY=memname allows you to change the definition of date-related SAS macros and affects the entire CA MICS system. Sharedprefix.MICS.SOURCE(memname) contains these macros that define the day, week, month, year, year start date, previous week (used to select data for weekly reports), and the number of months per year. You should use the supplied member #DWMY12 (containing macros corresponding to the data dictionary definitions), or #DWMY13 (supporting a 13 month calendar), or #DWMYWK1 (supporting a 12 month calendar with the first week of a year being the first week with 4 or more days in that year) as guides for constructing new members. For example, when #DWMY12 is modified directly, you also need to make the same changes to $DWMY12. The #DWMY members contain macro definitions written in SAS MACRO statements, while the $DWMY members contain the corresponding macros written in SAS MACRO language.

If this option is used to change the definition of one or more macros, you can retain the other macro definitions by starting your #DWMY member with a %INCLUDE statement to include one of the standard #DWMY members, followed by a redefinition of the macros you want to change.

Note: CA MICS abends if data is encountered for a year that does not have a defined starting date.

When you activate the Thirteen Month Fiscal Year Option, you change the definitions of the following CA MICS data elements:

YEAR - the fiscal year minus 1900 is set according to the first calendar year of the fiscal year (for example, if the start of your fiscal year is October 1, 2001, the data element YEAR equals 101 (2001-1900) for the fiscal year October 1, 2001 to September 30, 2002).

If your fiscal year value is different from the calendar year of the date that starts your fiscal year, edit the _YEAR macro of sharedprefix.MICS.SOURCE(#DWMYxx) and the YEAR macro of sharedprefix.MICS.SOURCE($DWMYxx) to reflect the correct year value. The standard macro computation subtracts 1900 from the calendar year of the fiscal year start date to produce the YEAR value. The

1900 value is the one you need to edit for both the _YEAR and YEAR macros. For example, if your fiscal year is one greater than the calendar year, change the macros as follows:

```
   default macro shipped with CA MICS in #DWMYxx:
   MACRO _YEAR _SETYRST YEAR=YEAR(YRSTART)-1900; %

   customized macro for your site:
   MACRO _YEAR _SETYRST YEAR=YEAR(YRSTART)-1899;  %

   default macro shipped with CA MICS in $DWMYxx:
   %MACRO YEAR;
    %SETYRST;
    YEAR=YEAR(YRSTART)-1900;
   %MEND YEAR;

   customized macro for your site:
   %MACRO YEAR;
    %SETYRST;
    YEAR=YEAR(YRSTART)-1899;
   %MEND YEAR;
```

MONTH - the month of the year ranges from 01 to 13 and is based on 28-day time periods. For example, if your fiscal year begins on October 1, the last day of MONTH01 is October 28 and the first day of MONTH02 is October 29.

WEEK - the week of the year ranges from 01 to 54 and is based on the first day of the fiscal year you specify. By default, CA MICS weeks always end on Saturday. For example, if your fiscal year begins on Wednesday, October 1, WEEK01 spans Wednesday, October 1 through Saturday, October 4. WEEK 02 spans Sunday, October 5 through Saturday, October 11.

The WEEKSTART option defined earlier in this section enables you to change the Sunday default that is shipped with CA MICS.

DAY - the day of the month ranges from 01 to 28, with the day determined by its position relative to the first day of the 28-day month. The exception to the 28-days-per-month rule comes in MONTH13, when DAY is greater than 28, if the year has more than 364 days. For example, if the fiscal year begins on October 1, the last day of MONTH01 is October 28. October 28's DAY value is 28. October 29 has a MONTH value of 02 and a DAY value of 01.

Consult the data dictionary appendixes in your DIA guides for more information on these data elements, and the CA MICS System Modification Guide, Section 4.7, for more information on calendar derivation.

Changing the values of the DAY, WEEK, MONTH, and YEAR data elements to match the Thirteen Month Fiscal Year Option changes the summarization of the DAYS, WEEKS, MONTHS, and YEARS timespans in the CA MICS database.

Use of the Thirteen Month Fiscal Year Option also affects operational and scheduling processes due to the new definitions of the DAY, WEEK, MONTH, and YEAR data elements. For example, monthly jobs are scheduled to run based on a 28-day month instead of on a standard month (for example, a month that might end on the 30th).

CA MICS uses MNTHFMT, a SAS format, to provide the full name of a month from a number (for example, December from 12). Modify sharedprefix.MICS.SOURCE(MNTHFMT) if you want to name the 13th month. The MNTHFMT format is recreated every time ALLPGEN or BASPGEN is run.

Checkpoint Size (CKPTCNT):

The CA MICS checkpoint (prefix.MICS.CHECKPT.DATA) is critical to proper CA MICS operation and data validity. By default, the checkpoint tracks up to 100 unique system, subsystem, and product combinations. This default is adequate for most sites and should NOT be changed unless you encounter CA MICS user abend code U410.

Note: There is NO complex-level default for the CKPTCNT parameter. Increasing checkpoint size for one unit database does NOT impact any other unit databases, and there is no risk that a complex-level change will result in unexpected changes to checkpoint size.

If you have exceeded the current checkpoint entry limit, you can increase the size of your checkpoint file up to a maximum of 9999 entries. However, you should first review your CA MICS configuration for alternatives that will improve CA MICS performance while concurrently resolving your checkpoint size constraint.

Checkpoint file size alone has negligible impact on CA MICS daily update performance; however, a large checkpoint file often indicates a heavily loaded unit database and long daily update elapsed times. You can improve daily job performance by splitting your database into two or more database units so that you can perform more of your daily CA MICS processing in parallel. The side benefit of splitting your database is that each of the database unit checkpoint files carries a subset of the original checkpoint entries and thus need not be as large as a single composite checkpoint.

Either of the database splitting techniques described below can be used independently or in conjunction with increasing checkpoint file size. Both can significantly reduce the overall CA MICS update batch window.

o Split processing by product

   The most common approach is to define a separate unit for each CA MICS product or related group of products. In this way, you can concurrently process data for multiple products, reducing overall daily update time.

   This approach is especially recommended for database products such as CICS, IMS, IDMS, and DB2, and network products such as SNT. These products are often characterized by multiple system/subsystem combinations and therefore require multiple checkpoint entries for each system. Quite often, the raw input data for these products (for example, a transaction log tape) is isolated from the data processed by other CA MICS products (for example, the system SMF log).

   Likewise, you may choose to keep products like SMF and RMF in a single unit database because of their shared input data source, the system SMF log.

o Split processing by subsystem

   Once you split CA MICS processing by product, you may find that you can gain additional parallelism by splitting the processing of multiple subsystems of a single product.

   For example, you might define a unit database for selected groups of IMS or CICS processing regions, possibly separating processing for major application areas or departments. If you are processing data from multiple remote locations, you might define a separate unit for each location, letting you parallel SMF and RMF processing for multiple locations.

If, after reviewing your CA MICS configuration, you still need to increase checkpoint size, then you can do so as described in section 5.14, Vertical Checkpoint Expansion.

Input cycles for Audit Archive (AUDITCYC):

This option provides control over the number of cycles that are selected and written to the audit archive within this unit.

The default for this option is 10. This option is most useful if your site executes multiple AUDIT archive jobs within a week, or if you keep fewer than 10 cycles and execute the AUDIT archive job more than 10 days after the start of the week. In these scenarios you could either be writing redundant data, or missing cycles that are no longer online and available at the time the AUDIT archive job executes.

```
+--------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET: Site Characteristics Definition      |
|                                                                          |
| PARMS Library Member is SITE                                             |
|                                                                          |
| Reference Sections: 2.3.2.4                                              |
+--------------------------------------------------------------------------+
|                                                                          |
|                                                                          |
| INSTALLATION NAME:                                                       |
| NAME '__________________________________________________________________'|
|                                                                          |
| YEARS TIMESPAN OPTION:                                                   |
| YEARS TIMESPAN ________ (ACTIVE or INACTIVE)                             |
|                                                                          |
|                                                                          |
| WEEKSTART OPTION:  (default is SUN)                                      |
| WEEKSTART ________ (MON TUE WED THU FRI SAT)                             |
|                                                                          |
|                                                                          |
| 13MONTHYEAR OPTION:  (default is NO)                                     |
| 13MONTHYEAR NO  #DWMY=________                                           |
|             YES ________ ________ ________                               |
|                                                                          |
| CKPTCNT OPTION:  (default is 100)                                        |
| CKPTCNT ____     (100 - 9999)                                            |
|                                                                          |
| AUDITCYC OPTION: (default is 10)                                         |
| AUDITCYC __      (1 - 99)                                                |
|                                                                          |
|                                                                          |
+--------------------------------------------------------------------------+
| ....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+--------------------------------------------------------------------------+
```

Figure 2-12. Site Characteristics Definition Worksheet

2.3.2.5 Performance Group Names (PRFGP) and Service Classes

Beginning with MVS Version 5, and continuing with the z/OS operating system, IBM introduced a method of managing workloads called Goal Mode. With Goal Mode, all workloads are assigned to a Service Class, an eight-byte character name describing the workload. Examples of Service Classes are "PRODTSO" and "TESTCICS". These descriptive names are included in CA MICS files containing information on workloads.

Prior to MVS Version 5, and with the z/OS operating system if running in Compatibility, rather than Goal Mode, different workloads were described by performance group number. The performance group number does not provide descriptive information about the workload assigned to the number.

One of the objectives of the CA MICS database and reporting process is to present information in a form readily understandable to even the casual user of the system. To this end, CA MICS provides a facility to associate a descriptive name, Performance Group Name, to accompany the performance group number in CA MICS database files. (See the PERFGRP data element in the CA MICS data dictionary appendix in the Hardware and SCP Analyzer Guide.) Section 2.3.2.5.1 explains how to make use of this facility.

2.3.2.5.1 Performance Group Names (PRFGP)

Note that this parameter is optional. If you do not want to define performance groups, (for example if your operating system is running in goal mode), simply code an empty member, or use the default member delivered with CA MICS.

Some components, such as the VM Component and SNT Component, do not use performance groups because this information is not provided in the data source. For any SYSID that is exclusively associated with a component that does not use performance group, no entry in the Performance Group Names Table is necessary. The table must be present even though it may contain no entries (i.e., only comments).

In prefix.MICS.PARMS(PRFGP), you are asked to supply character string names which CA MICS can carry in the database along with the corresponding performance group numbers. (This name is carried in the data element named PERFGRPN.)

Figure 2-13 illustrates a sample performance group number-to-name mapping.

Performance Group Number	Performance Group Name
10	'TCAM'
12	'JES2'
13	'TSO FAST'
15	'TSO MED'
20	'BATCH'

Figure 2-13. Sample Performance Group Name Definition

In assigning names to your performance groups, you are dependent on the specifications in your Installation Performance Specifications (IPS), the Installation Control Specifications (ICS--SE-2 level MVS and above), and on whether your shop is careful to run various workloads in the performance groups that are defined for them. Fortunately, this is not a problem in the majority of sites.

The form shown in Figure 2-14 may be used to gather the data you need to complete prefix.MICS.PARMS(PRFGP). In order to complete this form, get a current copy of all the Installation Performance Specifications (IPSs) and

Installation Control Specifications (ICSs--SE2 and above levels of MVS only). They are stored as members of SYS1.PARMLIB with member names of the form IEAIPSxx and IEAICSxx, respectively. With these in hand, you will be able to code prefix.MICS.PARMS(PRFGP) according to the following rules:

One line is to be coded for each performance group.

Comments are coded by beginning the statement with an asterisk (*).

Rather than coding these parameters from scratch, you should modify the sample PRFGP member provided in the prefix.MICS.PARMS library shipped with the system to reflect your definitions. Starting with an existing, correct member may spare you annoying syntax mistakes.

The format of the statements is free-form, but positional. Statements have the following form:

```
sysid nn 'name of performance group nn'
```

The parameters are:

System Identifier (sysid):

The system identifier of the system for which this performance group is being defined. Note that this is the logical SYSID defined for the system in prefix.MICS.PARMS member name SYSID.

PERFORMANCE GROUP NUMBER (0-9999):

The performance group number defined within the Installation Performance Specifications (IPS) for control performance groups and within the Installation Control Specifications (ICS) for report performance groups. Report performance groups are only available with MVS/SE2 and above releases.

PERFORMANCE GROUP NAME:

The one- to eight-character name, enclosed in single quotes, which is carried in the database along with the number to identify a given workload associated with a performance group number.

We recommend that you supply a name for every performance group number which even MIGHT be used. However, this is not mandatory. If CA MICS encounters a performance group number to which no name has been assigned, a name of the form 'GROUP nn' is generated.

A sample PRFGP definition is provided below:

```
*
* PERFORMANCE GROUP DEFINITIONS
*
T033 10 'TCAM'
T033 12 'JES2'
T033 13 'TSO FAST'
T033 15 'TSO MED'
T033 20 'BATCH'
Q033 10 'TCAM'
Q033 12 'JES2'
Q033 13 'TSO FAST'
Q033 15 'TSO MED'
Q033 20 'BATCH'
```

```
+------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET: Performance Group Name Definition  |
|                                                                        |
| PARMS Library Member is PRFGP                                          |
| Reference Sections: 2.3.2.5                                            |
+-----------------------------------+------------------------------------+
|            Performance            |            Performance             |
| Sysid    Group       Group        |  Sysid    Group       Group        |
|          Number      Name         |           Number      Name         |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
| ____      ___       '________'    |  ____      ___       '________'    |
+------------------------------------------------------------------------+
| ....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+------------------------------------------------------------------------+
```

Figure 2-14. Performance Group Name Definition Worksheet

2.3.2.6 Compression (COMPRESS)

About the SAS System Compression Feature

The SAS system gives you the option to create variable-length or fixed-length observations in a SAS data set. Variable-length observations differ from fixed-length observations in that the former are usually smaller because the blank spaces used to pad fixed-length observations are removed.

You instruct SAS to create variable length observations by specifying the COMPRESS= option. SAS data set compression can be implemented for individual data sets or across the entire SAS System. Compression is controlled by the placement of the COMPRESS= option on either an OPTIONS statement (for the entire system) or on a DATA statement (for an individual data set named on the DATA statement).

For more information about the COMPRESS= option, see the SAS Institute documentation.

CA MICS Implementation of Compression

The COMPRESS member in prefix.MICS.PARMS enables you to define which database files will be compressed in this database unit.

The COMPRESS parameter is generated into sharedprefix.MICS.GENLIB(CMPRSDEF). Each database unit's COPYLIBU job adjusts the unit's prefix.MICS.PARMS(COMPRESS) for new products. Once the COMPRESS member FILE statement exists, COPYLIBU will not update it.

For files that can be compressed, the COMPRESS parameter member controls compression specifications for the database unit. You can control, unit by unit, whether or not compression is active, and if it is, to what extent compression is active for each file in the unit.

There is one exception to using the COMPRESS member to compress files. CA MICS Accounting and Chargeback journal files must be compressed using the Journal File Specifications panel (MWF;4;2;3;6). Refer to the CA MICS Accounting and Chargeback User Guide for more information.

The format of the COMPRESS parameter member is:

```
  DBCOMPRESS sw  ALL/ccc  ccc  ...
  IFCOMPRESS ON ccc  ccc  ... -or- OFF
  FILE ccc iiifff  x  d  w  m  y
  FILE ccc iiifff  x  d  w  m  y
  FILE ...
```

Each statement is described below.

DBCOMPRESS Statement

The DBCOMPRESS statement enables you to activate or deactivate the compression facility.

The format of the DBCOMPRESS statement is

```
  DBCOMPRESS  sw  (ALL / ccc  ccc  ...)
```

where

sw is a switch whose values are ON and OFF.

ALL indicates that the value of sw (the switch) applies to
    all products in the unit. ALL is the default.

ccc is a three-character CA MICS product identifier.

When sw is ...	And the next value is ...	Then compression ...
ON	ALL	takes place for all files in the unit.
ON	ccc	takes place for only those files belonging to products in the list.
OFF	ALL	does not take place for any file in the unit.
OFF	ccc	takes place for all files EXCEPT the files belonging to the products in the list. (OFF for that product but ON for the rest of the products in the unit.)

For example, the DBCOMPRESS statement

```
  DBCOMPRESS OFF
```

indicates that compression will not occur for any of the products in this unit, while the DBCOMPRESS statement

```
  DBCOMPRESS OFF SMF
```

indicates that compression will occur for all files in the unit EXCEPT for those belonging to the Batch and Operations Analyzer (ccc=SMF).

If the unit contains only the Batch and Operations Analyzer, the statement

```
  DBCOMPRESS ON SMF
```

is equivalent to the statements

```
  DBCOMPRESS ON
```

and

```
  DBCOMPRESS ON ALL
```

IFCOMPRESS Statement

The IFCOMPRESS statement enables you to activate or deactivate compression for intermediate (work) files for products in this unit that support intermediate file compression.

The format of the IFCOMPRESS statement is

```
  IFCOMPRESS ON ccc ccc ...
  -or-
  IFCOMPRESS OFF
```

```
IFCOMPRESS ON ccc means that intermediate files for the
                   product(s) listed will be compressed

IFCOMPRESS OFF means that intermediate files are not
                   compressed
```

For example, the IFCOMPRESS statement

```
IFCOMPRESS ON CIC
```

indicates that the CICS Analyzer's intermediate files, which include the Loader Domain Global Stats (GLD) and Temporary Storage Global Stats (GTS) files, would be compressed.

The IFCOMPRESS statement does not support the ON ALL qualifier, nor does it provide an exclude capability like that provided with the format "DBCOMPRESS OFF ccc".

FILE Statement

FILE statements are initially copied by the COPYLIBU job from the COMPRESS statements in the cccGENIN member of the sharedprefix.MICS.GENLIB data set. (Once FILE statements exist in prefix.MICS.PARMS(COMPRESS), the COPYLIBU job will not copy them.)

Only products that support compression have FILE statements in prefix.MICS.PARMS(COMPRESS). You can edit FILE statements in prefix.MICS.PARMS(COMPRESS), but you should not add or delete statements. If you add or delete COMPRESS statements in cccGENIN, run sharedprefix.MICS.CNTL(cccCGEN) and then prefix.MICS.CNTL(COPYLIBU) for each unit that contains the ccc product to update the COMPRESS parameter library member.

The format of the FILE statement is

```
 FILE ccc iiifff x d w m y
```

where

ccc the three-character CA MICS product identifier
iii the three-character CA MICS information area identifier
fff the three-character CA MICS file identifier
x through y determine whether or not data in this file's timespan can be compressed when the compression facility is activated.

When ...	Is ...	Then ...	Timespan data ...
x	Y	DETAIL	can be compressed.
x	N	DETAIL	cannot be compressed.
d	Y	DAYS	can be compressed.

d	N	DAYS	cannot be compressed.
w	Y	WEEKS	can be compressed.
w	N	WEEKS	cannot be compressed.
m	Y	MONTHS	can be compressed.
m	N	MONTHS	cannot be compressed.
y	Y	YEARS	can be compressed.
y	N	YEARS	cannot be compressed.

For example, the FILE statement

```
  FILE SMF BATJOB Y Y Y Y Y
```

indicates that data in the DETAIL, DAYS, WEEKS, MONTHS, and YEARS timespans of the Batch User Job Activity file will be compressed when the compression facility is activated.

Chapter 6 of the System Modification Guide includes information about changing compression specifications. Chapter 4 of this guide describes the CA MICS Administrator Utilities available to help you evaluate data compression savings.

2.3.3 CA MICS JCL Planning and Parameters

This section shows you how to specify the JCL-related parameters required to install CA MICS. JCL-related parameters are used by JCLGEN jobs and allow the CA MICS installation and generation processes to customize batch jobs, Job Control Language (JCL) procedures, CA MICS MACROs, and TSO Command Lists (CLISTs) to run in your shop.

JCLGEN jobs are run when installing CA MICS because they perform JCL tailoring. Batch, online, and maintenance facilities that enable CA MICS daily update processing to operate in your environment are also tailored by the parameters you code in this step.

You code these JCL-related parameters in members of the sharedprefix.MICS.PARMS and sharedprefix.MICS.PROTOLIB libraries described in the following sections. The PROTOLIB library only needs to be tailored if you have unusual requirements for JCL tailoring.

We divide the discussion of JCL-related parameters into standard and advanced sections because few users need to customize their systems to the extent that CA MICS is able to support. Call CA MICS Product Support if you think you need to tailor the PROTOLIB library.

NOTE: DO NOT modify JCL or CLISTs in prefix.MICS.CNTL or sharedprefix.MICS.CLIST libraries. ONLY modify JCL or CLISTs using the standard or advanced JCLGEN parameters because system maintenance often requires that you perform a JCLGEN, which recreates the CNTL and CLIST libraries. Changes made to CNTL and CLIST libraries are lost when the JCLGEN is completed.

The following sections explain the JCL generation process, inputs, tables used during JCL generation, and the symbolic language used by the process:

1 - JCLGEN Process Flow
2 - Standard JCLGEN Parameters
3 - JCLGEN Parameters for Special Requirements

2.3.3.1 JCLGEN Process Flow

The CA MICS JCL generation (JCLGEN) process is accomplished by a series of jobs. When run, these jobs create the job streams and JCL procedures needed to install and run CA MICS.

The JCLGEN function is divided into small jobs in order to separate functions which will be performed with greater or lesser frequency. The function of each of these jobs is shown below.

```
 JCLGEN0  - Complex-wide jobs and clists
 JCLGEN1  - CA MICS Accounting and Chargeback specific jobs

 JCLGENA  - Database Unit Installation Jobs - Part 1
            Note:  This job is copied from
                   sharedprefix.MICS.CNTL(JCLGENA) by the
                   COPYLIBS job.  JCLGENA generates the same
                   JCL as JCLGENB and is used only during
                   creation of a new database unit.
 JCLGENB  - Database Unit Installation Jobs - Part 1
 JCLGENC  - Database Unit Installation Jobs - Part 2
 JCLGEND  - Database Unit Operational Jobs
 JCLGENE  - Jobs Affected by a DB Split After Installation
 JCLGENF  - SPECIAL Database Unit Installation Jobs
 JCLGENG  - SPECIAL Database Unit Operational Jobs

 JCLGENUC - Regenerate Complex Jobs from a User Supplied List
 JCLGENU  - Regenerate Unit Jobs from a User Supplied List
```

Note: The consolidated UNITGEN process is a combination of JCL, MACRO, and DATABASE generation processing that consolidates most CA MICS generation processes for a database unit into a single batch process. The UNITGEN job generates (and optionally submits) the UNITGEN1 job to perform, in a single batch process, the JCL generation processing of the JCLGENB, JCLGENC, and JCLGEND jobs, along with the COPYLIBU, ALLPGEN, CYCLEGEN, and GDGSGEN processing for a unit database. See section 5.6 for more information on the UNITGEN process.

The process flow of complex level JCLGEN is illustrated in Figure 2-15.

```
|-------------User Defined PARMS Members-------------| |-Skeleton JCL Members-| |--------CA MICS Definition Tables---------|


                    +-------------+
                    | JCL Options |                       +---------------+           +-------------+    +-------------+
                    | Definition  |                       | Prototype     |           |sharedprefix.|    |sharedprefix.|
                    |sharedprefix.|                       | Library       |           | MICS.GENLIB |    | MICS.ISPTLIB|
                    | MICS.PARMS  |                       | sharedprefix. |           | (COMPTDEF   |    +------+------+
                    | (JCLDEFC)   |                       | MICS.PROTOLIB |           |  DBTABLE)   |           |
                    | "GENLIST"   |                       +-------+-------+           +------+------+           |
                    |  USERJCL    |                               |                          |                  |
                    |  JCLINFO    |                               |                          |                  |
                    |  JCL$Iccc   |                               |                          |                  |
                    |  JCLNAMES   |                               |                          |                  |
                    |  CPLXDEF)   |                               |                          |                  |
                    +------+------+                               |                          |                  |
                           |                                      |                          |                  |
                           |                                      |                          |                  |
                           |                                      |                          |                  |
                           +--------------------------------------+--------------------------+------------------+
                                                                  |
                                                                  |
                                                                  V
                               +--------------+            +------------+
                               | sharedprefix.|            | JCL        |
                               | MICS.SOURCE  |----------->| Generation |
                               | (JCLGEN01    |            | Process    |
                               |  JCLGEN02    |            |            |
                               |  JCLPARM)    |            +------+-----+
                               +--------------+                   |
                                                                  |
                                                                  |
                        +-----------------+---------------+-------+--------+-----------------+--------------+
                        |                 |               |                |                 |              |
                        |                 |               |                |                 |              |
                        V                 V               V                V                 V              V
                 +--------------+ +--------------+ +--------------+ +--------------+ +--------------+ +-------------+
                 | Generated    | | Generated    | | Complex      | | ISPF Table   | | Updated      | | Generated   |
                 | CLISTs       | | Jobs         | | Options      | | Definitions  | | Database     | | Formats     |
                 | sharedprefix.| | sharedprefix.| | sharedprefix.| | sharedprefix.| | Table        | | shareprefix.|
                 | MICS.CLIST   | | MICS.CNTL    | | MICS.MACAUTOS| | MICS.ISPTLIB | | sharedprefix.| | MICS.MCOLIB |
                 +--------------+ +--------------+ +--------------+ +--------------+ | MICS.GENLIB  | +-------------+
                                                                                     |  (DBTABLE)   |
                                                                                     +--------------+
```

Figure 2-15. Complex Level JCLGEN Process Flow

Input consists of the user-modified JCLDEFC, USERJCL, JCLINFO, JCL$Iccc, JCLNAMES, and CPLXDEF members of sharedprefix.MICS.PARMS; all selected (GENLIST) prototype job stream (JCL), procedure, CLIST, and MACRO members from sharedprefix.MICS.PROTOLIB; and the CA MICS-provided definition tables from the sharedprefix.MICS.PARMS, GENLIB, and ISPTLIB libraries. These definition tables contain lists of jobs to be created (JCLGENx and JCL$xccc) along with definitions for data set names (JCLNAMES) and job attribute (JCLINFO and JCL$Iccc) specifications. The default tables cause the creation of all jobs and must be used the first time you do a JCLGEN. See Section 2.3.3.3.2.1, Job/CLIST Generation Lists, for a more complete description of these lists.

JCLGEN may generate any number of job streams (JCL), procedures, CLISTs, MACROs, and ISPF table definitions. These outputs are placed in the CA MICS CLIST, CNTL, MACAUTOS, ISPTLIB, GENLIB, and MCOLIB libraries.

The process flow of a unit level JCLGEN is illustrated in Figure 2-16.

```
 |-------------User-Defined PARMS Members-------------| |-Skeleton JCL Members-| |--------CA MICS Definition Tables------------|

           +--------------+
           | JCL Options  |    +---------------+           +----------------+          +--------------+   +--------------+
           | Definition   |    | JCL Options   |           | Prototype      |          |sharedprefix. |   |sharedprefix. |
           | prefix.      |    | Definition    |           | Library        |          | MICS.GENLIB  |   | MICS.ISPTLIB |
           | MICS.PARMS   |    | sharedprefix. |           | sharedprefix.  |          | (COMPTDEF    |   +-------+------+
           | (JCLDEF      |    | MICS.PARMS    |           | MICS.PROTOLIB  |          |  DBTABLE)    |           |
           | "GENLIST"    |    | (JCL$xccc     |           +-------+--------+          +------+-------+           |
           |  USERJCL     |    | (JCLINFO      |                   |                          |                   |
           |  JCLINFO     |    |  JCL$Iccc)    |                   |                          |                   |
           |  JCLNAMES    |    +------+--------+                   |                          |                   |
           |  DBSPLIT     |           |                            |                          |                   |
           |  SITE        |           |                            |                          |                   |
           |  EXECDEF     |           |                            |                          |                   |
           |  INPUTccc)   |           |                            |                          |                   |
           +------+-------+           |                            |                          |                   |
                  |                   |                            |                          |                   |
                  +-------------------+----------------------------+--------------------------+-------------------+
                                                                   |
                                                                   |
                                                                   V
                                +---------------+           +-------------+
                                | sharedprefix. |           | JCL         |
                                | MICS.SOURCE   |---------->| Generation  |
                                | (JCLGEN01     |           | Process     |
                                |  JCLGEN02     |           |             |
                                |  JCLPARM)     |           +------+------+
                                +---------------+                  |
                                                                   |
                                                                   |
                                                                   |
        +---+------------+-------+---+----+----------------+--------------+--+--------------+----------------+
        |   |            |       |        |                |              |  |              |                |
        |   |            |       |        |                |              |  |              |                |
        V   |            V       |        V                V              V  |              V                V
+--------------+ +--------------+|+--------------+ +--------------+ +--------------+ +--------------+ +--------------+
| Generated    | | Complex      ||| ISPF Table   | | Updated      | | Generated    | | Database     | | Complex      |
| Jobs         | | Options      ||| Definitions  | | Database     | | Formats      | | Configuration| | Options      |
| prefix.      | | sharedprefix.||| sharedprefix.| | Table        | | shareprefix. | | prefix.      | | sharedprefix.|
| MICS.CNTL    | | MICS.MACAUTOS||| MICS.ISPTLIB | | sharedprefix.| | MICS.MCOLIB  | | MICS.PARMS   | | MICS.SOURCE  |
+--------------+ +--------------+|+--------------+ | MICS.GENLIB  | +--------------+ | (DBCONFIG)   | | (#CPLXLST)   |
            |                    |                 |  (DBTABLE)   |          |       +--------------+ +--------------+
            |                    |                 +--------------+          V
            V                    V                                    +--------------+
      +--------------+  +-------------------+                         | Generated    |
      | Generated    |  | Generated         |                         | Formats      |
      | CLISTs       |  | Unit MACROs       |                         | prefix.      |
      | sharedprefix.|  | prefix.           |                         | MICS.MUOLIB  |
      | MICS.CLIST   |  | MICS.USER.SOURCE  |                         +--------------+
      +--------------+  +-------------------+
```

Figure 2-16. Unit Level JCLGEN Process Flow

Input consists of the user-modified JCLDEF, USERJCL, JCLINFO, JCLNAMES, DBSPLIT, SITE, EXECDEF, and the INPUTccc members of prefix.MICS.PARMS; all JCL$xccc GENLIST and all JCLINFO and JCL$Iccc members of sharedprefix.MICS.PARMS; all selected (GENLIST) prototype job stream (JCL), procedure, CLIST, and MACRO members of sharedprefix.MICS.PROTOLIB; and the CA MICS-provided definition tables from sharedprefix.MICS.PARMS, sharedprefix.MICS.GENLIB, and sharedprefix.MICS.ISPTLIB. These definition tables contain lists of jobs to be created (JCLGENx and JCL$xccc) along with definitions for data set names (JCLNAMES), job attributes (JCLINFO and JCL$Iccc), database split (DBSPLIT), site characteristics (SITE), and dynamic execution options (EXECDEF) specifications. The default tables cause the creation of all jobs and must be used the first time you do a JCLGEN. See Section 2.3.3.3.2.1, Job/CLIST Generation Lists, for a more complete description of these lists.

JCLGEN may generate any number of job streams (JCL), procedures, CLISTs, MACROs, CA MICS parameter members, and ISPF table definitions. These outputs are placed in the sharedprefix.MICS CLIST, CNTL, MACAUTOS, ISPTLIB, GENLIB, MCOLIB, and SOURCE libraries. Output is also placed in the prefix.MICS USER.SOURCE, MUOLIB, and PARMS libraries.

2.3.3.1.1 Processing Done by Standard JCLGEN Jobs

The output produced by each of the JCLGEN jobs is stored in the CA MICS CLIST, CNTL, GENLIB, ISPTLIB, MACAUTOS, PARMS, and SOURCE libraries. The default generation list for the job is in sharedprefix.MICS.PARMS(JCLGENx) where 'x' is 0 through 2, or prefix.MICS.PARMS(JCLGENx) where 'x' is B through G.

Figures 2-17, 2-18, and 2-19 list the CLISTs and jobs which will be generated by each of the JCLGEN jobs when the default generation lists are used.

```
+--------------------------------------------------------+
|  Jobs:  Added to sharedprefix.MICS.CNTL by Job JCLGEN0 |
|         For Complex-wide Jobs                          |
|--------+-----------------------------------------------|
|cccCGEN | Component Generation Job for Component "ccc"  |
|CICASM  | Assembly for CICS Analyzer Assembler Programs |
|COPYLIBS| Job to Copy CA MICS Lib's for a New DB Unit   |
|CPLXGEN | Complex Option Generation Job                 |
|DOCAGEN | Document Access Guide/Title Generation Job    |
|DOCAGEN1| Document Access Full Generation Job           |
|DOCUMENT| CA MICS Document Print                        |
|GDGSGENC| Establish Complex-Level Generation Data Groups|
|JCLGEN1 | JCLGEN of Installation Accounting Jobs        |
|JCLGEN2 | JCLGEN of I/S Management Alert Jobs           |
|JCLGENB | Model JCLGENB Job Copied to New Unit Libraries|
|JCLGENUC| JCLGEN of User Defined Jobs                   |
|LDSASFLS| Job to Load SAS Tables and Screens            |
|MDBCPYRN| Job to Expand the MCOLIB Data Set             |
|MDBGEN  | Job to Re-Generate the Platform Meta Database |
|MQRHOST | Job for Q&R Query Host Processing             |
|PROCJOB1| Job to Move Complex-Level PROCs to PROCLIB    |
|PSPSTAT | Generate Product Change Status Report         |
|TNGCBCSV| Consolidate TNGCB SAS Data Sets into 1 CSV    |
|USERFMTC| Update Complex-Level User-Defined SAS Formats |
|--------+-----------------------------------------------|
|   CLIST Members                                        |
|--------+-----------------------------------------------|
|MWFSMPL1| Sample MWF CLIST w/o Installation ISPLLIB     |
|MWFSMPL2| Sample MWF CLIST with Installation ISPLLIB    |
+--------------------------------------------------------+
+--------------------------------------------------------+
|  Jobs:  Added to sharedprefix.MICS.CNTL by Job JCLGEN1 |
|         For CA MICS Accounting and Chargeback Jobs     |
|--------+-----------------------------------------------|
|ACTBGTL0| Build ISPF Table From ACTBGT00 File Job       |
|ACTBGTL1| Build ISPF Table From ACTBGT01 File Job       |
|ACTBGTSM| Budget Summary Job                            |
|ACTBGT46| Convert Pre4600 ACTBGT Files                  |
|ACTCCALD| Build ISPF Table From ACTCCA01 File Job       |
|ACTCOSTR| Cost Reconciliation Report Job                |
|ACTCRDPT| Supplementary Credit Report Job               |
|ACTEXTTB| Accounting External Files Job                 |
|ACTJC   | ISPF/DM Table Utility Update Job              |
|ACTJT   | ISPF/DM Table Utility Update Job              |
|ACTMG   | Accounting Central Site Merge Job             |
|ACTMS   | Remote Site Unload Job                        |
|ACTRCP46| Convert Pre4600 ACTRCP Files                  |
|ACTSEQLD| Rate Simulation Job                           |
|ACTTBUTL| ISPF Table Maintenance Job                    |
|ACTUPDTB| Update Tables Job                             |
|ACTUTILG| Utilities Generation Job                      |
|ACTXBGT0| ACTBGT00 SAS File Generation Job              |
|ACTXBGT1| ACTBGT01 SAS File Generation Job              |
|ACTXCCA | ACTCCA01/ACTCNV01 SAS File Generation Job     |
|ACTXCCC | Cost Center Validation Table Gen Job          |
|ACTXOPTM| Shared Level Multi-site Acct Options Gen Job  |
|ACTXOPTS| Shared Level Standard Acct Options Gen Job    |
|ACTXSEQ | ACTSEQ01 SAS File Generation Job              |
|ACTXTBL | SAS File Generation Job                       |
|ACTYRBGT| Budget Projection Report Job                  |
|ACTYTDRF| Retrofit ACT_YR00 Invoice Level Job           |
|ACT0YR46| Convert Pre4600 ACT_YR Files                  |
|CHRGELEM| Charging Element Table Generation Job         |
|CHRGROUP| Charge Group Generation Job                   |
|CLOSETBL| Tables Close Job                              |
|CREDCAT | Credit Category Generation Job                |
|EXTFILE | External File Generation Job                  |
|FINRECAP| Financial Recap File Creation Job             |
|FINRPT  | Financial Report Job                          |
|FINRPTY | Year-to-date Financial Report Job             |
|INVCCAT | Invoice Category Generation Job               |
|INVOICE | Invoice Job                                   |
|INVOICEM| Multi-site Invoice Job                        |
|INVOICER| Invoice Report Job                            |
|INVOICET| Test Invoice Job                              |
|MNACTHLR| Higher Level Report Job                       |
|PRINTTBL| Tables Print Job                              |
|PRORATE | Prorating Generation Job                      |
|RATERPTE| External File Rate Report Job                 |
+--------------------------------------------------------+

+--------------------------------------------------------+
|  Jobs:  Added to sharedprefix.MICS.CNTL by Job JCLGEN2 |
|         For CA MICS I/S Management Alert Specific Jobs |
|--------+-----------------------------------------------|
|CIMALLOC| Allocate CIMANAGE Data Base Job               |
|CIMDSTUB| Appendix E Management Indicator Derivations   |
|CIMFTTB | Table Generations Job                         |
|CIMINFLS| CIMINFLS Table Generator Job                  |
|CIMINRPT| Management Reports from Appendix E            |
|CIMISDOC| Information Structure Documentation Job       |
|CIMMIDOC| Management Indicator Documentation Job        |
|CIMRPTGN| Reports Job                                   |
|SPCIMUPD| Monthly CIMANAGE Update Job                   |
+--------------------------------------------------------+
```

Figure 2-17. JCLGEN0-2 Generated Outputs

At the database unit level, the JCLGENA/JCLGENB and JCLGENC through JCLGENG jobs provide JCL generation processing as described in the tables that follow.

The consolidated UNITGEN process combines the JCL generation processing of the JCLGENB, JCLGENC, and JCLGEND jobs, with the COPYLIBU, ALLPGEN, CYCLEGEN, and GDGSGEN jobs. The UNITGEN job generates (and optionally submits) the UNITGEN1 job to perform the consolidated generation processing for a unit database. See section 5.6 for more information on the UNITGEN process.

```
+-+--------------------------------------------------------+
| |  Jobs:   Added to prefix.MICS.CNTL by Jobs JCLGENA     |
| |          and/or JCLGENB                                |
| |          For Data Base Unit Installation Jobs - part 1 |
+-+--------+-----------------------------------------------|
|x|cccCHECK| Data Base Definition Check Job for Component  |
|x|cccEVA  | Exception Evaluation Analysis for Component   |
|x|cccPGEN | Parameter Generation Job for Component        |
|x|ACTDAY1R| ACT Daily Audit Tape Recovery Job             |
|x|ACTINITA| ACT DAY1 Audit Tape Initialization Job        |
|x|ACTLGRSP| ACT Monthly Ledger Prorating Job              |
|x|ACTJU   | ISPF/DM Table Update Utility Job              |
|x|ACTRTSIM| ACT Rate Simulation of Charges Job            |
| |ALLPGEN | Parameter Generation Job for All Components   |
|x|AUDITRPT| ACT Component Audit Report Job                |
|x|BGTCNTL | ACT Component Budget Job                      |
|x|CICALONE| CICS Standalone Processing Program            |
|x|CICCDCPT| CICS Data Dictionary Print Program            |
|x|CICILIST| CICS Input Data List Program                  |
| |CKPTINIT| Checkpoint File Initialization Job            |
| |COPYLIBU| Job to Tailor CA MICS Lib's  for a New DB Unit|
| |CYCLEGEN| Data Base SAS MACRO Generator Job             |
| |DBPLAN  | Data Base Space Analysis Job                  |
|x|FINRPTU | ACT Component Financial Report Job            |
|x|IMSASM  | Job to Reassemble IMSLOGUT                    |
|x|INVOICEU| ACT Component Invoice Job                     |
| |JCLGENC | JCLGEN of Jobs Affected by Data Base Split    |
| |JCLGEND | JCLGEN of Operational Jobs                    |
| |JCLGENE | JCLGEN of Jobs Affected by Split After Install|
| |JCLGENF | JCLGEN of Data Base Backload/Special Jobs     |
| |JCLGENU | JCLGEN of User Defined Jobs                   |
|x|MAGRUN  | ACT CA MICS Algorithm Generator Job           |
| |PROCJOB2| Job to Move Operational Procs to PROCLIB      |
|x|PRORATEU| Unit Level Prorate Format Generation Job      |
|x|RATERPT | Unit Level Rate Report Job                    |
|x|RMFRPTS | RMF Post Processor Report Job                 |
| |RSTATUS | Run Status Report Job                         |
|x|SNTRPTSD| SNA Network Daily Tabular Reports             |
|x|SNTRPTSM| SNA Network Mnthly Tabular Reports            |
+-+--------+-----------------------------------------------+
```

NOTE: Column 1 in Figure 2-18 indicates those jobs which are only generated by JCLGENA and JCLGENB if their Component is defined in the COMPONENTS statement in the JCLDEF PARMS member.

Figure 2-18. JCLGENA and JCLGENB Generated Job Outputs

```
+--------------------------------------------------------+
| Jobs:    Added to prefix.MICS.CNTL by Job JCLGENC      |
|          For Database Unit Installation Jobs - part 2  |
+--------+-----------------------------------------------+
|ALLOCATE| Database Unit Allocation Job                  |
|GDGSGEN | Job to Establish Generation Data Groups       |
|PROCJOB3| Job to Move PROCs to PROCLIB                  |
|cccIUALC| ccc Product Incremental Update Allocation Job |
|cccIUGDG| ccc Product Job to Establish Optional         |
|        | Incremental Update Tape Data Groups           |
+--------+-----------------------------------------------+
```

```
+--------+-----------------------------------------------+
| Jobs:    Added to prefix.MICS.CNTL by job JCLGEND      |
|          For Database Unit Operational Jobs            |
+--------+-----------------------------------------------|
|AUDIT   | Optional Stand-Alone Archive Audit Job        |
|BACKUP  | Database Backup Job                           |
|DAILY   | Standard Daily Update and Reporting Job       |
|DAILYRPT| Special Daily Reporting Job                   |
|DAYSMFR | Daily DAYSMF Work File Recovery               |
|EXCFULL | Full List of Exception Occurrences            |
|HISTM   | Optional Monthly Archive History Job          |
|HISTW   | Optional Weekly Archive History Job           |
|INCRccc | ccc Product Incremental Update Job            |
|IUDBINIT| Incremental Update Database Initialization Job|
|MONTHACT| Special Accounting Dates Monthly Update Job   |
|MONTHLY | Standard Monthly Update and Reporting Job     |
|MONTHRPT| Standard Monthly Reporting Job                |
|RESTORE | Database Restore Job                          |
|RSTRTLIB| sharedprefix.MICS.ISPTLIB Restore Job         |
|RSTRTBLS| sharedprefix.MICS.TABLES Restore Job          |
|SCHEDULE| Job Scheduling Facility Job                   |
|SPLITSMF| Incremental Update DAYSMF Processing Job      |
|WEEKLY  | Standard Weekly Update and Reporting Job      |
|WEEKRPT | Standard Weekly Reporting Job                 |
|YEARACT | Special Accounting Dates Yearly Update Job    |
|YEARLY  | Standard Yearly Update and Reporting Job      |
+--------+-----------------------------------------------+
+--------------------------------------------------------+
|  Jobs:   Added to prefix.MICS.CNTL by Job JCLGENE      |
|          For Jobs Affected by a DB Split after Install |
|--------+-----------------------------------------------|
|ALLOCATE| Database Unit Allocation Job                  |
|BACKUP  | Database Backup Job                           |
|GDGSGEN | Job to Establish Generation Data Groups       |
|MONTHLY | Standard Monthly Update and Reporting Job     |
|PROCJOB3| Job to Move PROCs to PROCLIB                  |
|RESTORE | Database Restore Job                          |
+--------------------------------------------------------+

+--------------------------------------------------------+
|  Jobs:   Added to prefix.MICS.CNTL  by Job JCLGENF     |
|          For SPECIAL Database Unit Installation Jobs   |
|--------+-----------------------------------------------|
|ALLOCATE| Special Database Allocation Job               |
|ALLPGEN | Job to Generate Special Database Components   |
|CKPTINIT| Special Database Checkpoint Init Job          |
|COPYLIBU| Job to Tailor MICS Libraries for a New DB Unit|
|CYCLEGEN| Special Database Cycles Generation job        |
|GDGSGEN | Special Database GDG Job                      |
|JCLGENG | JCLGEN Job for Special DB Operational Jobs    |
|KILL    | Special Database Deletion Job                 |
|PROCJOB2| Special Database Job to Move PROCs            |
|PROCJOB3| Special Database Job to Move PROCs            |
|SELFILES| Special Database Job to Select Files          |
+--------------------------------------------------------+
+--------------------------------------------------------+
|  Jobs:   Added to prefix.MICS.CNTL  by Job JCLGENG     |
|          For SPECIAL Database Unit Operational Jobs    |
|--------+-----------------------------------------------|
|AUDIT   | Optional Stand-Alone Archive Audit Job        |
|BACKUP  | Special Database Backup Job                   |
|DAILY   | Special Database Daily Job                    |
|INCRccc | Special Database ccc Product Incremental      |
|        | Update Job                                    |
|IUDBINIT| Special Database Incremental Update           |
|        | Initialization Job                            |
|MONTHLY | Special Database Monthly Job                  |
|RESTORE | Special Database Restore Job                  |
|SPLITSMF| Special Database Incremental Update DAYSMF    |
|        | Processing Job                                |
|WEEKLY  | Special Database Weekly Job                   |
|YEARLY  | Special Database Yearly Job                   |
+--------------------------------------------------------+
```

Figure 2-19. JCLGENC-G Generated Outputs

2.3.3.2 Standard JCLGEN Parameters

Because installation standards and practices in the area of Job Control Language (JCL) and TSO Command (CLIST) language vary widely, CA MICS provides a facility for tailoring the JCL and CLISTs it uses.

Though experience has shown that the parameters supported in the members of MICS.PARMS discussed in the following sections provide sufficient flexibility for almost all CA MICS users, if you cannot meet your requirements using these members, more flexible (and complex) support exists and is documented in section 2.3.3.3, JCLGEN Parameters for Special Requirements. These include:

- o A data volume so great that certain database data sets (e.g., MICS.DETAIL) have to be split up into multiple OS data sets to supply the required DASD space.
- o A desire to use a different level of SAS for database inquiry than the ones supported for use by the CA MICS Update Process.
- o A desire to insert certain DD statements into the CA MICS database inquiry procedures (e.g.,//SYSUDUMP).
- o A desire to change the time and lines limits for certain CA MICS jobs.
- o Data set naming standards different from those of CA MICS (e.g., if you HAD to name sharedprefix.MICS.SOURCE sharedprefix.MICS.PROGRAMS).
- o A need to have the CA MICS PROCs and CLISTs (which allocate SORTWKxx files) allocate either more of them or fewer of them.
- o A desire to have all JCL control for multiple CA MICS databases centralized in one library (MICS.PROTOLIB) rather than being distributed across many (MICS.PARMS).
- o Any other modification which cannot be accomplished via the standard JCLGEN parameters discussed below.

The JCLGEN parameters exist in members in the CA MICS parameter library (MICS.PARMS). You must complete them prior to executing the CA MICS JCL generation process. JCLGEN parameters fall into three categories:

- o The JCLDEF member is by far the most complex containing all of the standard JCLGEN parameter statements. This parameter must be coded.

- o The USERJCL member is used to supply a JOB statement (optional) and any JES control statements that must come after the JOB statement (required), in order for jobs to run in your shop. This parameter must be coded.

- o The INPUTccc members supply the DD statements to be used to read the raw measurement data. For more information about these optional members, see the data integration product guides.

The following sections describe how to complete these parameters:

1 - JCL Option Definitions (JCLDEF)
2 - Job and JES Definitions (USERJCL)
3 - Input DD Statements for SMF (INPUTRDR)

2.3.3.2.1 JCL Option Definitions (JCLDEF)

There are a few required parameters and many optional ones, which you specify in prefix.MICS.PARMS(JCLDEF), to guide the JCLGEN process as it generates CLISTs, MACROs, JCL procedures, and job streams tailored for your processing environment. This member contains one statement for each parameter defined. The format of the statements is free-form but positional. Keywords cannot be abbreviated. Any line starting with an asterisk is considered a comment. Only parameters that contain embedded or trailing blanks require beginning and ending single quotes. (When in doubt, add quotes.) A sample member is illustrated in Figure 2-20 and the JCL Options worksheet is illustrated in Figure 2-21.

You should use the JCLDEF member in the prefix.MICS.PARMS library shipped with the CA MICS system as a starting point for specifying your options. These options are described in the following sections:

1 - Database Unit Control Definitions
2 - Database Unit Library Definitions
3 - Database Unit Execution Option Definitions
4 - Database Unit JCL Definitions
5 - Database Unit Tape Specifications
6 - Database Unit Extended JOB Statement Definitions
7 - Database Unit User Defined Symbols
8 - JCLGEN Parameter Sharing

Note: Many of the parameter values defined in this member are available as internal JCLGEN variables. For more information, see section 2.3.3.3.3.1, Generation Control Language Variables.

```
*
* SAMPLE PREFIX.MICS.PARMS(JCLDEF)
*       CONTROL DEFINITIONS
*
 COMPLEXPARMS   YES
 DATABASE       CPMPROD      P   PRIMARY
 COMPONENTS     RMF SMF TSO CIC IDM SNT IMS DB2 SRL VMC
 COMPONENTS     VCA ACT CAP MQS WEB TDM AST AUM
 SMFRECORDING   CIC IDM SNT VCA
 PREFIX         DPM981.DB1 MICSLEVEL
 TAPEPREFIX     TPM981.DB1 MICSLEVEL
*
*       LIBRARY DEFINITIONS
*
```

```
*TSMLINK       'TSOMON.R5.TSMLINK'
*TSMPARMS      'TSOMON.R5.TSMMAC'

*
*       EXECUTION OPTION DEFINITIONS
*
 ARCHIVE        AUDIT YES JOB  AUTOSUBMIT
 ARCHIVE        HISTM YES JOB  AUTOSUBMIT
 ARCHIVE        HISTW NO
 BACKUP         FREQ  DAILY
 DAYSMF         FILES TEMPORARY
*DAYSMF         EXCLUDE DB2
*SMFDIRECTOR    DB2
 EXCLUDESTEP    TIMESPAN STEPNUMBER
 MONTHLY        BACKUP YES
 RESTORE        BACKUP YES
*
*       JCL DEFINITIONS
*
 AUTOSUBMIT     YES
 COLIBSPACE     6144 1500 500
 DASDUNIT       SYSDA
 DASDSMSPARMS   STORCLAS=MICSDATA
 DIRSTATS       NO
*DYNAMWAIT      15
*INTRDRCLASS
 JES            2
 LOGS           *
 MICSDMDELIM    ~
*MICSLOG        *
 RENUMBER       YES
 REPORTS        *
 SORTCYL        6
 SORTUNIT       SYSDA
 SORTSMSPARMS   STORCLAS=WORK
 SORTUTILITY    CASORT
 SORTWKNO       6
 STEPCPULIMIT   YES
*UOLIBSPACE     6144 500 100
 WORKSPACE      TRK 500 250
 WORKUNIT       SYSDA
 WORKSMSPARMS   STORCLAS=MICSTEMP
*WORKVOLSER

*
*       TAPE SPECIFICATIONS
*
 TAPEPARM       UNIT=3480 EXPDT=1999/000 COMP
 TAPEPARM       NULLUNIT=3480 NULLSER=NULSER
```

```
*TAPEPARM        VOLCOUNT=nn
*TAPESMSPARM
 ACCTDTAPEPARM
 ACCTDTAPESMSPARM
 ACCTDAY1PARM
 ACCTDAY1SMSPARM
 ACCTDAY2PARM
 ACCTDAY2SMSPARM
 ACCTMTAPEPARM
 ACCTMTAPESMSPARM
*
 AUDITPARM       GDG=53
*AUDITPARM       UNIT=3480 EXPDT=1999/000 COMP
*AUDITPARM       NULLUNIT=3480 NULLSER=NULSER
*AUDITPARM       VOLCOUNT=nn
*AUDITSMSPARM
*
 BKUPPARM        GDG=10
*BKUPPARM        UNIT=3480 EXPDT=1999/000 COMP
*BKUPPARM        NULLUNIT=3480 NULLSER=NULSER
*BKUPPARM        VOLCOUNT=nn
*BKUPSMSPARM
*
 HISTMPARM       GDG=5
*HISTMPARM       UNIT=3480 EXPDT=1999/000 COMP
*HISTMPARM       NULLUNIT=3480 NULLSER=NULSER
*HISTMPARM       VOLCOUNT=nn
*HISTMSMSPARM
*
 HISTWPARM       GDG=7
*HISTWPARM       UNIT=3480 EXPDT=1999/000 COMP
*HISTWPARM       NULLUNIT=3480 NULLSER=NULSER
*HISTWPARM       VOLCOUNT=nn
*HISTWSMSPARM
*
 MBKUPPARM       GDG=6
*MBKUPPARM       UNIT=3480 EXPDT=1999/000 COMP
*MBKUPPARM       NULLUNIT=3480 NULLSER=NULSER
*MBKUPPARM       VOLCOUNT=nn
*MBKUPSMSPARM
*
 ACCTDGDG        12
*DENSITY
 TMS             YES
 VCATNAME        SYSTEM
*VCATOWNER
*VCATPASSWORD
*VCATSER
*VCATUNIT
```

```
 SYSTEMMODEL   'DPM981.DB1.MODEL'
*
*       EXTENDED JOB STATEMENT DEFINITIONS
*
 ACCT          'MICS'
 CLASS          A
 GROUP          XPM981
*JCLLIB         NO
 JOBNAME        DPM
*JOB7380
 MSGCLASS       A
 MSGLEVEL       1,1
 NOTIFY         &SYSUID
*PASSWORD
*PERFORM
 PROG          'MICS'
*PRTY
*REGION         4096
*USER           DPM981
*
*     USER DEFINED SYMBOLS
*
*USERDEF USR_____  __________________________________________
```

Figure 2-20. Sample JCLDEF Member

```
                          +---------------------------------------------------------------------------+
                          | INSTALLATION PREPARATION WORKSHEET: JCL Options Definitions (Part 1)      |
                          |                                                                           |
                          | PARMS Library Member is JCLDEF  -   Reference Sections: 2.3.3.2.1         |
                          +---------------------------------------------------------------------------+
Control Definitions       |*COMPLEXPARMS _____                   (YES or NO)                          |
                          | DATABASE   ________   _  ________                                         |
                          | COMPONENTS ____ ____ ____ ____ ____  (RMF, SMF, TSO, etc.)                |
                          | SMFRECORDING    ____ ____ ____       (AST, CIC, IDM, HSM, SNT, VCA, WEB)  |
                          | PREFIX       _______________         (MICSLEVEL or NOMICSLEVEL)           |
                          | TAPEPREFIX   _______________         (MICSLEVEL or NOMICSLEVEL)           |
                          |                                                                           |
Library Definitions       |*ISPLLIB1    '____________________________________________'                |
                          |*ISPLLIB2    '____________________________________________'                |
                          |*ISPMLIB1    '____________________________________________'                |
                          |*ISPMLIB2    '____________________________________________'                |
                          |*ISPTLIB1    '____________________________________________'                |
                          |*ISPTLIB2    '____________________________________________'                |
                          |*PROCLIB     '____________________________________________'                |
                          |*SASAUTOS    '____________________________________________'                |
                          |*SASCONFIG   '____________________________________________'                |
                          |*SASCONFIG1  '____________________________________________'                |
                          |*SASCONFIG2  '____________________________________________'                |
                          |*SASCONFIG3  '____________________________________________'                |
                          |*SASCONFIG4  '____________________________________________'                |
                          |*SASHELP     '____________________________________________'  (optional)    |
                          |*SASLIB      '____________________________________________'                |
                          |*SASMSG      '____________________________________________'                |
                          |*SASNAME      __________                                                   |
                          |*SASVER       __________                                                   |
                          |*SORTLIB     '____________________________________________'                |
                          |*STEPLIB1    '____________________________________________'  (optional)    |
                          |*STEPLIB2    '____________________________________________'  (optional)    |
                          | TSMLINK     '____________________________________________'                |
                          | TSMPARMS    '____________________________________________'                |
                          |*USEROBJLIB  '____________________________________________'  (optional)    |
                          |                                                                           |
Execution Option          | ARCHIVE     AUDIT  ___ ____ ____________ (YES/NO STEP/JOB AUTOSUBMIT)    |
  Definitions             | ARCHIVE     HISTM  ___ ____ ____________ (YES/NO STEP/JOB AUTOSUBMIT)    |
                          | ARCHIVE     HISTW  ___ ____ ____________ (YES/NO STEP/JOB AUTOSUBMIT)    |
                          | BACKUP      FREQ   ________  (DAILY, BI-DAILY, or WEEKLY)                 |
                          | DAYSMF      OFF                                                           |
                          | DAYSMF      FILES  _________ (TEMPORARY or PERMANENT)                     |
                          | DAYSMF      EXCLUDE ___ ___ ___ ___ (RMF, SMF, TSO, etc.)                 |
                          | SMFDIRECTOR ___ ___ ___ ___  (RMF, SMF, TSO, etc.)                        |
                          | EXCLUDESTEP _______ ___      (TIMESPAN STEPNUMBER)                        |
                          | MONTHLY     BACKUP ___       (YES or NO)                                  |
                          | RESTORE     BACKUP ___       (YES or NO)                                  |
                          |                                                                           |
                          +---------------------------------------------------------------------------+
```

Figure 2-21. JCL Options Definitions Worksheet (Part 1 of 6)

```
                          +--------------------------------------------------------------------------+
                          | INSTALLATION PREPARATION WORKSHEET: JCL Options Definitions (Part 2)     |
                          | PARMS Library Member is JCLDEF  -   Reference Sections: 2.3.3.2.1        |
                          +--------------------------------------------------------------------------+
JCL Definitions           | AUTOSUBMIT    ___         (YES or NO)                                    |
                          |*COLIBSPACE    ___ ___ ___ (CYL or TRK, primary, secondary)               |
                          | DASDUNIT      ________    (generic name)                                 |
                          | DASDSMSPARMS  ________    (SMS data set allocation parameters)           |
                          |*DIRSTATS      ___         (YES or NO)                                    |
                          |*DYNAMWAIT     ____        (# minutes to wait for unavailable data sets)  |
                          | INTRDRCLASS   ___         (default A)                                    |
                          | JES           _           (2 or 3)                                       |
                          | LOGS          __________  (SYSOUT class parameters - class,program,form) |
                          |*MICSDMDELIM   _           (default ~)                                    |
                          | MICSLOG       __________  (SYSOUT class parameters - class,program,form) |
                          | RENUMBER      ___         (YES or NO)                                    |
                          | REPORTS       __________  (SYSOUT class parameters - class,program,form) |
                          | SORTCYL       __          (up to 999 cylinders)                          |
                          | SORTUNIT      ________    (generic name)                                 |
                          | SORTSMSPARMS  ________    (SMS data set allocation parameters)           |
                          |*SORTUTILITY   ________    (sort product)                                 |
                          |*SORTWKNO      ________    (number of sort work data sets - 0 to 99)      |
                          |*STEPCPULIMIT  ___         (YES or NO)                                    |
                          | UOLIBSPACE    ___ ___ ___ (CYL or TRK, primary, secondary)               |
                          | WORKSPACE     ___ ___ ___ (CYL or TRK, primary, secondary)               |
                          | WORKUNIT      ________    (generic name)                                 |
                          | WORKVOLSER    ______      (volume name)                                  |
                          | WORKSMSPARMS  ________    (SMS data set allocation parameters)           |
                          |                                                                          |
Tape Specifications       | TAPEPARM                  (one or more TAPEPARM parameters)              |
                          |          UNIT=________    (generic name)                                 |
                          |         EXPDT=_______     (YYDDD or YYYY/DDD)                            |
                          |       NOEXPDT             (omit both EXPDT= and RETPD= from DD stmts.)   |
                          |        NOCOMP vs. COMP    (no tape data compaction -- IDRC=NOCOMP        |
                          |                            tape data compaction -- IDRC=COMP)            |
                          |       NOSTACK vs. STACK   (non-stacked tapes = 1 file per tape volume;   |
                          |                            default STACK = multiple files per tape vol. )|
                          |         NOAFF vs. AFF     (no UNIT=AFF on DD stmts. = 1 unit per file    |
                          |                            default AFF = specify UNIT=AFF on DD stmts.)  |
                          |      VOLCOUNT=___         (default 5)                                    |
                          |      NULLUNIT=________    (generic name)                                 |
                          |       NULLSER=______      (dummy volume serial number)                   |
                          |         MODEL='___________________________________________'              |
                          | TAPESMSPARM   ________    (SMS data set allocation parameters)           |
                          | ACCTDTAPEPARM ________    (Accounting parameters)                        |
                          | ACCTDTAPESMSPARM _____    (SMS data set allocation parameters)           |
                          | ACCTDAY1PARM  ________    (Accounting parameters)                        |
                          | ACCTDAY1SMSPARM  _____    (SMS data set allocation parameters)           |
                          | ACCTDAY2PARM  ________    (Accounting parameters)                        |
                          | ACCTDAY2SMSPARM  _____    (SMS data set allocation parameters)           |
                          | ACCTMTAPEPARM ________    (Accounting parameters)                        |
                          | ACCTMTAPESMSPARM _____    (SMS data set allocation parameters)           |
                          +--------------------------------------------------------------------------+
```

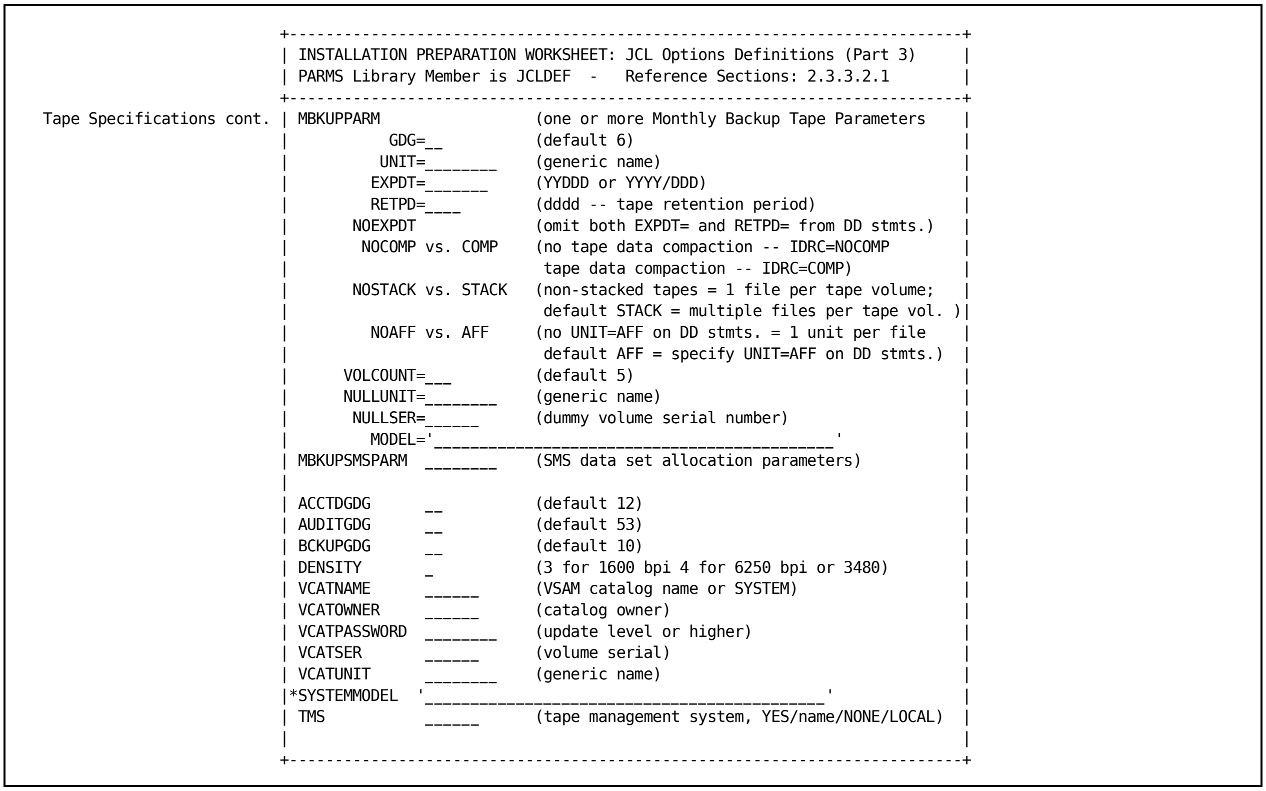
Figure 2-21. JCL Options Definitions Worksheet (Part 2 of 6)

```
                          +--------------------------------------------------------------------------+
                          | INSTALLATION PREPARATION WORKSHEET: JCL Options Definitions (Part 3)     |
                          | PARMS Library Member is JCLDEF  -   Reference Sections: 2.3.3.2.1        |
                          +--------------------------------------------------------------------------+
Tape Specifications cont. | AUDITPARM                  (one or more Archive Audit Parameters          |
                          |           GDG=__           (default 53)                                   |
                          |          UNIT=________    (generic name)                                 |
                          |         EXPDT=_______     (YYDDD or YYYY/DDD)                            |
                          |         RETPD=____        (dddd -- tape retention period)                |
                          |       NOEXPDT             (omit both EXPDT= and RETPD= from DD stmts.)   |
                          |        NOCOMP vs. COMP    (no tape data compaction -- IDRC=NOCOMP        |
                          |                            tape data compaction -- IDRC=COMP)            |
                          |       NOSTACK vs. STACK   (non-stacked tapes = 1 file per tape volume;   |
                          |                            default STACK = multiple files per tape vol. )|
                          |         NOAFF vs. AFF     (no UNIT=AFF on DD stmts. = 1 unit per file    |
                          |                            default AFF = specify UNIT=AFF on DD stmts.)  |
                          |      VOLCOUNT=___         (default 5)                                    |
                          |      NULLUNIT=________    (generic name)                                 |
                          |       NULLSER=______      (dummy volume serial number)                   |
                          |         MODEL='____________________________________________'             |
                          | AUDITSMSPARM  ________    (SMS data set allocation parameters)           |
                          |                                                                          |
                          |  BKUPPARM                  (one or more Backup Tape Parameters            |
                          |           GDG=__           (default 10)                                   |
                          |          UNIT=________    (generic name)                                 |
                          |         EXPDT=_______     (YYDDD or YYYY/DDD)                            |
                          |         RETPD=____        (dddd -- tape retention period)                |
                          |       NOEXPDT             (omit both EXPDT= and RETPD= from DD stmts.)   |
                          |        NOCOMP vs. COMP    (no tape data compaction -- IDRC=NOCOMP        |
                          |                            tape data compaction -- IDRC=COMP)            |
                          |       NOSTACK vs. STACK   (non-stacked tapes = 1 file per tape volume;   |
                          |                            default STACK = multiple files per tape vol. )|
                          |         NOAFF vs. AFF     (no UNIT=AFF on DD stmts. = 1 unit per file    |
                          |                            default AFF = specify UNIT=AFF on DD stmts.)  |
                          |      VOLCOUNT=___         (default 5)                                    |
                          |      NULLUNIT=________    (generic name)                                 |
                          |       NULLSER=______      (dummy volume serial number)                   |
                          |         MODEL='____________________________________________'             |
                          |  BKUPSMSPARM  ________    (SMS data set allocation parameters)           |
                          |                                                                          |
                          +--------------------------------------------------------------------------+
```

Figure 2-21. JCL Options Definitions Worksheet (Part 3 of 6)

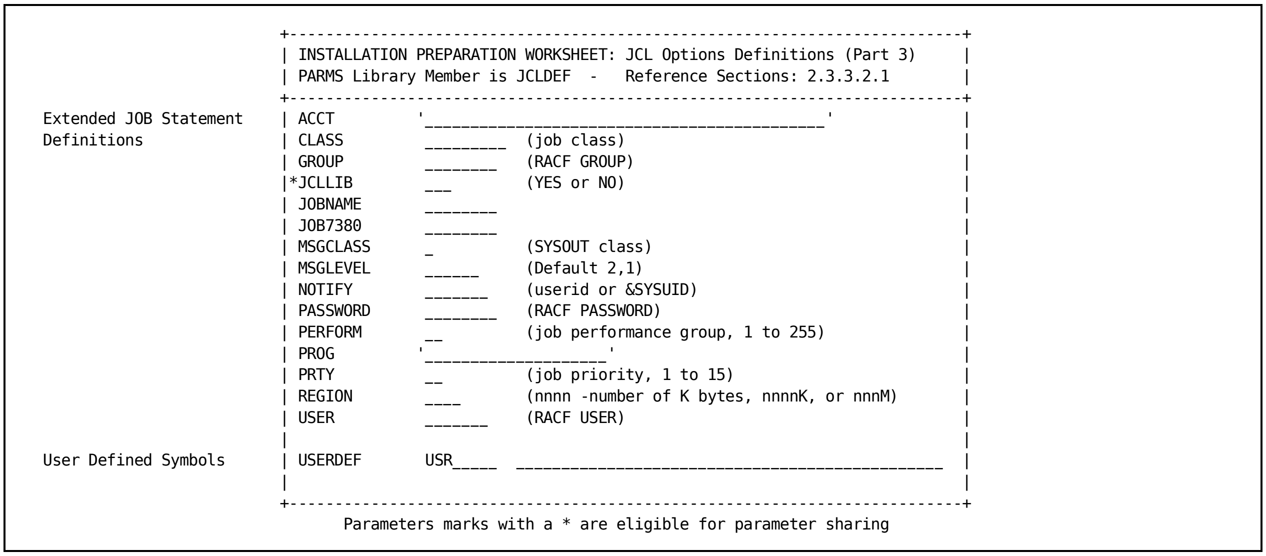
```
                          +--------------------------------------------------------------------------+
                          | INSTALLATION PREPARATION WORKSHEET: JCL Options Definitions (Part 3)     |
                          | PARMS Library Member is JCLDEF  -   Reference Sections: 2.3.3.2.1        |
                          +--------------------------------------------------------------------------+
Tape Specifications cont. | HISTMPARM                 (one or more Archive Monthly History Parameters|
                          |           GDG=__          (default 5)                                    |
                          |          UNIT=________    (generic name)                                 |
                          |         EXPDT=_______     (YYDDD or YYYY/DDD)                            |
                          |         RETPD=____        (dddd -- tape retention period)                |
                          |       NOEXPDT             (omit both EXPDT= and RETPD= from DD stmts.)   |
                          |        NOCOMP vs. COMP    (no tape data compaction -- IDRC=NOCOMP        |
                          |                            tape data compaction -- IDRC=COMP)            |
                          |       NOSTACK vs. STACK   (non-stacked tapes = 1 file per tape volume;   |
                          |                            default STACK = multiple files per tape vol. )|
                          |         NOAFF vs. AFF     (no UNIT=AFF on DD stmts. = 1 unit per file    |
                          |                            default AFF = specify UNIT=AFF on DD stmts.)  |
                          |      VOLCOUNT=___         (default 5)                                    |
                          |      NULLUNIT=________    (generic name)                                 |
                          |       NULLSER=______      (dummy volume serial number)                   |
                          |         MODEL='____________________________________________'             |
                          | HISTMSMSPARM  ________    (SMS data set allocation parameters)           |
                          |                                                                          |
                          | HISTWPARM                 (one or more Archive Weekly History Parameters |
                          |           GDG=__          (default 7)                                    |
                          |          UNIT=________    (generic name)                                 |
                          |         EXPDT=_______     (YYDDD or YYYY/DDD)                            |
                          |         RETPD=____        (dddd -- tape retention period)                |
                          |       NOEXPDT             (omit both EXPDT= and RETPD= from DD stmts.)   |
                          |        NOCOMP vs. COMP    (no tape data compaction -- IDRC=NOCOMP        |
                          |                            tape data compaction -- IDRC=COMP)            |
                          |       NOSTACK vs. STACK   (non-stacked tapes = 1 file per tape volume;   |
                          |                            default STACK = multiple files per tape vol. )|
                          |         NOAFF vs. AFF     (no UNIT=AFF on DD stmts. = 1 unit per file    |
                          |                            default AFF = specify UNIT=AFF on DD stmts.)  |
                          |      VOLCOUNT=___         (default 5)                                    |
                          |      NULLUNIT=________    (generic name)                                 |
                          |       NULLSER=______      (dummy volume serial number)                   |
                          |         MODEL='____________________________________________'             |
                          | HISTWSMSPARM  ________    (SMS data set allocation parameters)           |
                          |                                                                          |
                          +--------------------------------------------------------------------------+
```

Figure 2-21. JCL Options Definitions Worksheet (Part 4 of 6)



```
                          +---------------------------------------------------------------------------+
                          | INSTALLATION PREPARATION WORKSHEET: JCL Options Definitions (Part 3)      |
                          | PARMS Library Member is JCLDEF  -   Reference Sections: 2.3.3.2.1         |
                          +---------------------------------------------------------------------------+
Tape Specifications cont. | MBKUPPARM                 (one or more Monthly Backup Tape Parameters     |
                          |           GDG=__          (default 6)                                     |
                          |          UNIT=________    (generic name)                                  |
                          |         EXPDT=_______     (YYDDD or YYYY/DDD)                             |
                          |         RETPD=____        (dddd -- tape retention period)                 |
                          |       NOEXPDT             (omit both EXPDT= and RETPD= from DD stmts.)    |
                          |        NOCOMP vs. COMP    (no tape data compaction -- IDRC=NOCOMP         |
                          |                            tape data compaction -- IDRC=COMP)             |
                          |       NOSTACK vs. STACK   (non-stacked tapes = 1 file per tape volume;    |
                          |                            default STACK = multiple files per tape vol. )|
                          |         NOAFF vs. AFF     (no UNIT=AFF on DD stmts. = 1 unit per file     |
                          |                            default AFF = specify UNIT=AFF on DD stmts.)   |
                          |      VOLCOUNT=___         (default 5)                                     |
                          |      NULLUNIT=________    (generic name)                                  |
                          |       NULLSER=______      (dummy volume serial number)                    |
                          |         MODEL='____________________________________________'              |
                          | MBKUPSMSPARM  ________    (SMS data set allocation parameters)            |
                          |                                                                           |
                          | ACCTDGDG      __          (default 12)                                    |
                          | AUDITGDG      __          (default 53)                                    |
                          | BCKUPGDG      __          (default 10)                                    |
                          | DENSITY       _           (3 for 1600 bpi 4 for 6250 bpi or 3480)         |
                          | VCATNAME      ______      (VSAM catalog name or SYSTEM)                   |
                          | VCATOWNER     ______      (catalog owner)                                 |
                          | VCATPASSWORD  ________    (update level or higher)                        |
                          | VCATSER       ______      (volume serial)                                 |
                          | VCATUNIT      ________    (generic name)                                  |
                          |*SYSTEMMODEL  '____________________________________________'               |
                          | TMS           ______      (tape management system, YES/name/NONE/LOCAL)   |
                          |                                                                           |
                          +---------------------------------------------------------------------------+
```

Figure 2-21. JCL Options Definitions Worksheet (Part 5 of 6)



```
                          +---------------------------------------------------------------------------+
                          | INSTALLATION PREPARATION WORKSHEET: JCL Options Definitions (Part 3)      |
                          | PARMS Library Member is JCLDEF  -   Reference Sections: 2.3.3.2.1         |
                          +---------------------------------------------------------------------------+
Extended JOB Statement    | ACCT         '____________________________________________'               |
Definitions               | CLASS         _________  (job class)                                      |
                          | GROUP         ________   (RACF GROUP)                                     |
                          |*JCLLIB        ___        (YES or NO)                                      |
                          | JOBNAME       ________                                                    |
                          | JOB7380       ________                                                    |
                          | MSGCLASS      _          (SYSOUT class)                                   |
                          | MSGLEVEL      ______     (Default 2,1)                                    |
                          | NOTIFY        _______    (userid or &SYSUID)                              |
                          | PASSWORD      ________   (RACF PASSWORD)                                  |
                          | PERFORM       __         (job performance group, 1 to 255)                |
                          | PROG         '____________________'                                       |
                          | PRTY          __         (job priority, 1 to 15)                          |
                          | REGION        ____       (nnnn -number of K bytes, nnnnK, or nnnM)        |
                          | USER          _______    (RACF USER)                                      |
                          |                                                                           |
User Defined Symbols      | USERDEF       USR_____  _______________________________________________   |
                          |                                                                           |
                          +---------------------------------------------------------------------------+
                                 Parameters marks with a * are eligible for parameter sharing
```

Figure 2-21. JCL Options Definitions Worksheet (Part 6 of 6)

2.3.3.2.1.1 Database Unit Control Definitions

The control definitions define which CA MICS products are to be used, the name of the database, which products (for which SMF is an optional data source) will take their input from SMF, and the data set names' qualifiers or prefixes to be used for data set naming and cataloging.

COMPLEXPARMS:

This optional statement controls the parameter sharing status. 'YES' indicates that parameter sharing will be active; 'NO' deactivates parameter sharing. If used, this must be the first statement in JCLDEFC.

For more information about JCLGEN Parameter Sharing, see section 2.3.3.2.1.8.

DATABASE:

A DATABASE statement must be provided. The first parameter is the database name, a 1- to 8-character name used to identify the database during generation. The name should begin with an alphabetic character and cannot contain any special values or imbedded blanks.

The second parameter, following the database name, is a 1-character JCL procedure identifier called the Database Identifier. Valid values are A-Z and 0-9. This identifier must be specified; a blank is NOT a valid identifier.

Note: You should not use numeric database identifiers (0-9) until you have exhausted all possible alphabetic identifiers (A-Z). Due to syntax restrictions on ddnames and SAS names, special constructs are required for numeric units. For example, MICF allocates the A unit DAYS timespan with the ddname ADAYS and references this ddname by the &AHARD macro variable (for the Hardware Information Area). However, for the 5 unit, the ddname and macro variable are N5DAYS and &_5HARD.

The identifier is appended to the names of the following JCL procedures created by JCLGEN:

```
MICSDB   - Access database DISP=OLD
MICSDU   - Batch TSO Execution
```

```
     MICSNDB  - Invoke SAS without database allocation
     MICSSHR  - Access database DISP=SHR
```

It is necessary to have different JCL procedures for each CA MICS database unit at an installation. To prevent two databases from using the same procedure identifier, a table of previously used identifiers is maintained in sharedprefix.MICS.GENLIB(DBTABLE). This table is checked to see whether the identifier specified has been used with a different PREFIX statement. If the identifier has been previously used with a different PREFIX statement, JCLGEN processing is immediately terminated.

Since the DBTABLE is in sharedprefix.MICS.GENLIB, the duplicate procedure protection extends only to CA MICS unit databases in a complex. It is the responsibility of the CA MICS System Administrator to ensure the uniqueness of the procedure identifier, when multiple databases do not share the same shareprefix.MICS.GENLIB library. For techniques on using alternate procedure names for multiple complexes sharing a single PROCLIB, see section 2.3.3.3.2.3.

The third keyword parameter on the DATABASE statement is the database type. The possible values are PRIMARY, UNIT, SPECIAL, or TEST. For preliminary testing purposes, a TEST database unit may be installed without defining a PRIMARY database unit. However, the first production database unit installed in a database complex should have PRIMARY specified. Only the PRIMARY database backs up the complex level data libraries. Other database units may then be designated as UNIT, SPECIAL, or TEST.

Information from TEST database units is not included in CA MICS Accounting and Chargeback processing if at least one UNIT or PRIMARY unit in the complex contains CA MICS Accounting and Chargeback. TEST units are also treated differently by JCLGEN. SPECIAL databases are temporary databases for special studies or other uses.

COMPONENTS:

One or more COMPONENTS statements must be provided to specify which CA MICS products should be included in the CA MICS unit. The parameters for component identification are the standard 3-character product identifiers.

Products that do not have processing activity at the database unit level (e.g., no step in the DAILY job), can be specified but they are ignored in JCLGEN processing. PER and STG are examples of such products.

All components are optional, but at least one must be selected. Note the following special considerations:

- o BAS (CA MICS Platform) is not specified on the COMPONENTS statement, but is automatically included in every database unit.

- o The ACT identifier for CA MICS Accounting and Chargeback may not be specified in a SPECIAL database unit.

- o If you want to include sharedprefix.MICS.CAPACITY in the PRIMARY unit database BACKUP process, you can specify the CAP component on the COMPONENTS statement in the PRIMARY database unit. Otherwise, there is NO need to specify the CAP component on any COMPONENTS statement. Note, CAP is only valid when the third keyword of the DATABASE statement is PRIMARY (indicating the PRIMARY unit).

No processing support is included in the generated system for products that are not specified in the COMPONENTS statement.

SMFRECORDING:

Many CA MICS products that take input from external non-SMF log files also support the standard SMF data set as an optional data source. If your site uses SMF data rather than or in addition to external non-SMF log data, you must code one or more SMFRECORDING statements in the JCLDEF member of the prefix.MICS.PARMS data set. Valid values include AST, CICS, HSM, IDM, SNT, and VCA. For more information on products that can use SMFRECORDING, see the product guides.

If more than one CA MICS product that takes input from SMF is specified on the COMPONENTS statement described earlier in this section, a job step named DAYSMF is generated as the second step of the DAILY job. DAYSMF splits the SMF input file (defined by prefix.MICS.PARMS(INPUTRDR)) into multiple work files, one for each CA MICS product which takes its input from

SMF. The ddname of INPUTSMF is used to address the work file in DAILY job steps for products that input such work files.

If only one CA MICS product uses SMF input, JCLGEN does not create a DAYSMF step to split the SMF input file. Rather, the SMF file that is defined in prefix.MICS.PARMS(INPUTRDR) goes directly into the DAILY job step for that one product, improving CA MICS performance in this special case.

If you are using the optional incremental update facility, you may choose to execute a stand-alone version of the DAYSMF step (the SPLITSMF job) to pre-select input data for your INCRccc jobs. The SPLITSMF job dynamically allocates and populates a data set for each CA MICS product which takes its input from SMF AND which has been marked with the INCRSPLIT USE option in prefix.MICS.PARMS(cccOPS). The ddname of INPUTSMF is used to address the input file in the INCRccc jobs.

If CICS is specified on the SMFRECORDING statement, a DD statement for ddname INPUTSMF is generated by JCLGEN according to the above rules. The DAY040 step may also take input from non-SMF files defined in prefix.MICS.PARMS(INPUTCIC) even when SMF input is used (depending on the parameters that are specified on the CICOPTS statements in prefix.MICS.PARMS(CICOPS)). If CICS is not specified on the SMFRECORDING statement, then the only input available to the CICS (DAY040) step in the CA MICS DAILY update job will be that supplied by DD statements in prefix.MICS.PARMS(INPUTCIC). (The ddnames must also be identified in prefix.MICS.PARMS(CICOPS)).

If VCA is specified on the SMFRECORDING statement, a DD statement for ddname INPUTSMF is generated by JCLGEN according to the above rules. The DAY090 step may also take input from non-SMF files defined in prefix.MICS.PARMS(INPUTVCA) even when SMF input is also used. All input for the VCA Analyzer is taken from the INPUTSMF DD statement. If SMF and non-SMF data are mixed, the INPUTVCA member must contain only the DD statements to be concatenated to the INPUTSMF DD statement made by JCLGEN. If VCA is not specified on the SMFRECORDING statement, the only input available to the DAY090 step in the CA MICS DAILY update job will be that supplied by DD statements in prefix.MICS.PARMS(INPUTVCA). This DD statement must define the ddname INPUTSMF.

In the appropriate product guide, see the chapter on planning, installation definitions, or parameters for more information on the options for the product.

PREFIX:

The PREFIX statement is used to identify the data sets associated with a CA MICS database unit; therefore, this statement is required. Since there may be more than one CA MICS database at your site, the prefix must be unique and cannot have the same value as the SHAREDPREFIX. A table of prefixes and the value of sharedprefix is maintained in sharedprefix.MICS.GENLIB(DBTABLE). A prefix can consist of more than one node (qualifier), but cannot exceed 14 characters in length. The prefix is used to complete the data set names for the following data sets:

```
prefix.MICS.CHECKPT.DATA
prefix.MICS.CNTL
prefix.MICS.IMSSUS1
prefix.MICS.IMSSUS2
prefix.MICS.MODEL
prefix.MICS.MUOLIB
prefix.MICS.PARMS
prefix.MICS.RESTART.CNTL
prefix.MICS.USER.SOURCE
prefix.MICS.SPECIAL.SOURCE

prefix.MICS.DETAIL
prefix.MICS.DAYS
prefix.MICS.WEEKS
prefix.MICS.MONTHS
prefix.MICS.YEARS
```

An additional qualifier (MICS.) will be added behind your prefix value unless you specify the optional NOMICSLEVEL parameter following your prefix value. See the "Generation of CA MICS Data Set Names" discussion below for further details.

TAPEPREFIX:

The TAPEPREFIX statement is optional. It is used to uniquely identify tape data sets created by CA MICS. The prefix can consist of more than one node (qualifier), but cannot exceed 14 characters in length. If this statement is not specified or left blank, the tape data sets will have the same prefix defined on the PREFIX statement.

Note: Many (GDG) data sets will be cataloged under this index. If the prefix is a TSO user ID, it may impair that user's use of the LISTCAT command due to the voluminous output.

The TAPEPREFIX statement is used to form the name of the following CA MICS data sets:

```
tapeprefix.MICS.AUDIT.iiifff.GggggV00
tapeprefix.MICS.BACKUP.tttttt.GggggV00
tapeprefix.MICS.BACKUP.CHECKPT.GggggV00
tapeprefix.MICS.HISTW.iiifff.GggggV00
tapeprefix.MICS.HISTM.iiifff.GggggV00
tapeprefix.MICS.MBACKUP.tttttt.GggggV00
tapeprefix.MICS.MBACKUP.CHECKPT.GggggV00
```

Note: In the list above, iiifff represents a CA MICS file name and tttttt is a time-span or can be SCREENS, TABLES, CAPACITY, CIMANAGE, or ISPTLIB.

Generation of CA MICS Data Set Names

You can specify whether or not ".MICS." follows the sharedprefix, prefix, and tapeprefix levels when data set names are generated. If you want ".MICS." appended to any of these prefixes, specify the keyword "MICSLEVEL." If you do NOT want ".MICS." appended to any of these prefixes, specify the keyword "NOMICSLEVEL." Place the keyword "MICSLEVEL" or "NOMICSLEVEL" after, but on the same line as, the definition of SHAREDPREFIX, PREFIX, or TAPEPREFIX. When "NOMICSLEVEL" is specified, the prefix length limit increases from 14 to 19. If no keyword is specified, the default MICSLEVEL is used so that ".MICS." follows the prefix. Figure 2-22 shows the resultant data set names that are formed when MICSLEVEL and NOMICSLEVEL are specified in conjunction with the three prefixes.

JCLDEF Keyword Value	Data Set Names Formed
SHAREDPREFIX EYCS MICSLEVEL	EYCS.MICS.SOURCE
PREFIX EYP MICSLEVEL	EYP.MICS.CNTL
TAPEPREFIX EYCT MICSLEVEL	EYCT.MICS.BACKUP.DAYS

```
+------------- ---------------------+-----------------------+
|SHAREDPREFIX   EYCS NOMICSLEVEL    | EYCS.SOURCE           |
|PREFIX         EYP  NOMICSLEVEL    | EYP.CNTL              |
|TAPEPREFIX     EYCT NOMICSLEVEL    | EYCT.BACKUP.DAYS      |
+------------- ---------------------+-----------------------+
|SHAREDPREFIX   EYCS                | EYCS.MICS.SOURCE      |
|PREFIX EYP.TEST.MICS4 NOMICSLEVEL  | EYP.TEST.MICS4.CNTL   |
|TAPEPREFIX EYCT.DAILY NOMICSLEVEL  | EYCT.DAILY.BACKUP.DAYS|
+-----------------------------------+-----------------------+
```

Figure 2-22. NOMICSLEVEL Data Set Name Formation

A more comprehensive control over CA MICS data set names is available. For more information about CA MICS data set names, see section 2.3.3.3.2.3, CA MICS User Names Table (JCLNAMES).

2.3.3.2.1.2 Database Unit Library Definitions

The following parameter statements are identical in purpose to those found in sharedprefix.MICS.PARMS(JCLDEFC), described in section 2.3.1.6.2.

```
ISPLLIB1-2    ISPMLIB1-2  ISPTLIB1-2  OBJACCESS
OBJWAIT       PROCLIB     SASAUTOS    SASCONFIG
SASCONFIG1-4  SASHELP     SASLIB      SASMSG
SASNAME       SASVER      SORTLIB     STEPLIB1-2
USEROBJLIB
```

TSMLINK:

For CA TSO/MON PM users running Release 5, the TSMLINK statement specifies a fully-qualified name of the CA TSO/MON PM load library. CA MICS loads the CA TSO/MON PM parameter parsing module from this library. If the CA MICS Analyzer for TSO is not installed in this unit or if the CA TSO/MON PM load modules are contained in the system LPA or LINKLIST libraries, comment out this statement.

For CA TSO/MON PM users running Release 6, do not code the TSMLINK statement.

TSMPARMS:

For CA TSO/MON PM users running Release 5, the TSMPARMS statement specifies a fully-qualified name of the CA TSO/MON PM parameter library containing your current commands table. CA MICS calls a CA TSO/MON PM program that reads your command table which is used to translate CA TSO/MON PM's two-character command abbreviations back into the full-length real names of the TSO commands. If the CA MICS Analyzer for TSO is not installed in this unit, comment out this statement.

For CA TSO/MON PM users running Release 6, do not code the TSMPARMS statement.

2.3.3.2.1.3 Database Unit Execution Option Definitions

The execution options parameter statements enable you to select database archive options, select the method for file storage of work files used in the daily CA MICS job, and define the frequency of database backup.

The options discussed in this section must be specified before the CA MICS system is generated. If they are changed, part of the generation process must be repeated before the change takes effect. There are additional execution options that may be changed without running any of the CA MICS generation processes. These parameters are discussed in Section 2.3.5.

The database archive options include Audit Data, Weekly History Data, and Monthly History Data. Each archive option must be specified as either YES (activate and use) or NO (do not implement). Optional parameters let you execute archive processing as part of the standard WEEKLY and/or MONTHLY jobs or in stand-alone jobs that can execute in parallel with reporting and end-user inquiry processing. For more information about audit archiving and history archiving, see sections 4.3.13.1 and 4.3.13.2 respectively.

ARCHIVE AUDIT:

Specify either YES or NO to indicate whether the Archive Audit is to be activated. If you specify YES, then further control over the amount of data available in the Archive Audit files is available through the AUDITGDG parameter described below and parameters you specify in prefix.MICS.PARMS(DBMODEL). The default is YES.

If you specify YES, you can also specify either STEP or JOB in the optional third positional parameter to indicate whether the Archive Audit processing is to be included in the WEEK300 step of the WEEKLY operational job, or if Archive Audit processing is to be offloaded to the stand-alone AUDIT operational job. STEP is the default.

With the JOB option, you can reduce WEEKLY operational job execution time and increase database availability by moving the Archive Audit processing to the stand-alone AUDIT job. This job allocates the CA MICS database files DISP=SHR and thus can execute concurrently with reporting, end-user inquiry, and other read-only processing. It must complete before the next DAILY operational job is allowed to execute. If AUDIT has not completed, then the DAILY job DAYALL step will abend with messages reporting that AUDIT must complete before executing DAILY.

Note: The STEP/JOB parameter is ignored when you specify NO (no Archive Audit processing).

Specify either AUTOSUBMIT or NOAUTOSUBMIT in the optional fourth positional parameter to indicate whether the WEEK300 step of the WEEKLY operational job will automatically write the AUDIT job to the internal reader (AUTOSUBMIT) or whether you will submit the AUDIT job externally, per your data center's production job scheduling facilities (NOAUTOSUBMIT). NOAUTOSUBMIT is the default.

- o If your data center standards allow job submission through the internal reader, specifying AUTOSUBMIT guarantees that the AUDIT job is submitted for execution at the proper time. Remember that the AUDIT job must complete before the next DAILY operational job can execute.

- o If you specify NOAUTOSUBMIT (or take the default) you must ensure that the AUDIT job is submitted for execution after the WEEKLY job completes.

- o The AUTOSUBMIT/NOAUTOSUBMIT parameter is ignored when you specify NO (no Archive Audit processing) in the second positional parameter, or if you specify STEP in the third positional parameter (perform Archive Audit processing in the WEEK300 step). You must specify the third positional parameter in order to specify a value for the AUTOSUBMIT/NOAUTOSUBMIT parameter.

The following example ARCHIVE specification activates Archive Audit processing and specifies that the WEEKLY job's WEEK300 step automatically submit the stand-alone AUDIT operational job to perform Archive Audit processing.

```
ARCHIVE AUDIT YES JOB AUTOSUBMIT
```

ARCHIVE HISTW:

Specify either YES or NO to indicate whether the Archive Weekly History is to be activated or not. Note that if

YES is specified, further control over the contents of this file is available through parameters you specify in prefix.MICS.PARMS(DBMODEL). The default is NO.

If YES is specified, you can specify either STEP or JOB in the optional third positional parameter to indicate whether the Archive Weekly History processing is to be included in the WEEK300 step of the WEEKLY operational job, or if Archive Weekly History processing is to be offloaded to the stand-alone HISTW operational job. STEP is the default.

With the JOB option, you can reduce WEEKLY operational job execution time and increase database availability by moving the Archive Weekly History processing to the stand-alone HISTW job. This job allocates the CA MICS database files DISP=SHR and thus can execute concurrently with reporting, end-user inquiry, and other read-only processing. It must complete before the next WEEKLY operational job is allowed to execute. If HISTW has not completed, then the WEEKLY job will abend with messages reporting that HISTW must complete before executing WEEKLY.

Note: The STEP/JOB parameter is ignored when you specify NO (no Archive Weekly History processing).

Specify either AUTOSUBMIT or NOAUTOSUBMIT in the optional fourth positional parameter to indicate whether the WEEK300 step of the WEEKLY operational job will automatically write the HISTW job to the internal reader (AUTOSUBMIT) or whether you will submit the HISTW job externally, per your data center's production job scheduling facilities (NOAUTOSUBMIT). NOAUTOSUBMIT is the default.

o If your data center standards allow job submission through the internal reader, specifying AUTOSUBMIT guarantees that the HISTW job is submitted for execution at the proper time. Remember that the HISTW job must complete before the next WEEKLY operational job can execute.

o If you specify NOAUTOSUBMIT (or take the default) you must ensure that the HISTW job is submitted for execution after the WEEKLY job completes.

o The AUTOSUBMIT/NOAUTOSUBMIT parameter is ignored when you specify NO (no Archive Weekly History processing) in the second positional parameter, or if you specify STEP in the third positional parameter (perform Archive Weekly History processing in the WEEK300 step). You must specify the third positional parameter in order to specify a value for the AUTOSUBMIT/NOAUTOSUBMIT parameter.

The following example ARCHIVE specification activates Archive Weekly History processing and specifies that the WEEKLY job's WEEK300 step automatically submit the stand-alone HISTW operational job to perform Archive Weekly History processing.

```
ARCHIVE HISTW YES JOB AUTOSUBMIT
```

ARCHIVE HISTM:

Specify either YES or NO to indicate whether the Archive Monthly History is to be activated. Note that if YES is specified, further control over the contents of this file is available through parameters you specify in prefix.MICS.PARMS(DBMODEL). The default is YES.

If YES is specified, you can specify either STEP or JOB in the optional third positional parameter to indicate whether the Archive Monthly History processing is to be included in the MONTH300 step of the MONTHLY operational job, or if Archive Monthly History processing is to be offloaded to the stand-alone HISTM operational job. STEP is the default.

With the JOB option, you can reduce MONTHLY operational job execution time and increase database availability by moving the Archive Monthly History processing to the stand-alone HISTM job. This job allocates the CA MICS database files DISP=SHR and thus can execute concurrently with reporting, end-user inquiry, and other read-only processing. It must complete before the next MONTHLY operational job is allowed to execute. If HISTM has not completed, then the MONTHLY job will abend with messages reporting that HISTM must complete before executing MONTHLY.

Note: The STEP/JOB parameter is ignored when you specify NO (no Archive Monthly History processing).

Specify either AUTOSUBMIT or NOAUTOSUBMIT in the optional fourth positional parameter to indicate whether the MONTH300 step of the MONTHLY operational job will automatically write the HISTM job to the internal reader (AUTOSUBMIT) or whether you will submit the HISTM job externally, per your data center's production job scheduling facilities (NOAUTOSUBMIT). NOAUTOSUBMIT is the default.

- If your data center standards allow job submission through the internal reader, specifying AUTOSUBMIT guarantees that the HISTM job is submitted for execution at the proper time. Remember that the HISTM job must complete before the next MONTHLY operational job can execute.

- If you specify NOAUTOSUBMIT (or take the default), you must ensure that the HISTM job is submitted for execution after the MONTHLY job completes.

- The AUTOSUBMIT/NOAUTOSUBMIT parameter is ignored when you specify NO (no Archive Monthly History processing) in the second positional parameter, or if you specify STEP in the third positional parameter (perform Archive Monthly History processing in the MONTH300 step). You must specify the third positional parameter in order to specify a value for the AUTOSUBMIT/NOAUTOSUBMIT parameter.

The following example ARCHIVE specification activates Archive Monthly History processing and specifies that the MONTHLY job's MONTH300 step automatically submit the stand-alone HISTM operational job to perform Archive Monthly History processing.

```
ARCHIVE HISTM YES JOB AUTOSUBMIT
```

BACKUP FREQ:

This parameter tells CA MICS how often to back up the entire online database. Your options are DAILY, BI-DAILY (every other day, which is really every other CA MICS update run), and WEEKLY. The default is DAILY.

Note: We recommend BI-DAILY. The trade-off you must make is between processing time to do the backups and processing time to do recovery if the database is destroyed (the more recent the backup, the less raw measurement data you will have to reprocess to get back up-to-date).

DAYSMF FILES:

It is possible for the CA MICS DAILY update job to abend for reasons other than program errors (for example, a tape I/O error). CA MICS supports a comprehensive restart facility should this occur. The intermediate files written by the DAYSMF step must not be scratched in order to restart the DAILY job. If you specify PERMANENT on this statement, these files are always available and there is no danger of your local disk space manager scratching them before you can restart the job. If you specify TEMPORARY, these files are allocated at the beginning of the CA MICS DAILY job and scratched at its end. They are allocated and cataloged with "permanent" OS data set names even if you specify TEMPORARY. However, they might be on disk volumes that get "cleaned" before the job can be restarted.

This parameter only has meaning if the DAILY job uses intermediate files for SMF data. For more information, see the SMFRECORDING option in section 2.3.3.2.1.1, "Database Unit Control Definitions."

Also, this parameter has no effect on processing performed or data sets allocated by the optional incremental update facility SPLITSMF job (the stand-alone DAYSMF step) that splits data for INCRccc processing. SPLITSMF processing is controlled by the SMFRECORDING option and by the individual product INCRSPLIT USE parameter in prefix.MICS.PARMS(cccOPS).

DAYSMF OFF:

To avoid reading the same SMF data records into multiple DAILY job steps, you may create a job that separates the records into permanent OS data sets. This is the same process performed by the DAYSMF step except that you must provide for the needs of all the components in the database unit. The data sets thus created have to be defined to CA MICS via the INPUTccc members in prefix.MICS.PARMS. Since you are preprocessing the SMF data, the CA MICS DAYSMF step must not be generated in the DAILY jobs. To turn off the DAYSMF step, code DAYSMF OFF in your JCLDEF member and regenerate each DAILY job stream.

This parameter applies only to the DAILY job, has no effect on the optional incremental update facility SPLITSMF job, and is unrelated to the INCRSPLIT USE parameter in prefix.MICS.PARMS(cccOPS).

```
DAYSMF EXCLUDE ccc ccc (component list)

     In some cases, it may be desirable to exclude a
     component from DAYSMF processing. For example, the DB2
     data may be so voluminous that it has already been
     filtered by an external process and therefore does not
     need to pass through the DAYSMF step.  Or, you may be
     using the CA SMF Director split facility which has
     already separated the input file.  Use this form of the
     DAYSMF parameter to exclude one or more components (not
     applicable to Field Developed Applications (FDAs)).

SMFDIRECTOR ccc ccc (component list)

     This optional parameter allows you to specify which
     components will use a CA SMF Director duplicate split
     index as input in their respective DAILY update step.
     For each component specified on this statement, the
     prefix.MICS.PARMS(INPUTccc) must be updated to add the
     SMFDRCTR DD statement.  See section 4.10 of PIOM guide
     for complete information on using the CA SMF Director
     duplicate split indices as input.

EXCLUDESTEP timespan stepnumber:

     This optional parameter lets you exclude certain steps
     from the operational jobs.  The timespan and stepnumber
     that can be specified are as follows:

        timespan:   DAILY WEEKLY MONTHLY YEARLY
        stepnumber: 200 400 500

     The 200 steps are exception analyzer database updates,
     the 400 steps are for reporting, and the 500 steps are
     for user processing.  If any of these steps are not
     needed, excluding them from the operational jobs results
     in less resource consumption.

     More than one stepnumber can be specified on one line,
     and more than one EXCLUDESTEP line is allowed. Here is
     an example:

        EXCLUDESTEP DAILY 500
        EXCLUDESTEP WEEKLY 400 500

     In this example the DAY500 step is excluded from the
     DAILY job, and the WEEK400 and WEEK500 steps are
     excluded from the WEEKLY job.
```

The default is to include all steps.

MONTHLY BACKUP:

This parameter determines whether the MONTHLY job will perform a database backup. If NO is specified, the MONTHLY job does not submit the backup job BACKUPM. In this case, consider submitting the BACKUPM job independently after the MONTHLY job to keep a monthly backup of your database. The default is MONTHLY BACKUP YES.

RESTORE BACKUP:

This parameter determines whether the RESTORE job will attempt to back up the database before performing the restore operation. If YES is specified and the RESTORE job cannot perform the backup because of damage to the database, the RESTORE job can be restarted at the restore step bypassing the failing backup step. The default is RESTORE BACKUP NO.

2.3.3.2.1.4 Database Unit JCL Definitions

```
The following parameter statements are identical in purpose
to those found in sharedprefix.MICS.PARMS(JCLDEFC), described
in section 2.3.1.6.3.

     AUTOSUBMIT
     COLIBSPACE
     DASDUNIT
     DASDSMSPARMS
     DIRSTATS
     DYNAMWAIT
     INTRDRCLASS
     JES
     LOGS
     MICSDMDELIM
     MICSLOG
     RENUMBER
     REPORTS
     SORTCYL
     SORTUNIT
     SORTSMSPARMS
     SORTUTILITY
     SORTWKNO
     STEPCPULIMIT
     UOLIBSPACE
     WORKSPACE
     WORKUNIT
     WORKVOLSER
     WORKSMSPARMS
```

2.3.3.2.1.5 Database Unit Tape Specifications

Tape specifications let you control the number of generation data group entries maintained for backup and archive tape files, unit names, volume count, tape volume density, retention and expiration dates, as well as the VSAM or ICF catalog used to maintain the tape catalog information.

The accounting daily, accounting monthly, backup, monthly backup, archive audit, archive history, and DETAIL tape processes all use generation data groups.

- o The accounting processes are described in the Accounting and Chargeback Option User Guide.
- o The backup process is described in section 4.3.3.2.5, BACKUP Operational Job, of this guide.
- o The monthly backup process is described in section 4.3.3.2.3, MONTHLY Operational Job, of this guide.
- o The two database archive processes, audit and history, are described in sections 4.3.13.1 and 4.3.13.2 of this guide and section 1.2.2, Archive Data Structure, in the Database Structure and Content Guide.
- o The DETAIL tape process is described in the relevant component guides.
- o The data set names used by these processes are itemized in section 4.3.10, Data Set Names, of this guide.

The numbers specified for the GDG=nnn parameters on the AUDITPARM, BKUPPARM, HISTMPARM, HISTWPARM, and MBKUPPARM statements (or on the older ACCTDGDG, AUDITGDG, BCKUPGDG, HISTWGDG, and HISTMGDG statements) serve two purposes.

- o The numbers define the number of entries in the respective GDG (BLDG ENTRIES=n).
- o The numbers, when multiplied by the specified retention factor, provide the default JCL LABEL retention period (RETPD=days) that is used in creating the respective GDG files. For instance, the AUDITGDG files are created weekly. Therefore, the retention period specified by CA MICS is the number of GDG entries you specify multiplied by 7 (days). However, any value specified in the EXPDT=expdt parameter overrides the calculated RETPD.

ACCTDGDG:

This is a required statement if you are installing the Accounting and Chargeback Option; otherwise, it is optional and can be ignored if specified. Specify the number of generations of daily audit files that are to be maintained in the daily generation data group. The default value is 12, and the allowable range is 1 to 999.

TAPEPARM

Use the TAPEPARM statement to specify parameters that define default processing options for CA MICS tape data sets.

The TAPEPARM statement overrides and replaces many of the old-style individual tape data set statements in order to consolidate tape data set definitions into a single, easy-to-locate statement. The individual old-style statements that are replaced are noted in the detailed descriptions of the TAPEPARM parameters that follow.

You can, optionally, override individual TAPEPARM parameter specifications using statements that apply to specific types of CA MICS tape data sets, such as the CA MICS backup, monthly backup, and archive tape files. As a result, you have the option of establishing a common set of defaults that you override, as required, for specific tape applications.

The general form of the TAPEPARM statement is the keyword TAPEPARM followed by one or more TAPEPARM parameters. TAPEPARM parameters include both keyword parameters and keyword=value specifications. You can specify the TAPEPARM statement as many times as you want in order to specify all of the tape data set definitions you require.

The following summarizes the TAPEPARM parameters:

- o Each parameter is described in detail below.

- o While the following syntax sample shows only one or two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character-maximum line length). You may also choose to specify only one parameter per line. Either approach is acceptable.

```
TAPEPARM UNIT=unitname VOLCOUNT=n
TAPEPARM EXPDT=expdt / NOEXPDT
TAPEPARM COMP/NOCOMP
TAPEPARM NULLUNIT=unitname NULLSER=nulser
TAPEPARM STACK/NOSTACK  AFF/NOAFF
TAPEPARM MODEL='modeldsn' / MODEL=NOMODEL
TAPEPARM DISK / TAPE  SPACE=spacedef
TAPEPARM VOLSER=volser
```

The following is a sample TAPEPARM specification:

```
TAPEPARM UNIT=CART VOLCOUNT=10 EXPDT=1999/000 COMP
TAPEPARM NULLUNIT=CART NULLSER=NULSER
TAPEPARM MODEL='CPM981.CMN.MICS.MODEL'
```

The following describes the TAPEPARM parameters:

- o UNIT=unitname

  Defaults to UNIT=3400-5.

  The generic unit name for tape file allocation. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

  Replaces and overrides the old-style TAPEUNIT statement.

  Note: The ASIS sub-parameter is not required when using the TAPEPARM statement UNIT parameter.

- VOLCOUNT=n

  No default. The OS/390 default volume count of 5 will apply.

  The volume count JCL parameter for CA MICS tape data sets. Specify a decimal number from 1 to 255.

  Replaces and overrides the old-style VOLCOUNT statement.
- EXPDT=expdt

  No default. If no expiration date is specified, a default retention period will be derived as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

  Specify an expiration date definition which replaces the derived default retention period (RETPD) specification(s). This facility is specifically intended for use by installations with tape management systems which require specific EXPDT definitions.

  Note that for some tape management systems, certain EXPDT values will make the tape available for scratching immediately. Such values are NOT desirable for CA MICS tapes.

  The EXPDT=expdt parameter is optional, and, if omitted, CA MICS will derive a RETPD=nnn specification as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

  Note: If you specify an EXPDT=expdt, then you must also specify a value for the TMS statement (e.g., TMS NONE).

  Replaces and overrides the old-style EXPDT statement.

  Note: The EXPDT=expdt and NOEXPDT parameters are mutually exclusive.

o NOEXPDT

Specify NOEXPDT if you want CA MICS to omit both the EXPDT=date and RETPD=days JCL parameters from generated JCL statements for new tape data set creation. This option is provided for those installations where the tape management system or SMS specifications automatically control tape expiration dates and/or retention periods.

Note: If you specify the NOEXPDT parameter, you must also specify a value for the TMS statement (e.g., TMS NONE).

Replaces and overrides the old-style EXPDT statement.

Note: The NOEXPDT and EXPDT=expdt parameters are mutually exclusive.

o COMP / NOCOMP

No default. The OS/390 default for the selected tape unit type will apply.

This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) for a tape device enabled for hardware compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified, no TRTCH subparameter will be generated.

Replaces and overrides the old-style IDRC statement.

o NULLUNIT=unitname

Defaults to the tape unit value (UNIT=unitname).

The generic unit name to be used for tape file allocation for the NULLSER. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

Replaces and overrides the old-style NULLUNIT statement.

- NULLSER=nulser

  Defaults to nulser.

  The value to be used to signify a dummy tape volume to CA MICS. This value is used for pre-allocating dummy tape files when the system is first run. This technique allows CA MICS to run without error even though certain input files do not exist. A trap in the macro routine recognizes the NULLSER value and generates the appropriate SET statements. If your installation standards prevent using the default value (NULSER), then you must change this value. An example is in the case of automatic tape loaders (ATLs).

  For example, if you are adding a new file to history archive in an existing unit, and the existing history archive tapes are in an ATL, you may need to change your NULLSER value to an actual volume serial number contained in the ATL. This prevents an additional tape device (outside the ATL) from being allocated, as well as the MOUNT for the volume serial NULSER. This is imperative for JES3 customers with ATLs because the job will not continue until the mount is satisfied.

  Replaces and overrides the old-style NULLSER statement.
- STACK / NOSTACK

  Defaults to STACK.

  Specify STACK (the default) to "stack" multiple CA MICS tape files on a single tape volume. This is the normal mode of operation where a minimum number of tape volumes are used and tape mounts are minimized. CA MICS JCL statements are generated with the VOL=REF=ddname construct and unit affinity is enforced (that is, STACK also means AFF).

  Specify NOSTACK to place exactly one CA MICS tape file on a tape volume. With this option the generated CA MICS JCL statements will NOT contain the VOL=REF=ddname construct. You may find this option useful if your implementation of a virtual tape system prohibits "stacked" tape volumes, or if you are directing CA MICS tape files to a DASD pool backed with an archival facility.

- AFF / NOAFF

  Defaults to AFF.

  Specify AFF to use a single tape unit for all related tape files. This is the normal mode of operation where a minimum number of tape units are used and tape mounts are minimized. CA MICS JCL statements are generated with the UNIT=AFF=ddname construct. AFF is required for the STACK option.

  Specify NOAFF to enable allocation of different tape units for each CA MICS tape data set. With this option the generated CA MICS JCL statements will NOT contain the UNIT=AFF=ddname construct. You may find this option useful with certain virtual tape system implementations, or where you are directing CA MICS tape files to a DASD pool backed with an archival facility.

  Note, if you specify NOAFF, then NOSTACK is automatically set.
- MODEL='modeldsn' / MODEL=NOMODEL

  No default.

  The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted, a CA MICS generation job will allocate a default model GDG data set.

  If NOMODEL is specified as the model dataset name, the use of the model dataset in the DCB specification will be bypassed.
  Replaces and overrides the old-style SYSTEMMODEL statement.

  Note: The MODEL='modeldsn' and MODEL=NOMODEL parameters are mutually exclusive.

- DISK / TAPE

  Defaults to TAPE.

  Specify DISK if you want to write the dataset to a disk instead of a tape. Specify TAPE to write the dataset to a tape.

  When you specify DISK, make sure that you have both a UNIT parameter with a generic unit name of a disk device and a SPACE parameter defining disk space. You can also define a VOLSER parameter to direct the data to a specific disk.
- SPACE=spacedef

  No default.

  Defines the space allocation parameter for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.
- VOLSER=volser

  No default.

  Defines the volume serial number of a disk for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

TAPESMSPARM

Optional parameter for specifying System Managed Storage (SMS) keywords (for example, STORCLAS=storclas) to be used as the default for CA MICS tape data set allocations. Specify one or more SMS parameters according to JCL language syntax. There is no default, and continuation is not allowed.

ACCTDTAPEPARM

Use the optional ACCTDTAPEPARM statement to specify parameters defining processing options for Accounting and Chargeback Daily Audit tape data sets. When using the Daily Audit tapes, ACCTDTAPEPARM parameter specifications override the tape data set defaults established by the TAPEPARM statement.

The primary reason to specify the ACCTDTAPEPARM statement is to override the default number of daily Audit data group entries to be defined (the GDG= parameter). However, your site may want to direct the Accounting and Chargeback Daily Audit to a unique tape volume pool, or to a tape device type such as a virtual tape unit.

Note: Directing the Daily Audit data set to disk with the TAPEPARM DISK parameter is currently not supported.

The general form of the ACCTDTAPEPARM statement is the keyword ACCTDTAPEPARM followed by one or more ACCTDTAPEPARM parameters. ACCTDTAPEPARM parameters include both keyword parameters and keyword=value specifications. You can specify the ACCTDTAPEPARM statement as many times as you want in order to specify all of the tape data set definitions you require.
If you specify a parameter multiple times, the last instance defines the value in effect.

The following summarizes the ACCTDTAPEPARM parameters:

- o Each parameter is described in detail below.

- o While the following syntax sample shows only one or two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character-maximum line length). You may also choose to specify only one parameter per line. Either approach is acceptable.

```
    ACCTDTAPEPARM GDG=nnn UNIT=unitname VOLCOUNT=n
    ACCTDTAPEPARM EXPDT=expdt / RETPD=nnn / NOEXPDT
    ACCTDTAPEPARM COMP/NOCOMP
    ACCTDTAPEPARM NULLUNIT=unitname NULLSER=nulser
    ACCTDTAPEPARM STACK/NOSTACK AFF/NOAFF
    ACCTDTAPEPARM MODEL='modeldsn' / MODEL=NOMODEL
```

The following is a sample ACCTDTAPEPARM statement:

```
ACCTDTAPEPARM GDG=3 VOLCOUNT=7
```

In this example, the TAPEPARM parameter values are used for most of the ACCTDTAPEPARM parameters.

The following describes each of the ACCTDTAPEPARM parameters.

o GDG=nnn

Defaults to 12.

The number of entries to be created in the GDG for each Accounting and Chargeback Daily Audit data set.

Overrides and replaces the ACCTDGDG statement.

o UNIT=unitname

Defaults to the value specified on the TAPEPARM statement or to UNIT=3400-5 if UNIT=unitname is not specified on the TAPEPARM statement.

The generic unit name for tape file allocation. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

Overrides the TAPEPARM statement UNIT=unitname value and/or the old-style TAPEUNIT statement.

o VOLCOUNT=n

Defaults to the value specified on the TAPEPARM statement. If no VOLCOUNT is specified, then the OS/390 default volume count of 5 will apply.

The volume count JCL parameter for CA MICS tape data sets. Specify a decimal number from 1 to 255.

Overrides the TAPEPARM statement VOLCOUNT=nnn parameter or the old-style VOLCOUNT statement.

- EXPDT=expdt

  Defaults to the value specified on the TAPEPARM statement. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

  Specify an expiration date definition which replaces the derived default retention period (RETPD) specification(s). This facility is specifically intended for use by installations with tape management systems which require specific EXPDT definitions.

  Note that for some tape management systems, certain EXPDT values will make the tape available for scratching immediately. Such values are NOT desirable for CA MICS tapes.

  The EXPDT=expdt parameter is optional, and, if omitted, CA MICS will derive a RETPD=nnn specification as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

  Note: If you specify an EXPDT=expdt, then you must also specify a value for the TMS statement (e.g., TMS NONE).

  Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameter, or the old-style EXPDT statement.

  Note: The EXPDT=expdt, RETPD=nnn, and NOEXPDT parameters are mutually exclusive.

- NOEXPDT

  Defaults to the value specified on the TAPEPARM

statement. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

Specify NOEXPDT if you want CA MICS to omit both the EXPDT=date and RETPD=days JCL parameters from generated JCL statements for new tape data set creation. This option is provided for those installations where the tape management system or SMS specifications automatically control tape expiration dates and/or retention periods.

Note: If you specify the NOEXPDT parameter, you must also specify a value for the TMS statement (e.g., TMS NONE).

Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameters, or the old-style EXPDT statement.

Note: The NOEXPDT, EXPDT=expdt, and RETPD=nnn parameters are mutually exclusive.

o COMP / NOCOMP

Defaults to the value specified on the TAPEPARM statement. If neither COMP nor NOCOMP is specified, then the OS/390 default for the selected tape unit type will apply.

This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) for a tape device enabled for hardware compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified, no TRTCH subparameter will be generated.

Overrides the TAPEPARM statement COMP/NOCOMP parameter or the old-style IDRC statement.

o NULLUNIT=unitname

Defaults to the value specified on the TAPEPARM statement or to the tape unit value.

The generic unit name to be used for tape file allocation for the NULLSER. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

Overrides the TAPEPARM statement NULLUNIT=unitname parameter or the old-style NULLUNIT statement.

o NULLSER=nulser

Defaults to the value specified on the TAPEPARM statement or to NULLSER.

The value to be used to signify a dummy tape volume to CA MICS. This value is used for pre-allocating dummy tape files when the system is first run. This technique allows CA MICS to run without error even though certain input files do not exist. A trap in the macro routine recognizes the NULLSER value and generates the appropriate SET statements. If your installation standards prevent using the default value (NULSER), then you must change this value. An example is in the case of automatic tape loaders (ATLs).

For example, if you are adding a new file to history archive in an existing unit, and the existing history archive tapes are in an ATL, you may need to change your NULLSER value to an actual volume serial number contained in the ATL. This prevents an additional tape device (outside the ATL) from being allocated, as well as the MOUNT for the volume serial NULSER. This is imperative for JES3 customers with ATLs because the job will not continue until the mount is satisfied.

Overrides the TAPEPARM statement NULLSER=nulser parameter or the old-style NULLSER statement.

o STACK / NOSTACK

Defaults to the value specified on the TAPEPARM statement or to STACK.

Specify STACK (the default) to "stack" multiple CA MICS tape files on a single tape volume. This is the normal mode of operation where a minimum number of tape volumes are used and tape mounts are minimized. CA MICS JCL statements are generated with the VOL=REF=ddname construct and unit affinity is enforced (that is, STACK also means AFF).

Specify NOSTACK to place exactly one CA MICS tape file on a tape volume. With this option the generated CA MICS JCL statements will NOT contain the VOL=REF=ddname construct. You may find this option useful if your implementation of a virtual tape system prohibits "stacked" tape volumes, or if you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Overrides the TAPEPARM statement STACK/NOSTACK parameter.

o AFF / NOAFF

Defaults to the value specified on the TAPEPARM statement or to AFF.

Specify AFF to use a single tape unit for all related tape files. This is the normal mode of operation where a minimum number of tape units are used and tape mounts are minimized. CA MICS JCL statements are generated with the UNIT=AFF=ddname construct. AFF is required for the STACK option.

Specify NOAFF to enable allocation of different tape units for each CA MICS tape data set. With this option the generated CA MICS JCL statements will NOT contain the UNIT=AFF=ddname construct. You may find this option useful with certain virtual tape system implementations, or where you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Note, if you specify NOAFF, then NOSTACK is automatically set.

Overrides the TAPEPARM statement AFF/NOAFF parameter.

o MODEL='modeldsn' / MODEL=NOMODEL

Defaults to the value specified on the TAPEPARM statement.

The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted, a CA MICS generation job will allocate a default model GDG data set.

If NOMODEL is specified as the model dataset name, the use of the model dataset in the DCB specification will be bypassed.

Overrides the TAPEPARM statement MODEL='modeldsn' parameter or the old-style SYSTEMMODEL statement.

ACCTDTAPESMSPARM

Defaults to the values specified on the TAPESMSPARM statement.

Optional parameter for specifying System Managed Storage (SMS) keywords (e.g., STORCLAS=storclas) for Accounting and Chargeback Daily Audit data set allocations. Specify one or more SMS parameters according to JCL language syntax, separated by commas. Continuation is not allowed.

Overrides the TAPESMSPARM statement.

ACCTDAY1PARM

Use the optional ACCTDAY1PARM statement to specify parameters defining processing options for the Accounting and Chargeback DAY1 Audit data sets. When using the Daily Audit data sets, ACCTDAY1PARM parameter specifications override the data set defaults established by the ACCTDTAPEPARM statement.

The primary reason to specify the ACCTDAY1PARM statement is to override the default writing of the Accounting and Chargeback Daily Audit data sets to tape. With the ACCTDAY1PARM statement, the Accounting and Chargeback Daily Audit data set can be directed to a disk device.

The general form of the ACCTDAY1PARM statement is the keyword ACCTDAY1PARM followed by one or more ACCTDAY1PARM parameters. ACCTDAY1PARM parameters include both keyword parameters and keyword=value specifications. You can specify the ACCTDAY1PARM statement as many times as you want in order to specify all of the tape data set definitions you require. If you specify a parameter multiple times, the last instance defines the value in effect.

The following summarizes the ACCTDAY1PARM parameters:

- Each parameter is described in detail next.

- While the following syntax sample shows only one or two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character-maximum line length). You may also choose to specify only one parameter per line. Either approach is acceptable.

```
    ACCTDAY1PARM DISK SPACE=spacedef
    ACCTDAY1PARM UNIT=unitname VOLSER=volser
```

The following is a sample ACCTDAY1PARM statement:

```
ACCTDAY1PARM DISK SPACE=(CYL,(20,10)) UNIT=SYSDA
```

The following describes each of the ACCTDAY1PARM parameters.

- DISK

  Required if you want the Accounting and Chargeback DAY1 Audit data set written to a disk device. If not specified, the data set is written to tape and other parameters on the ACCTDAY1PARM statement have no effect.

- SPACE=spacedef

  Default: No SPACE parameter specified.

  Defines the space allocation parameter for the Accounting and Chargeback DAY1 Audit data set on disk. Specify this parameter according to JCL language syntax.

- UNIT=unitname

  Default: No UNIT parameter specified.

The generic unit name to be used for the Accounting and Chargeback DAY1 Audit data set on disk. Specify this parameter according to JCL language syntax.

o VOLSER=volser

Default: No VOLSER parameter specified.

Define the VOLSER parameter if you want the Accounting and Chargeback DAY1 Audit data set written onto a specific disk. Specify this parameter according to JCL language syntax.

ACCTDAY1SMSPARM

Optional parameter for specifying System Managed Storage (SMS) keywords (e.g., STORCLAS=storclas) for Accounting and Chargeback DAY1 Audit data set allocations on disk. Specify one or more SMS parameters according to JCL language syntax, separated by commas. Continuation is not allowed.

The parameters on this statement are effective only when the DISK parameter on the ACCTDAY1PARM statement is specified.

ACCTDAY2PARM

Use the optional ACCTDAY2PARM statement to specify parameters defining processing options for the Accounting and Chargeback DAY2 Audit data sets. When using the Daily Audit data sets, ACCTDAY2PARM parameter specifications override the data set defaults established by the ACCTDTAPEPARM statement.

The primary reason to specify the ACCTDAY2PARM statement is to override the default writing of the Accounting and Chargeback Daily Audit data sets to tape. With the ACCTDAY2PARM statement, the Accounting and Chargeback Daily Audit data set can be directed to a disk device.

The general form of the ACCTDAY2PARM statement is the keyword ACCTDAY2PARM followed by one or more ACCTDAY2PARM parameters. ACCTDAY2PARM parameters include both keyword parameters and keyword=value specifications. You can specify the ACCTDAY2PARM

statement as many times as you want in order to specify all of the tape data set definitions you require.
If you specify a parameter multiple times, the last instance defines the value in effect.

The following summarizes the ACCTDAY2PARM parameters:

- Each parameter is described in detail next.

- While the following syntax sample shows only one or two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character-maximum line length). You may also choose to specify only one parameter per line. Either approach is acceptable.

```
ACCTDAY2PARM DISK SPACE=spacedef
ACCTDAY2PARM UNIT=unitname VOLSER=volser
```

The following is a sample ACCTDAY2PARM statement:

```
ACCTDAY2PARM DISK SPACE=(CYL,(20,10)) UNIT=SYSDA
```

The following describes each of the ACCTDAY2PARM parameters.

- DISK

  Required if you want the Accounting and Chargeback DAY2 Audit data set written to a disk device. If not specified, the data set is written to tape and other parameters on the ACCTDAY2PARM statement have no effect.

- SPACE=spacedef

  Default: No SPACE parameter specified.

  Defines the space allocation parameter for the Accounting and Chargeback DAY2 Audit data set on disk. Specify this parameter according to JCL language syntax.

- UNIT=unitname

  Default: No UNIT parameter specified

  The generic unit name to be used for the Accounting and Chargeback DAY2 Audit data set on disk. Specify this parameter according to JCL language syntax.

- o VOLSER=volser

  Default: No VOLSER parameter specified.

  Define the VOLSER parameter if you want the Accounting and Chargeback DAY2 Audit data set written onto a specific disk. Specify this parameter according to JCL language syntax.

ACCTDAY2SMSPARM

Optional parameter for specifying System Managed Storage (SMS) keywords (e.g., STORCLAS=storclas) for Accounting and Chargeback DAY2 Audit data set allocations on disk. Specify one or more SMS parameters according to JCL language syntax, separated by commas. Continuation is not allowed.

The parameters on this statement are effective only when the DISK parameter on the ACCTDAY2PARM statement is specified.

ACCTMTAPEPARM

Use the optional ACCTMTAPEPARM statement to specify parameters defining processing options for Accounting and Chargeback Monthly Audit tape data sets. When using the Monthly Audit tapes, the ACCTMTAPEPARM parameter specifications override the tape data set defaults established by the TAPEPARM statement.

The primary reason to specify the ACCTMTAPEPARM statement is to override the default number of monthly Audit generation data group entries to be defined (the GDG= parameter). However, your site may want to direct the Accounting and Chargeback Monthly Audit to a unique tape volume pool, or to a tape device type such as a virtual tape unit.

Note: Directing the Monthly Audit data set to disk with the TAPEPARM DISK parameter is currently not supported.

The general form of the ACCTMTAPEPARM statement is the keyword ACCTMTAPEPARM followed by one or more ACCTMTAPEPARM parameters. ACCTMTAPEPARM parameters include both keyword parameters and keyword=value specifications. You can specify the ACCTMTAPEPARM statement as many times as you want in order to specify all of the tape data set definitions you require. If you specify a parameter multiple times, the last instance defines the value in effect.

The following summarizes the ACCTMTAPEPARM parameters:

- Each parameter is described in detail next.

- While the following sample shows only one or two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character-maximum line length). You may also choose to specify only one parameter per line. Either approach is acceptable.

```
    ACCTMTAPEPARM GDG=nnn UNIT=unitname VOLCOUNT=n
    ACCTMTAPEPARM EXPDT=expdt / RETPD=nnn / NOEXPDT
    ACCTMTAPEPARM COMP/NOCOMP
    ACCTMTAPEPARM NULLUNIT=unitname NULLSER=nulser
    ACCTMTAPEPARM STACK/NOSTACK AFF/NOAFF
    ACCTMTAPEPARM MODEL='modeldsn' / MODEL=NOMODEL
```

The following is a sample ACCTMTAPEPARM statement:

```
     ACCTMTAPEPARM GDG=90 VOLCOUNT=7
```

In this example, the TAPEPARM parameter values are used for most of the ACCTMTAPEPARM parameters.

The following describes each of the ACCTMTAPEPARM parameters:

- GDG=nnn

  Defaults to 85.

  The number of entries to be created in the GDG for each Accounting and Chargeback Monthly Audit data set.

  Overrides and replaces the ACCTMGDG statement.

- UNIT=unitname

  Defaults to the value specified on the TAPEPARM

statement or to UNIT=3400-5 if UNIT=unitname is not specified on the TAPEPARM statement.

The generic unit name for tape file allocation. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

Overrides the TAPEPARM statement UNIT=unitname value and/or the old-style TAPEUNIT statement.

o VOLCOUNT=n

Defaults to the value specified on the TAPEPARM statement. If no VOLCOUNT is specified, then the OS/390 default volume count of 5 will apply.

The volume count JCL parameter for CA MICS tape data sets. Specify a decimal number from 1 to 255.

Overrides the TAPEPARM statement VOLCOUNT=nnn parameter or the old-style VOLCOUNT statement.

o EXPDT=expdt

Defaults to the value specified on the TAPEPARM statement. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

Specify an expiration date definition which replaces the derived default retention period (RETPD) specification(s). This facility is specifically intended for use by installations with tape management systems which require specific EXPDT definitions.

Note that for some tape management systems, certain EXPDT values will make the tape available for scratching immediately. Such values are NOT desirable for CA MICS tapes.

The EXPDT=expdt parameter is optional, and, if omitted, CA MICS will derive a RETPD=nnn specification as appropriate for the specific

CA MICS tape data set being created and the number of GDG entries defined.

Note: If you specify an EXPDT=expdt, then you must also specify a value for the TMS statement (e.g., TMS NONE).

Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameter, or the old-style EXPDT statement.

Note: The EXPDT=expdt, RETPD=nnn, and NOEXPDT parameters are mutually exclusive.

- NOEXPDT

  Defaults to the value specified on the TAPEPARM statement. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

  Specify NOEXPDT if you want CA MICS to omit both the EXPDT=date and RETPD=days JCL parameters from generated JCL statements for new tape data set creation. This option is provided for those installations where the tape management system or SMS specifications automatically control tape expiration dates and/or retention periods.

  Note: If you specify the NOEXPDT parameter, you must also specify a value for the TMS statement (e.g., TMS NONE).

  Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameters, or the old-style EXPDT statement.

  Note: The NOEXPDT, EXPDT=expdt, and RETPD=nnn parameters are mutually exclusive.

- COMP / NOCOMP

  Defaults to the value specified on the TAPEPARM statement. If neither COMP nor NOCOMP is specified, then the OS/390 default for the selected tape unit type will apply.

  This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) for a tape device enabled for hardware compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified, no TRTCH subparameter will be generated.

Overrides the TAPEPARM statement COMP/NOCOMP parameter or the old-style IDRC statement.

o NULLUNIT=unitname

Defaults to the value specified on the TAPEPARM statement or to the tape unit value.

The generic unit name to be used for tape file allocation for the NULLSER. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

Overrides the TAPEPARM statement NULLUNIT=unitname parameter or the old-style NULLUNIT statement.

o NULLSER=nulser

Defaults to the value specified on the TAPEPARM statement or to NULLSER.

The value to be used to signify a dummy tape volume to CA MICS. This value is used for pre-allocating dummy tape files when the system is first run. This technique allows CA MICS to run without error even though certain input files do not exist. A trap in the macro routine recognizes the NULLSER value and generates the appropriate SET statements. If your installation standards prevent using the default value (NULSER), then you must change this value. An example is in the case of automatic tape loaders (ATLs).

For example, if you are adding a new file to history archive in an existing unit, and the existing history archive tapes are in an ATL, you may need to change your NULLSER value to an actual volume serial number contained in the ATL. This prevents an additional tape device (outside the

ATL) from being allocated, as well as the MOUNT for the volume serial NULSER. This is imperative for JES3 customers with ATLs because the job will not continue until the mount is satisfied.

Overrides the TAPEPARM statement NULLSER=nulser parameter or the old-style NULLSER statement.

- STACK / NOSTACK

  Defaults to the value specified on the TAPEPARM statement or to STACK.

  Specify STACK (the default) to "stack" multiple CA MICS tape files on a single tape volume. This is the normal mode of operation where a minimum number of tape volumes are used and tape mounts are minimized. CA MICS JCL statements are generated with the VOL=REF=ddname construct and unit affinity is enforced (that is, STACK also means AFF).

  Specify NOSTACK to place exactly one CA MICS tape file on a tape volume. With this option the generated CA MICS JCL statements will NOT contain the VOL=REF=ddname construct. You may find this option useful if your implementation of a virtual tape system prohibits "stacked" tape volumes, or if you are directing CA MICS tape files to a DASD pool backed with an archival facility.

  Overrides the TAPEPARM statement STACK/NOSTACK parameter.

- AFF / NOAFF

  Defaults to the value specified on the TAPEPARM statement or to AFF.

  Specify AFF to use a single tape unit for all related tape files. This is the normal mode of operation where a minimum number of tape units are used and tape mounts are minimized. CA MICS JCL statements are generated with the UNIT=AFF=ddname construct. AFF is required for the STACK option.

  Specify NOAFF to enable allocation of different tape units for each CA MICS tape data set. With this option the generated CA MICS JCL statements will NOT contain the UNIT=AFF=ddname construct.

You may find this option useful with certain virtual tape system implementations, or where you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Note, if you specify NOAFF, then NOSTACK is automatically set.

Overrides the TAPEPARM statement AFF/NOAFF parameter.

o MODEL='modeldsn' / MODEL=NOMODEL

Defaults to the value specified on the TAPEPARM statement.

The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted, a CA MICS generation job will allocate a default model GDG data set.

If NOMODEL is specified as the model dataset name, the use of the model dataset in the DCB specification will be bypassed.

Overrides the TAPEPARM statement MODEL='modeldsn' parameter or the old-style SYSTEMMODEL statement.

ACCTMTAPESMSPARM

Defaults to the values specified on the TAPESMSPARM statement.

Optional parameter for specifying System Managed Storage (SMS) keywords (e.g., STORCLAS=storclas) for Accounting and Chargeback Monthly Audit data set allocations. Specify one or more SMS parameters according to JCL language syntax, separated by commas. Continuation is not allowed.

Overrides the TAPESMSPARM statement.

AUDITPARM

Use the optional AUDITPARM statement to specify parameters defining processing options for audit archive tape data sets. When using the audit archive tapes, the AUDITPARM parameter specifications override the tape data set defaults established by the TAPEPARM statement.

The primary reason to specify the AUDITPARM statement is to override the default number of archive audit generation data group entries to be defined (the GDG= parameter). However, your site may want to direct the archive audit tapes to a unique tape volume pool, or to a tape device type such as a virtual tape unit, or to a DASD pool backed by an archive facility. The AUDITPARM statement gives you the flexibility to uniquely define tape data set parameters for archive audit tapes.

The general form of the AUDITPARM statement is the keyword AUDITPARM followed by one or more AUDITPARM parameters. AUDITPARM parameters include both keyword parameters and keyword=value specifications. You can specify the AUDITPARM statement as many times as you want in order to specify all of the tape data set definitions you require. If you specify a parameter multiple times, the last instance defines the value in effect.

The following summarizes the AUDITPARM parameters:

- Each parameter is described in detail next.

- While the following syntax sample shows only one or two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character-maximum line length). You may also choose to specify only one parameter per line. Either approach is acceptable.

```
AUDITPARM GDG=nnn UNIT=unitname VOLCOUNT=n
AUDITPARM EXPDT=expdt / RETPD=nnn / NOEXPDT
AUDITPARM COMP/NOCOMP
AUDITPARM NULLUNIT=unitname NULLSER=nulser
AUDITPARM STACK/NOSTACK AFF/NOAFF
AUDITPARM MODEL='modeldsn' / MODEL=NOMODEL
AUDITPARM DISK / TAPE  SPACE=spacedef
AUDITPARM VOLSER=volser
```

The following is a sample AUDITPARM statement:

```
     AUDITPARM GDG=75 VOLCOUNT=6
```

In this example, the TAPEPARM parameter values are used for most of the AUDITPARM parameters.

The following describes each of the AUDITPARM parameters:

- GDG=nnn

  Defaults to 53.

  The number of entries to be created in the GDG for each archive audit data set. The number of entries control the number of weeks for which archive audit data will be maintained for each CA MICS file. This process will create one tape of data per week, unless you have massive files.

  The recommended value is GDG=53. The calculated default retention period for these files is equal to the number of entries specified, multiplied by 7 (7 days per entry).

  Overrides and replaces the old-style AUDITGDG statement.

  Note: If you choose to split audit archive processing out of the WEEKLY operational job and perform this processing in the stand-alone AUDIT operational job, specify this option:

  ```
       ARCHIVE AUDIT YES JOB
  ```

  You will then be able to execute audit archive processing (the AUDIT job)and AUDIT more frequently (e.g., twice a week or daily) if DASD space is inadequate for retaining sufficient DETAIL/DAYS cycles for weekly audit tape creation.

  In this case, it is necessary to increase the number of tape cycles (or GDG entries) retained in order to meet your requirements for "long-term" retention of audit tape data.

  If you choose this option, then you must add the following option to prefix.MICS.PARMS(EXECDEF)

```
     USERDEF AUDITCWK YES
```

This parameter overrides the default audit archive processing, so that data for the current week is retained and copied to the new audit archive tape cycle.

- o UNIT=unitname

  Defaults to the value specified on the TAPEPARM statement or to UNIT=3400-5 if UNIT=unitname is not specified on the TAPEPARM statement.

  The generic unit name for tape file allocation. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

  Overrides the TAPEPARM statement UNIT=unitname value and/or the old-style TAPEUNIT statement.

- o VOLCOUNT=n

  Defaults to the value specified on the TAPEPARM statement. If no VOLCOUNT is specified, then the OS/390 default volume count of 5 will apply.

  The volume count JCL parameter for CA MICS tape data sets. Specify a decimal number from 1 to 255.

  Overrides the TAPEPARM statement VOLCOUNT=nnn parameter or the old-style VOLCOUNT statement.

- o EXPDT=expdt

  Defaults to the value specified on the TAPEPARM statement. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

  Specify an expiration date definition which replaces the derived default retention period (RETPD) specification(s). This facility is specifically intended for use by installations with tape management systems which require specific

EXPDT definitions.

Note that for some tape management systems, certain EXPDT values will make the tape available for scratching immediately. Such values are NOT desirable for CA MICS tapes.

The EXPDT=expdt parameter is optional, and, if omitted, CA MICS will derive a RETPD=nnn specification as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

Note: If you specify an EXPDT=expdt, then you must also specify a value for the TMS statement (e.g., TMS NONE).

Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameter, or the old-style EXPDT statement.

Note: The EXPDT=expdt, RETPD=nnn, and NOEXPDT parameters are mutually exclusive.

- NOEXPDT

Defaults to the value specified on the TAPEPARM statement. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

Specify NOEXPDT if you want CA MICS to omit both the EXPDT=date and RETPD=days JCL parameters from generated JCL statements for new tape data set creation. This option is provided for those installations where the tape management system or SMS specifications automatically control tape expiration dates and/or retention periods.

Note: If you specify the NOEXPDT parameter, you must also specify a value for the TMS statement (e.g., TMS NONE).

Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameters, or the old-style EXPDT statement.

Note: The NOEXPDT, EXPDT=expdt, and RETPD=nnn parameters are mutually exclusive.

o COMP / NOCOMP

Defaults to the value specified on the TAPEPARM statement. If neither COMP nor NOCOMP is specified, then the OS/390 default for the selected tape unit type will apply.

This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) for a tape device enabled for hardware compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified, no TRTCH subparameter will be generated.

Overrides the TAPEPARM statement COMP/NOCOMP parameter or the old-style IDRC statement.

o NULLUNIT=unitname

Defaults to the value specified on the TAPEPARM statement or to the tape unit value.

The generic unit name to be used for tape file allocation for the NULLSER. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

Overrides the TAPEPARM statement NULLUNIT=unitname parameter or the old-style NULLUNIT statement.

o NULLSER=nulser

Defaults to the value specified on the TAPEPARM statement or to NULLSER.

The value to be used to signify a dummy tape volume to CA MICS. This value is used for pre-allocating dummy tape files when the system is first run. This technique allows CA MICS to run without error even though certain input files do not exist. A trap in the macro routine recognizes the NULLSER value and generates the appropriate SET statements. If your installation standards prevent using the default value (NULSER), then you must change this value. An example is in the case of automatic tape loaders (ATLs).

For example, if you are adding a new file to history archive in an existing unit, and the existing history archive tapes are in an ATL, you may need to change your NULLSER value to an actual volume serial number contained in the ATL. This prevents an additional tape device (outside the ATL) from being allocated, as well as the MOUNT for the volume serial NULSER. This is imperative for JES3 customers with ATLs because the job will not continue until the mount is satisfied.

Overrides the TAPEPARM statement NULLSER=nulser parameter or the old-style NULLSER statement.

o STACK / NOSTACK

Defaults to the value specified on the TAPEPARM statement or to STACK.

Specify STACK (the default) to "stack" multiple CA MICS tape files on a single tape volume. This is the normal mode of operation where a minimum number of tape volumes are used and tape mounts are minimized. CA MICS JCL statements are generated with the VOL=REF=ddname construct and unit affinity is enforced (that is, STACK also means AFF).

Specify NOSTACK to place exactly one CA MICS tape file on a tape volume. With this option the generated CA MICS JCL statements will NOT contain the VOL=REF=ddname construct. You may find this option useful if your implementation of a virtual tape system prohibits "stacked" tape volumes, or if you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Overrides the TAPEPARM statement STACK/NOSTACK parameter.

o AFF / NOAFF

Defaults to the value specified on the TAPEPARM statement or to AFF.

Specify AFF to use a single tape unit for all related tape files. This is the normal mode of operation where a minimum number of tape units are used and tape mounts are minimized. CA MICS JCL statements are generated with the UNIT=AFF=ddname construct. AFF is required for the STACK option.

Specify NOAFF to enable allocation of different tape units for each CA MICS tape data set. With this option the generated CA MICS JCL statements will NOT contain the UNIT=AFF=ddname construct. You may find this option useful with certain virtual tape system implementations, or where you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Note, if you specify NOAFF, then NOSTACK is automatically set.

Overrides the TAPEPARM statement AFF/NOAFF parameter.

o MODEL='modeldsn' / MODEL=NOMODEL

Defaults to the value specified on the TAPEPARM statement.

The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted, a CA MICS generation job will allocate a default model GDG data set.

If NOMODEL is specified as the model dataset name, the use of the model dataset in the DCB specification will be bypassed.

Overrides the TAPEPARM statement MODEL='modeldsn' parameter or the old-style SYSTEMMODEL statement.

o DISK / TAPE

Defaults to TAPE.

Specify DISK if you want to write the dataset to a disk instead of a tape. Specify TAPE to write the dataset to a tape.

When you specify DISK, make sure that you have both a UNIT parameter with a generic unit name of a disk device and a SPACE parameter defining disk space. You can also define a VOLSER parameter to direct the data to a specific disk.

- SPACE=spacedef

  No default.

  Defines the space allocation parameter for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

- VOLSER=volser

  No default.

  Defines the volume serial number of a disk for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

AUDITSMSPARM

Defaults to the values specified on the TAPESMSPARM statement.

Optional parameter for specifying System Managed Storage (SMS) keywords (e.g., STORCLAS=storclas) for archive audit tape data set allocations. Specify one or more SMS parameters according to JCL language syntax, separated by commas. Continuation is not allowed.

Overrides the TAPESMSPARM statement.

BKUPPARM

Use the optional BKUPPARM statement to specify parameters defining processing options for CA MICS backup tape data sets. When using the backup tapes, the BKUPPARM parameter specifications override the tape data set defaults established by the TAPEPARM statement.

The primary reason to specify the BKUPPARM statement is to override the default number of backup generation data group entries to be defined (the GDG= parameter). However, your site may want to direct the CA MICS backup tapes to a unique tape volume pool, or to a tape device type such as a virtual tape unit, or to a DASD pool backed by an archive facility. The BKUPPARM statement gives you the flexibility to uniquely define tape data set parameters for backup tapes.

The general form of the BKUPPARM statement is the keyword BKUPPARM followed by one or more BKUPPARM parameters. BKUPPARM parameters include both keyword parameters and keyword=value specifications. You can specify the BKUPPARM statement as many times as you want in order to specify all of the tape data set definitions you require. If you specify a parameter multiple times, the last instance defines the value in effect.

The following summarizes the BKUPPARM parameters.

- Each parameter is described in detail next.

- While the following syntax sample shows only one or two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character-maximum line length). You may also choose to specify only one parameter per line. Either approach is acceptable.

```
BKUPPARM GDG=nnn UNIT=unitname VOLCOUNT=n
BKUPPARM EXPDT=expdt / RETPD=nnn / NOEXPDT
BKUPPARM COMP/NOCOMP
BKUPPARM NULLUNIT=unitname NULLSER=nulser
BKUPPARM STACK/NOSTACK AFF/NOAFF
BKUPPARM MODEL='modeldsn' / MODEL=NOMODEL
BKUPPARM DISK / TAPE  SPACE=spacedef
BKUPPARM VOLSER=volser
```

The following is a sample BKUPPARM statement:

```
BKUPPARM GDG=10 VOLCOUNT=8
```

In this example, the TAPEPARM parameter values are used for most of the BKUPPARM parameters.

The following describes each of the BKUPPARM parameters.

- GDG=nnn

Defaults to 10.

The number of entries to be created in the GDG for each backup data set. The number of entries controls the number of days for which backup data will be maintained for each CA MICS timespan. This process will create one tape of data per backup, unless you have massive files.

The recommended value is GDG=10. The calculated default retention period for these files is derived from the number of entries specified and the BACKUP FREQ specification.

Overrides and replaces the old BCKUPGDG statement.

o UNIT=unitname

Defaults to the value specified on the TAPEPARM statement or to UNIT=3400-5 if UNIT=unitname is not specified on the TAPEPARM statement.

The generic unit name for tape file allocation. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

Overrides the TAPEPARM statement UNIT=unitname value and/or the old-style TAPEUNIT statement.

o VOLCOUNT=n

Defaults to the value specified on the TAPEPARM statement. If no VOLCOUNT is specified, then the OS/390 default volume count of 5 will apply.

The volume count JCL parameter for CA MICS tape data sets. Specify a decimal number from 1 to 255.

Overrides the TAPEPARM statement VOLCOUNT=nnn parameter or the old-style VOLCOUNT statement.

o EXPDT=expdt

Defaults to the value specified on the TAPEPARM

statement. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

Specify an expiration date definition which replaces the derived default retention period (RETPD) specification(s). This facility is specifically intended for use by installations with tape management systems which require specific EXPDT definitions.

Note that for some tape management systems, certain EXPDT values will make the tape available for scratching immediately. Such values are NOT desirable for CA MICS tapes.

The EXPDT=expdt parameter is optional, and, if omitted, CA MICS will derive a RETPD=nnn specification as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

Note: If you specify an EXPDT=expdt, then you must also specify a value for the TMS statement (e.g., TMS NONE).

Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameter, or the old-style EXPDT statement.

Note: The EXPDT=expdt, RETPD=nnn, and NOEXPDT parameters are mutually exclusive.

o NOEXPDT

Defaults to the value specified on the TAPEPARM statement. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

Specify NOEXPDT if you want CA MICS to omit both the EXPDT=date and RETPD=days JCL parameters from generated JCL statements for new tape data set creation. This option is provided for those installations where the tape management system or SMS specifications automatically control tape expiration dates and/or retention periods.

Note: If you specify the NOEXPDT parameter, you must also specify a value for the TMS statement (e.g., TMS NONE).

Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameters, or the old-style EXPDT statement.

Note: The NOEXPDT, EXPDT=expdt, and RETPD=nnn parameters are mutually exclusive.

- COMP / NOCOMP

  Defaults to the value specified on the TAPEPARM statement. If neither COMP nor NOCOMP is specified, then the OS/390 default for the selected tape unit type will apply.

  This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) for a tape device enabled for hardware compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified, no TRTCH subparameter will be generated.

  Overrides the TAPEPARM statement COMP/NOCOMP parameter or the old-style IDRC statement.

- NULLUNIT=unitname

  Defaults to the value specified on the TAPEPARM statement or to the tape unit value.

  The generic unit name to be used for tape file allocation for the NULLSER. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

  Overrides the TAPEPARM statement NULLUNIT=unitname parameter or the old-style NULLUNIT statement.

- NULLSER=nulser

  Defaults to the value specified on the TAPEPARM statement or to NULLSER.

The value to be used to signify a dummy tape volume to CA MICS. This value is used for pre-allocating dummy tape files when the system is first run. This technique allows CA MICS to run without error even though certain input files do not exist. A trap in the macro routine recognizes the NULLSER value and generates the appropriate SET statements. If your installation standards prevent using the default value (NULSER), then you must change this value. An example is in the case of automatic tape loaders (ATLs).

For example, if you are adding a new file to history archive in an existing unit, and the existing history archive tapes are in an ATL, you may need to change your NULLSER value to an actual volume serial number contained in the ATL. This prevents an additional tape device (outside the ATL) from being allocated, as well as the MOUNT for the volume serial NULSER. This is imperative for JES3 customers with ATLs because the job will not continue until the mount is satisfied.

Overrides the TAPEPARM statement NULLSER=nulser parameter or the old-style NULLSER statement.

o STACK / NOSTACK

Defaults to the value specified on the TAPEPARM statement or to STACK.

Specify STACK (the default) to "stack" multiple CA MICS tape files on a single tape volume. This is the normal mode of operation where a minimum number of tape volumes are used and tape mounts are minimized. CA MICS JCL statements are generated with the VOL=REF=ddname construct and unit affinity is enforced (that is, STACK also means AFF).

Specify NOSTACK to place exactly one CA MICS tape file on a tape volume. With this option the generated CA MICS JCL statements will NOT contain the VOL=REF=ddname construct. You may find this option useful if your implementation of a virtual tape system prohibits "stacked" tape volumes, or if you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Overrides the TAPEPARM statement STACK/NOSTACK parameter.

- AFF / NOAFF

  Defaults to the value specified on the TAPEPARM statement or to AFF.

  Specify AFF to use a single tape unit for all related tape files. This is the normal mode of operation where a minimum number of tape units are used and tape mounts are minimized. CA MICS JCL statements are generated with the UNIT=AFF=ddname construct. AFF is required for the STACK option.

  Specify NOAFF to enable allocation of different tape units for each CA MICS tape data set. With this option the generated CA MICS JCL statements will NOT contain the UNIT=AFF=ddname construct. You may find this option useful with certain virtual tape system implementations, or where you are directing CA MICS tape files to a DASD pool backed with an archival facility.

  Note, if you specify NOAFF, then NOSTACK is automatically set.

  Overrides the TAPEPARM statement AFF/NOAFF parameter.

- MODEL='modeldsn' / MODEL=NOMODEL

  Defaults to the value specified on the TAPEPARM statement.

  The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted, a CA MICS generation job will allocate a default model GDG data set.

  If NOMODEL is specified as the model dataset name, the use of the model dataset in the DCB specification will be bypassed.

Overrides the TAPEPARM statement MODEL='modeldsn' parameter or the old-style SYSTEMMODEL statement.

- DISK / TAPE

  Defaults to TAPE.

  Specify DISK if you want to write the dataset to a disk instead of a tape. Specify TAPE to write the dataset to a tape.

  When you specify DISK, make sure that you have both a UNIT parameter with a generic unit name of a disk device and a SPACE parameter defining disk space. You can also define a VOLSER parameter to direct the data to a specific disk.

- SPACE=spacedef

  No default.

  Defines the space allocation parameter for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

- VOLSER=volser

  No default.

  Defines the volume serial number of a disk for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

BKUPSMSPARM

Defaults to the values specified on the TAPESMSPARM statement.

Optional parameter for specifying System Managed Storage (SMS) keywords (e.g., STORCLAS=storclas) for CA MICS backup tape data set allocations. Specify one or more SMS parameters according to JCL language syntax, separated by commas. Continuation is not allowed.

Overrides the TAPESMSPARM statement.

MBKUPPARM

Use the optional MBKUPPARM statement to specify parameters defining processing options for CA MICS monthly backup tape data sets. When using the monthly backup tapes, the MBKUPPARM parameter specifications override the tape data set defaults established by the TAPEPARM or BKUPPARM statements.

The primary reason to specify the MBKUPPARM statement is to override the default number of archive audit generation data group entries to be defined (the GDG= parameter). However, your site may want to direct the CA MICS monthly backup tapes to a unique tape volume pool, or to a tape device type such as a virtual tape unit, or to a DASD pool backed by an archive facility. The MBKUPPARM statement gives you the flexibility to uniquely define tape data set parameters for CA MICS monthly backup tapes.

The general form of the MBKUPPARM statement is the keyword MBKUPPARM followed by one or more MBKUPPARM parameters. MBKUPPARM parameters include both keyword parameters and keyword=value specifications. You can specify the MBKUPPARM statement as many times as you want in order to specify all of the tape data set definitions you require. If you specify a parameter multiple times, the last instance defines the value in effect.

The following summarizes the MBKUPPARM parameters:

- Each parameter is described in detail next.

- While the following syntax sample shows only one or two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character-maximum line length). You may also choose to specify only one parameter per line. Either approach is acceptable.

```
MBKUPPARM GDG=nnn UNIT=unitname VOLCOUNT=n
MBKUPPARM EXPDT=expdt / RETPD=nnn / NOEXPDT
MBKUPPARM COMP/NOCOMP
MBKUPPARM NULLUNIT=unitname NULLSER=nulser
MBKUPPARM STACK/NOSTACK AFF/NOAFF
MBKUPPARM MODEL='modeldsn' / MODEL=NOMODEL
MBKUPPARM DISK / TAPE  SPACE=spacedef
MBKUPPARM VOLSER=volser
```

The following is a sample MBKUPPARM statement:

```
MBKUPPARM GDG=6 VOLCOUNT=8
```

In this example, the TAPEPARM or BKUPPARM parameter values are used for most of the MBKUPPARM parameters.

The following describes each of the MBKUPPARM parameters:

- GDG=nnn

  Defaults to 6.

  The number of entries to be created in the GDG for each monthly backup data set. The number of entries control the number of months for which monthly backup data will be maintained for each CA MICS timespan. This process will create one tape of data per month, unless you have massive files.

  The recommended value is GDG=6. The calculated default retention period for these files is equal to the number of entries specified, multiplied by 31 (31 days per entry).

  Overrides and replaces the previously hard-coded default value of 6.

- UNIT=unitname

  Defaults to the BKUPPARM specification (or default).

  The generic unit name for tape file allocation. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

  Overrides the BKUPPARM specification (or default).

- VOLCOUNT=n

Defaults to the BKUPPARM specification (or default).

The volume count JCL parameter for CA MICS tape data sets. Specify a decimal number from 1 to 255.

Overrides the BKUPPARM specification (or default).

- EXPDT=expdt

  Defaults to the BKUPPARM specification. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

  Specify an expiration date definition which replaces the derived default retention period (RETPD) specification(s). This facility is specifically intended for use by installations with tape management systems which require specific EXPDT definitions.

  Note that for some tape management systems, certain EXPDT values will make the tape available for scratching immediately. Such values are NOT desirable for CA MICS tapes.

  The EXPDT=expdt parameter is optional, and, if omitted, CA MICS will derive a RETPD=nnn specification as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

  Note: If you specify an EXPDT=expdt, then you must also specify a value for the TMS statement (e.g., TMS NONE).

  Overrides the BKUPPARM specification.

  Note: The EXPDT=expdt, RETPD=nnn, and NOEXPDT parameters are mutually exclusive.

- RETPD=nnn

  Defaults to the BKUPPARM specification. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

Specify a fixed retention period definition which replaces the derived default retention period (RETPD) specification. This facility is specifically intended for use by installations with tape management systems which requires specific RETPD definitions.

Note that for some tape management systems, certain RETPD values will make the tape available for scratching immediately. Such values are NOT desirable for CA MICS tapes.

The RETPD=nnn parameter is optional, and, if omitted, CA MICS will derive a default RETPD=nnn specification as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

Note: If you specify RETPD=nnn, then you must also specify a value for the TMS statement (e.g., TMS NONE).

Overrides the BKUPPARM specification (or default).

Note: The RETPD=nnn, EXPDT=expdt, and NOEXPDT parameters are mutually exclusive.

o NOEXPDT

Defaults to the BKUPPARM specification. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

Specify NOEXPDT if you want CA MICS to omit both the EXPDT=date and RETPD=days JCL parameters from generated JCL statements for new tape data set creation. This option is provided for those installations where the tape management system or SMS specifications automatically control tape expiration dates and/or retention periods.

Note: If you specify the NOEXPDT parameter, you must also specify a value for the TMS statement (e.g., TMS NONE).

Overrides the BKUPPARM specification (or default).

Note: The NOEXPDT, EXPDT=expdt, and RETPD=nnn parameters are mutually exclusive.

- COMP / NOCOMP

  Defaults to the BKUPPARM specification (or default).

  This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) for a tape device enabled for hardware compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified, no TRTCH subparameter will be generated.

  Overrides the BKUPPARM specification (or default).

- NULLUNIT=unitname

  Defaults to the BKUPPARM specification (or default).

  The generic unit name to be used for tape file allocation for the NULLSER. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

  Overrides the BKUPPARM specification (or default).

- NULLSER=nulser

  Defaults to the BKUPPARM specification (or default).

  The value to be used to signify a dummy tape volume to CA MICS. This value is used for pre-allocating dummy tape files when the system is first run. This technique allows CA MICS to run without error even though certain input files do not exist. A trap in the macro routine recognizes the NULLSER value and generates the appropriate SET statements. If your installation standards prevent using the default value (NULSER), then you must change this value. An example is in the case of automatic tape loaders (ATLs).

For example, if you are adding a new file to history archive in an existing unit, and the existing history archive tapes are in an ATL, you may need to change your NULLSER value to an actual volume serial number contained in the ATL. This prevents an additional tape device (outside the ATL) from being allocated, as well as the MOUNT for the volume serial NULSER. This is imperative for JES3 customers with ATLs because the job will not continue until the mount is satisfied.

Overrides the BKUPPARM specification (or default).

o STACK / NOSTACK

Defaults to the BKUPPARM specification (or default).

Specify STACK (the default) to "stack" multiple CA MICS tape files on a single tape volume. This is the normal mode of operation where a minimum number of tape volumes are used and tape mounts are minimized. CA MICS JCL statements are generated with the VOL=REF=ddname construct and unit affinity is enforced (that is, STACK also means AFF).

Specify NOSTACK to place exactly one CA MICS tape file on a tape volume. With this option the generated CA MICS JCL statements will NOT contain the VOL=REF=ddname construct. You may find this option useful if your implementation of a virtual tape system prohibits "stacked" tape volumes, or if you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Overrides the BKUPPARM specification (or default).

o AFF / NOAFF

Defaults to the BKUPPARM specification (or default).

Specify AFF to use a single tape unit for all related tape files. This is the normal mode of operation where a minimum number of tape units are used and tape mounts are minimized. CA MICS JCL statements are generated with the UNIT=AFF=ddname construct. AFF is required for the STACK option.

Specify NOAFF to enable allocation of different tape units for each CA MICS tape data set. With this option the generated CA MICS JCL statements will NOT contain the UNIT=AFF=ddname construct. You may find this option useful with certain virtual tape system implementations, or where you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Note, if you specify NOAFF, then NOSTACK is automatically set.

Overrides the BKUPPARM specification (or default).

o MODEL='modeldsn' / MODEL=NOMODEL

Defaults to the BKUPPARM specification (or default).

The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted, a CA MICS generation job will allocate a default model GDG data set.

If NOMODEL is specified as the model dataset name, the use of the model dataset in the DCB specification will be bypassed.

Overrides the BKUPPARM specification (or default).

o DISK / TAPE

Defaults to the BKUPPARM specification (or default).

Specify DISK if you want to write the dataset to a disk instead of a tape. Specify TAPE to write the dataset to a tape.

When you specify DISK, make sure that you have both a UNIT parameter with a generic unit name of a disk device and a SPACE parameter defining disk space. You can also define a VOLSER parameter to direct the data to a specific disk.

Overrides the BKUPPARM specification (or default).

- SPACE=spacedef

  Defaults to the BKUPPARM specification (or default).

  Defines the space allocation parameter for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

  Overrides the BKUPPARM specification (or default).

- VOLSER=volser

  Defaults to the BKUPPARM specification (or default).

  Defines the volume serial number of a disk for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

  Overrides the BKUPPARM specification (or default).

MBKUPSMSPARM

Defaults to the values specified on (or defaulted for) BKUPSMSPARM statement.

Optional parameter for specifying System Managed Storage (SMS) keywords (e.g., STORCLAS=storclas) for CA MICS monthly backup tape data set allocations. Specify one or more SMS parameters according to JCL language syntax, separated by commas. Continuation is not allowed.

Overrides the BKUPSMSPARM statement (or default).

HISTMPARM

Use the optional HISTMPARM statement to specify parameters defining processing options for Archive Monthly History tape data sets. When using the Archive Monthly History tapes, the HISTMPARM parameter specifications override the tape data set defaults established by the TAPEPARM statement.

The primary reason to specify the HISTMPARM statement is to override the default number of archive monthly history generation data group entries to be defined (the GDG= parameter). However, your site may want to direct the archive monthly history tapes to a unique tape volume pool, or to a tape device type such as a virtual tape unit, or to a DASD pool backed by an archive facility. The HISTMPARM statement gives you the flexibility to uniquely define tape data set parameters for archive monthly history tapes.

The general form of the HISTMPARM statement is the keyword HISTMPARM followed by one or more HISTMPARM parameters. HISTMPARM parameters include both keyword parameters and keyword=value specifications. You can specify the HISTMPARM statement as many times as you want in order to specify all of the tape data set definitions you require. If you specify a parameter multiple times, the last instance defines the value in effect.

The following summarizes the HISTMPARM parameters:

- Each parameter is described in detail next.

- While the following syntax sample shows only one or two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character-maximum line length). You may also choose to specify only one parameter per line. Either approach is acceptable.

```
     HISTMPARM GDG=nnn UNIT=unitname VOLCOUNT=n
     HISTMPARM EXPDT=expdt / RETPD=nnn / NOEXPDT
     HISTMPARM COMP/NOCOMP
     HISTMPARM NULLUNIT=unitname NULLSER=nulser
     HISTMPARM STACK/NOSTACK AFF/NOAFF
     HISTMPARM MODEL='modeldsn' / MODEL=NOMODEL
     HISTMPARM DISK / TAPE  SPACE=spacedef
     HISTMPARM VOLSER=volser
```

The following is a sample HISTMPARM statement:

```
     HISTMPARM GDG=3 VOLCOUNT=7
```

In this example, the TAPEPARM parameter values are used for most of the HISTMPARM parameters.

The following describes each of the HISTMPARM parameters:

- o GDG=nnn

  Defaults to 5.

  The number of entries to be created in the GDG for each archive monthly history data set.

  The recommended value is GDG=5. The calculated default retention period for these files is equal to the number of entries specified, multiplied by 31 (31 days per month).

  Note: This specification is NOT what determines the length of time for which archive monthly history data is kept. See the description of the "cutoff" in the prefix.MICS.PARMS(DBMODEL) discussion in section 2.3.4.1.2, "File Statements."

  Overrides and replaces the old-style HISTMGDG statement.

- o UNIT=unitname

  Defaults to the value specified on the TAPEPARM statement or to UNIT=3400-5 if UNIT=unitname is not specified on the TAPEPARM statement.

  The generic unit name for tape file allocation. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

  Overrides the TAPEPARM statement UNIT=unitname value and/or the old-style TAPEUNIT statement.

- o VOLCOUNT=n

  Defaults to the value specified on the TAPEPARM statement. If no VOLCOUNT is specified, then the OS/390 default volume count of 5 will apply.

  The volume count JCL parameter for CA MICS tape data sets. Specify a decimal number from 1 to 255.

Overrides the TAPEPARM statement VOLCOUNT=nnn parameter or the old-style VOLCOUNT statement.

- EXPDT=expdt

  Defaults to the value specified on the TAPEPARM statement. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

  Specify an expiration date definition which replaces the derived default retention period (RETPD) specification(s). This facility is specifically intended for use by installations with tape management systems which require specific EXPDT definitions.

  Note that for some tape management systems, certain EXPDT values will make the tape available for scratching immediately. Such values are NOT desirable for CA MICS tapes.

  The EXPDT=expdt parameter is optional, and, if omitted, CA MICS will derive a RETPD=nnn specification as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

  Note: If you specify an EXPDT=expdt, then you must also specify a value for the TMS statement (e.g., TMS NONE).

  Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameter, or the old-style EXPDT statement.

  Note: The EXPDT=expdt, RETPD=nnn, and NOEXPDT parameters are mutually exclusive.

- NOEXPDT

  Defaults to the value specified on the TAPEPARM statement. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

  Specify NOEXPDT if you want CA MICS to omit both the EXPDT=date and RETPD=days JCL parameters from generated JCL statements for new tape data set creation. This option is provided for those installations where the tape management system or SMS specifications automatically control tape expiration dates and/or retention periods.

Note: If you specify the NOEXPDT parameter, you must also specify a value for the TMS statement (e.g., TMS NONE).

Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameters, or the old-style EXPDT statement.

Note: The NOEXPDT, EXPDT=expdt, and RETPD=nnn parameters are mutually exclusive.

- COMP / NOCOMP

  Defaults to the value specified on the TAPEPARM statement. If neither COMP nor NOCOMP is specified, then the OS/390 default for the selected tape unit type will apply.

  This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) for a tape device enabled for hardware compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified, no TRTCH subparameter will be generated.

  Overrides the TAPEPARM statement COMP/NOCOMP parameter or the old-style IDRC statement.

- NULLUNIT=unitname

  Defaults to the value specified on the TAPEPARM statement or to the tape unit value.

  The generic unit name to be used for tape file allocation for the NULLSER. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

Overrides the TAPEPARM statement NULLUNIT=unitname parameter or the old-style NULLUNIT statement.

- o NULLSER=nulser

  Defaults to the value specified on the TAPEPARM statement or to NULLSER.

  The value to be used to signify a dummy tape volume to CA MICS. This value is used for pre-allocating dummy tape files when the system is first run. This technique allows CA MICS to run without error even though certain input files do not exist. A trap in the macro routine recognizes the NULLSER value and generates the appropriate SET statements. If your installation standards prevent using the default value (NULSER), then you must change this value. An example is in the case of automatic tape loaders (ATLs).

  For example, if you are adding a new file to history archive in an existing unit, and the existing history archive tapes are in an ATL, you may need to change your NULLSER value to an actual volume serial number contained in the ATL. This prevents an additional tape device (outside the ATL) from being allocated, as well as the MOUNT for the volume serial NULSER. This is imperative for JES3 customers with ATLs because the job will not continue until the mount is satisfied.

  Overrides the TAPEPARM statement NULLSER=nulser parameter or the old-style NULLSER statement.

- o STACK / NOSTACK

  Defaults to the value specified on the TAPEPARM statement or to STACK.

  Specify STACK (the default) to "stack" multiple CA MICS tape files on a single tape volume. This is the normal mode of operation where a minimum number of tape volumes are used and tape mounts are minimized. CA MICS JCL statements are generated with the VOL=REF=ddname construct and unit affinity is enforced (that is, STACK also means AFF).

  Specify NOSTACK to place exactly one CA MICS tape file on a tape volume. With this option the generated CA MICS JCL statements will NOT contain the VOL=REF=ddname construct. You may find this option useful if your implementation of a virtual tape system prohibits "stacked" tape volumes, or if you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Overrides the TAPEPARM statement STACK/NOSTACK parameter.

o AFF / NOAFF

Defaults to the value specified on the TAPEPARM statement or to AFF.

Specify AFF to use a single tape unit for all related tape files. This is the normal mode of operation where a minimum number of tape units are used and tape mounts are minimized. CA MICS JCL statements are generated with the UNIT=AFF=ddname construct. AFF is required for the STACK option.

Specify NOAFF to enable allocation of different tape units for each CA MICS tape data set. With this option the generated CA MICS JCL statements will NOT contain the UNIT=AFF=ddname construct. You may find this option useful with certain virtual tape system implementations, or where you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Note, if you specify NOAFF, then NOSTACK is automatically set.

Overrides the TAPEPARM statement AFF/NOAFF parameter.

o MODEL='modeldsn' / MODEL=NOMODEL

Defaults to the value specified on the TAPEPARM statement.

The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted, a CA MICS generation job will allocate a default model GDG data set.

If NOMODEL is specified as the model dataset name, the use of the model dataset in the DCB specification will be bypassed.

Overrides the TAPEPARM statement MODEL='modeldsn' parameter or the old-style SYSTEMMODEL statement.

o DISK / TAPE

Defaults to TAPE.

Specify DISK if you want to write the dataset to a disk instead of a tape. Specify TAPE to write the dataset to a tape.

When you specify DISK, make sure that you have both a UNIT parameter with a generic unit name of a disk device and a SPACE parameter defining disk space. You can also define a VOLSER parameter to direct the data to a specific disk.

o SPACE=spacedef

No default.

Defines the space allocation parameter for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

o VOLSER=volser

No default.

Defines the volume serial number of a disk for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

HISTMSMSPARM

Defaults to the values specified on the TAPESMSPARM statement.

Optional parameter for specifying System Managed Storage

(SMS) keywords (e.g., STORCLAS=storclas) for archive monthly history tape data set allocations. Specify one or more SMS parameters according to JCL language syntax, separated by commas. Continuation is not allowed.

Overrides the TAPESMSPARM statement.

HISTWPARM

Use the optional HISTWPARM statement to specify parameters defining processing options for Audit Weekly History tape data sets. When using the Audit Weekly History tapes, the HISTWPARM parameter specifications override the tape data set defaults established by the TAPEPARM statement.

The primary reason to specify the HISTWPARM statement is to override the default number of archive weekly history generation data group entries to be defined (the GDG= parameter). However, your site may want to direct the archive weekly history tapes to a unique tape volume pool, or to a tape device type such as a virtual tape unit, or to a DASD pool backed by an archive facility. The HISTWPARM statement gives you the flexibility to uniquely define tape data set parameters for archive weekly history tapes.

The general form of the HISTWPARM statement is the keyword HISTWPARM followed by one or more HISTWPARM parameters. HISTWPARM parameters include both keyword parameters and keyword=value specifications. You can specify the HISTWPARM statement as many times as you want in order to specify all of the tape data set definitions you require. If you specify a parameter multiple times, the last instance defines the value in effect.

The following summarizes the HISTWPARM parameters:

- o Each parameter is described in detail next.

- o While the following syntax sample shows only one or two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character-maximum line length). You may also choose to specify only one parameter per line. Either approach is acceptable.

```
HISTWPARM GDG=nnn UNIT=unitname VOLCOUNT=n
HISTWPARM EXPDT=expdt / RETPD=nnn / NOEXPDT
HISTWPARM COMP/NOCOMP
HISTWPARM NULLUNIT=unitname NULLSER=nulser
HISTWPARM STACK/NOSTACK AFF/NOAFF
HISTWPARM MODEL='modeldsn' / MODEL=NOMODEL
HISTWPARM DISK / TAPE  SPACE=spacedef
HISTWPARM VOLSER=volser
```

The following is a sample HISTWPARM statement:

```
HISTWPARM GDG=3 VOLCOUNT=6
```

In this example, the TAPEPARM parameter values are used for most of the HISTWPARM parameters.

The following describes each of the AUDITPARM parameters:

- GDG=nnn

  Defaults to 7.

  The number of entries to be created in the GDG for each archive weekly history data set.

  The recommended value is GDG=7. The calculated retention period for these files is equal to the number of entries specified, multiplied by 7 (7 days per week).

  Note: This specification is NOT what determines the length of time for which archive monthly history data is kept. See the description of the "cutoff" in the prefix.MICS.PARMS(DBMODEL) discussion in section 2.3.4.1.2, "File Statements."

  Overrides and replaces the old-style HISTWGDG statement.

- UNIT=unitname

  Defaults to the value specified on the TAPEPARM statement or to UNIT=3400-5 if UNIT=unitname is not specified on the TAPEPARM statement.

  The generic unit name for tape file allocation. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

Overrides the TAPEPARM statement UNIT=unitname value and/or the old-style TAPEUNIT statement.

o VOLCOUNT=n

Defaults to the value specified on the TAPEPARM statement. If no VOLCOUNT is specified, then the OS/390 default volume count of 5 will apply.

The volume count JCL parameter for CA MICS tape data sets. Specify a decimal number from 1 to 255.

Overrides the TAPEPARM statement VOLCOUNT=nnn parameter or the old-style VOLCOUNT statement.

o EXPDT=expdt

Defaults to the value specified on the TAPEPARM statement. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

Specify an expiration date definition which replaces the derived default retention period (RETPD) specification(s). This facility is specifically intended for use by installations with tape management systems which require specific EXPDT definitions.

Note that for some tape management systems, certain EXPDT values will make the tape available for scratching immediately. Such values are NOT desirable for CA MICS tapes.

The EXPDT=expdt parameter is optional, and, if omitted, CA MICS will derive a RETPD=nnn specification as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

Note: If you specify an EXPDT=expdt, then you must also specify a value for the TMS statement (e.g., TMS NONE).

Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameter, or the old-style EXPDT statement.

Note: The EXPDT=expdt, RETPD=nnn, and NOEXPDT parameters are mutually exclusive.

- o NOEXPDT

  Defaults to the value specified on the TAPEPARM statement. If no expiration date or retention period is specified, a default retention period is derived based on the GDG=nnn value.

  Specify NOEXPDT if you want CA MICS to omit both the EXPDT=date and RETPD=days JCL parameters from generated JCL statements for new tape data set creation. This option is provided for those installations where the tape management system or SMS specifications automatically control tape expiration dates and/or retention periods.

  Note: If you specify the NOEXPDT parameter, you must also specify a value for the TMS statement (e.g., TMS NONE).

  Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameters, or the old-style EXPDT statement.

  Note: The NOEXPDT, EXPDT=expdt, and RETPD=nnn parameters are mutually exclusive.

- o COMP / NOCOMP

  Defaults to the value specified on the TAPEPARM statement. If neither COMP nor NOCOMP is specified, then the OS/390 default for the selected tape unit type will apply.

  This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) for a tape device enabled for hardware compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified, no TRTCH subparameter will be generated.

Overrides the TAPEPARM statement COMP/NOCOMP parameter or the old-style IDRC statement.

- NULLUNIT=unitname

  Defaults to the value specified on the TAPEPARM statement or to the tape unit value.

  The generic unit name to be used for tape file allocation for the NULLSER. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

  Overrides the TAPEPARM statement NULLUNIT=unitname parameter or the old-style NULLUNIT statement.

- NULLSER=nulser

  Defaults to the value specified on the TAPEPARM statement or to NULLSER.

  The value to be used to signify a dummy tape volume to CA MICS. This value is used for pre-allocating dummy tape files when the system is first run. This technique allows CA MICS to run without error even though certain input files do not exist. A trap in the macro routine recognizes the NULLSER value and generates the appropriate SET statements. If your installation standards prevent using the default value (NULSER), then you must change this value. An example is in the case of automatic tape loaders (ATLs).

  For example, if you are adding a new file to history archive in an existing unit, and the existing history archive tapes are in an ATL, you may need to change your NULLSER value to an actual volume serial number contained in the ATL. This prevents an additional tape device (outside the ATL) from being allocated, as well as the MOUNT for the volume serial NULSER. This is imperative for JES3 customers with ATLs because the job will not continue until the mount is satisfied.

  Overrides the TAPEPARM statement NULLSER=nulser parameter or the old-style NULLSER statement.

- STACK / NOSTACK

  Defaults to the value specified on the TAPEPARM statement or to STACK.

  Specify STACK (the default) to "stack" multiple CA MICS tape files on a single tape volume. This is the normal mode of operation where a minimum number of tape volumes are used and tape mounts are minimized. CA MICS JCL statements are generated with the VOL=REF=ddname construct and unit affinity is enforced (that is, STACK also means AFF).

  Specify NOSTACK to place exactly one CA MICS tape file on a tape volume. With this option the generated CA MICS JCL statements will NOT contain the VOL=REF=ddname construct. You may find this option useful if your implementation of a virtual tape system prohibits "stacked" tape volumes, or if you are directing CA MICS tape files to a DASD pool backed with an archival facility.

  Overrides the TAPEPARM statement STACK/NOSTACK parameter.

- AFF / NOAFF

  Defaults to the value specified on the TAPEPARM statement or to AFF.

  Specify AFF to use a single tape unit for all related tape files. This is the normal mode of operation where a minimum number of tape units are used and tape mounts are minimized. CA MICS JCL statements are generated with the UNIT=AFF=ddname construct. AFF is required for the STACK option.

  Specify NOAFF to enable allocation of different tape units for each CA MICS tape data set. With this option the generated CA MICS JCL statements will NOT contain the UNIT=AFF=ddname construct. You may find this option useful with certain virtual tape system implementations, or where you are directing CA MICS tape files to a DASD pool backed with an archival facility.

  Note, if you specify NOAFF, then NOSTACK is automatically set.

Overrides the TAPEPARM statement AFF/NOAFF parameter.

- MODEL='modeldsn' / MODEL=NOMODEL

  Defaults to the value specified on the TAPEPARM statement.

  The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted, a CA MICS generation job will allocate a default model GDG data set.

  If NOMODEL is specified as the model dataset name, the use of the model dataset in the DCB specification will be bypassed.

  Overrides the TAPEPARM statement MODEL='modeldsn' parameter or the old-style SYSTEMMODEL statement.

- DISK / TAPE

  Defaults to TAPE.

  Specify DISK if you want to write the dataset to a disk instead of a tape. Specify TAPE to write the dataset to a tape.

  When you specify DISK, make sure that you have both a UNIT parameter with a generic unit name of a disk device and a SPACE parameter defining disk space. You can also define a VOLSER parameter to direct the data to a specific disk.

- SPACE=spacedef

  No default.

  Defines the space allocation parameter for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

- VOLSER=volser

  No default.

  Defines the volume serial number of a disk for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

HISTWSMSPARM

Defaults to the values specified on the TAPESMSPARM statement.

Optional parameter for specifying System Managed Storage (SMS) keywords (e.g., STORCLAS=storclas) for CA MICS archive weekly history tape data set allocations. Specify one or more SMS parameters according to JCL language syntax separated by commas. Continuation is not allowed.

Overrides the TAPESMSPARM statement.

AUDITGDG

Replaced by the AUDITPARM statement GDG=nnn parameter.

The number of entries to be created in the GDG for each archive audit data set. The number of entries control the number of weeks for which archive audit data will be maintained for each CA MICS file for which it is kept. This process will create one tape of data per week, unless you have massive files.

The recommended value for the AUDITGDG parameter is 53. The calculated retention period for these files is equal to the number of entries specified, multiplied by 7 (7 days per entry).

BCKUPGDG

Replaced by the BKUPPARM statement GDG=nnn parameter.

The number of generation data group entries to be maintained for the database backup files. The recommended number is 10. The calculated retention period for these files is based upon BACKUP FREQ.

HISTMGDG

Replaced by the HISTMPARM statement GDG=nnn parameter.

The number of generation data group entries to be maintained for the archive monthly history files. The recommended number is 2. The retention period calculated by CA MICS for these files is equal to the number of entries specified, multiplied by 31 (31 days per month).

Note: This specification is NOT what determines the length of time for which archive monthly history data is kept. See the description of the "cutoff" in the prefix.MICS.PARMS(DBMODEL) discussion in section 2.3.4.1.2, "File Statements."

HISTWGDG

Replaced by the HISTWPARM statement GDG=nnn parameter.

The number of generation data group entries to be maintained for the archive weekly history files. The recommended number is 7. The retention period that CA MICS calculates for these files is equal to the number of entries specified, multiplied by 7 (7 days per entry).

Note: This specification is NOT what determines the length of time for which archive weekly history data is kept. See the description of the "cutoff" in the prefix.MICS.PARMS(DBMODEL) discussion in section 2.3.4.1.2, "BAS File Statements."

DETAILTAPEPARM

Use the optional DETAILTAPEPARM statement to specify parameters defining processing options for CA MICS DETAIL tape data sets. When using the DETAIL tapes, the DETAILTAPEPARM parameter specifications override the tape data set defaults established by the TAPEPARM statement.

The primary reason to specify the DETAILTAPEPARM statement is to override parameters such as the default number of generation data group entries to be defined (the GDG= parameter). However, your site may want to direct the DETAIL tapes to a unique tape volume pool, or to a tape device type such as a virtual tape unit, or to a DASD pool backed by an archive facility.

The DETAILTAPEPARM statement gives you the flexibility to define tape data set parameters for DETAIL tapes at the unit level. Note that you can specify unique DETAIL tape parameters at the component and individual data set level with statements in the prefix.PARMS(cccOPS) members (where ccc is the component identifier). For more information, see Chapter 7 of the relevant component guide.

The general form of the DETAILTAPEPARM statement is the keyword DETAILTAPEPARM, followed by one or more DETAILTAPEPARM parameters. DETAILTAPEPARM parameters include both keyword parameters and keyword=value specifications. You can specify the DETAILTAPEPARM statement as many times as you want in order to specify all of the tape data set definitions you require.
If you specify a parameter multiple times, the last instance defines the value in effect.

The following summarizes the DETAILTAPEPARM parameters:

o Each parameter is described in detail below.

o While the following syntax sample shows only one or two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character-maximum line length).

```
DETAILTAPEPARM GDG=nnn UNIT=unitname VOLCOUNT=n
DETAILTAPEPARM EXPDT=expdt / RETPD=nnn / NOEXPDT
DETAILTAPEPARM COMP/NOCOMP
DETAILTAPEPARM STACK/NOSTACK AFF/NOAFF
DETAILTAPEPARM MODEL='modeldsn' / MODEL=NOMODEL
DETAILTAPEPARM DISK / TAPE  SPACE=spacedef
DETAILTAPEPARM VOLSER=volser
```

The following is a sample DETAILTAPEPARM statement:

```
DETAILTAPEPARM GDG=3 VOLCOUNT=7
```

In this example, the default parameter values are used for most of the DETAILTAPEPARM parameters.

The following describes each of the DETAILTAPEPARM parameters:

- GDG=nnn

  Defaults to 7.

  The number of entries to be created in the GDG for each DETAIL tape data set.

  Overrides the TAPEPARM statement GDG=nnn.

- UNIT=unitname

  Defaults to the value specified on the TAPEPARM statement or to UNIT=3400-5 if UNIT=unitname is not specified on the TAPEPARM statement.

  The generic unit name for tape file allocation. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

  Overrides the TAPEPARM statement UNIT=unitname value and/or the old-style TAPEUNIT statement.

- VOLCOUNT=n

  Defaults to the value specified on the TAPEPARM statement. If no VOLCOUNT is specified, the z/OS default volume count of 5 will apply.

  The volume count JCL parameter for CA MICS tape data sets. Specify a decimal number from 1 to 255.

  Overrides the TAPEPARM statement VOLCOUNT=nnn parameter or the old-style VOLCOUNT statement.

- EXPDT=expdt

  Defaults to the value specified on the TAPEPARM statement. If no expiration date or retention period is specified, a default retention period is used which equals the GDG=nnn value.

  Specify an expiration date definition which replaces the derived default retention period (RETPD) specification(s). This facility is specifically intended for use by installations with tape management systems which require specific EXPDT definitions.

Note that for some tape management systems, certain EXPDT values will make the tape available for scratching immediately. Such values are NOT desirable for CA MICS tapes.

The EXPDT=expdt parameter is optional, and, if omitted, CA MICS will derive a RETPD=nnn specification as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

Note: If you specify an EXPDT=expdt, then you must also specify a value for the TMS statement (e.g., TMS NONE).

Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameter, or the old-style EXPDT statement.

Note: The EXPDT=expdt, RETPD=nnn, and NOEXPDT parameters are mutually exclusive.

o   NOEXPDT

Defaults to the value specified on the TAPEPARM statement.

Specify NOEXPDT if you want CA MICS to omit both the EXPDT=date and RETPD=days JCL parameters from generated JCL statements for new tape data set creation. This option is provided for those installations where the tape management system or SMS specifications automatically control tape expiration dates and/or retention periods.

Note: If you specify the NOEXPDT parameter, you must also specify a value for the TMS statement (e.g., TMS NONE).

Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameters, or the old-style EXPDT statement.

Note: The NOEXPDT, EXPDT=expdt, and RETPD=nnn parameters are mutually exclusive.

o COMP / NOCOMP

Defaults to the value specified on the TAPEPARM statement. If neither COMP nor NOCOMP is specified, the z/OS default for the selected tape unit type will apply.

This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) for a tape device enabled for hardware compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified, no TRTCH subparameter will be generated.

Overrides the TAPEPARM statement COMP/NOCOMP parameter or the old-style IDRC statement.

o STACK / NOSTACK

Defaults to NOSTACK for DETAIL tapes.

Specify STACK to place multiple CA MICS tape files on a single tape volume. This is a mode of operation where a minimum number of tape volumes are used and tape mounts are minimized. CA MICS JCL statements are generated with the VOL=REF=ddname construct and unit affinity is enforced (that is, STACK also means AFF).

Specify NOSTACK (the default) to place exactly one CA MICS tape file on a tape volume. With this option the generated CA MICS JCL statements will NOT contain the VOL=REF=ddname construct. This option is useful if your implementation of a virtual tape system prohibits "stacked" tape volumes, or if you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Overrides the TAPEPARM statement STACK/NOSTACK parameter.

o AFF / NOAFF

Defaults to the value specified on the TAPEPARM statement or to AFF.

Specify AFF to use a single tape unit for all related tape files. This is the normal mode of operation where a minimum number of tape units are used and tape mounts are minimized. CA MICS JCL statements are generated with the UNIT=AFF=ddname construct. AFF is required for the STACK option.

Specify NOAFF to enable allocation of different tape units for each CA MICS tape data set. With this option the generated CA MICS JCL statements will NOT contain the UNIT=AFF=ddname construct. You may find this option useful with certain virtual tape system implementations, or where you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Note, if you specify NOAFF, then NOSTACK is automatically set.

Overrides the TAPEPARM statement AFF/NOAFF parameter.

o MODEL='modeldsn' / MODEL=NOMODEL

Defaults to the value specified on the TAPEPARM statement.

The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted, a CA MICS generation job will allocate a default model GDG data set.

If NOMODEL is specified as the model dataset name, the use of the model dataset in the DCB specification will be bypassed.

Overrides the TAPEPARM statement MODEL='modeldsn' parameter or the old-style SYSTEMMODEL statement.

o DISK / TAPE

Defaults to TAPE.

Specify DISK if you want to write the dataset to a disk instead of a tape. Specify TAPE to write the dataset to a tape.

When you specify DISK, make sure that you have both a UNIT parameter with a generic unit name of a disk device and a SPACE parameter defining disk space. You can also define a VOLSER parameter to direct the data to a specific disk.

Overrides the TAPEPARM statement DISK/TAPE parameter.

o SPACE=spacedef

No default.

Defines the space allocation parameter for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

Overrides the TAPEPARM statement SPACE= parameter.

o VOLSER=volser

No default.

Defines the volume serial number of a disk for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

Overrides the TAPEPARM statement VOLSER= parameter.

DETAILTAPESMSPARM

Use the optional DETAILTAPESMSPARM statement to specify System Managed Storage (SMS) keywords (e.g., STORCLAS=storclas) to be used for DETAIL tape data sets. When using DETAIL tapes, the DETAILTAPESMSPARM parameter specifications override the SMS defaults established by the TAPESMSPARM statement. More than one parameter must be separated by commas.

The DETAILTAPESMSPARM statement gives you the flexibility to define tape data set SMS parameters for DETAIL tapes at the unit level. Note that you can specify unique DETAIL tape SMS parameters at the component and individual data set level with statements in the prefix.PARMS(cccOPS) members (where ccc is the component identifier). For more information, see Chapter 7 of the relevant component guide.

TAPEUNIT

Replaced by the TAPEPARM statement UNIT=unitname parameter.

The generic unit name to be used for tape file allocation. The values allowed for this parameter are 2400, 2400-1, 2400-2, 2400-3, 2400-4, 3400-2, 3400-3, 3400-4, 3400-5, 3400-6, 3400-9, and 3480. The default value is 3400-5. Specify the parameter ASIS if a non-device-dependent name (for example, TAPE) is used at your data center. A sample TAPEUNIT statement is:

```
TAPEUNIT TAPE ASIS
```

VOLCOUNT

Replaced by the TAPEPARM statement VOLCOUNT=nnn parameter.

The volume count is to be used for the backup and archive tape data sets. It is a decimal number from 1 to 255. If the VOLCOUNT keyword is omitted or is null, the default volume count of 5 will be used.

IDRC

Replaced by the TAPEPARM statement COMP/NOCOMP parameter.

This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) on a tape device enabled for compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified or left blank, no TRTCH subparameter will be generated.

TMS

This parameter identifies the tape management system in use at your data center. It is required to allow CA MICS to test for EXPDT values that could cause CA MICS tapes to be accidentally scratched. Valid values are YES, the name of a tape management system, or NONE.

If you specify the TAPEPARM statement EXPDT=expdt or NOEXPDT parameter, or the old-style EXPDT statement, then you must also specify a value for TMS (for example, TMS NONE).

EXPDT

Replaced by the TAPEPARM statement EXPDT=expdt or NOEXPDT parameters.

Specify an expiration date definition that replaces the standard retention period (RETPD) definitions for the tape files used in backup, archive audit, archive weekly history, and archive monthly history. This facility is specifically directed for use by data centers that use a tape management system that requires specific EXPDT definitions. Note that for some tape management systems, certain EXPDT values will make the tape available for scratching immediately. Such values are not desirable for CA MICS tapes. This statement is an option, and if omitted, will result in the standard retention period (calculated in the manner described in the previous section) being used on the tape creation DDs. However, if you do specify an EXPDT value, you must also specify a value for the TMS parameter (for example, TMS NONE).

SYSTEMMODEL

Replaced by the TAPEPARM statement MODEL='modeldsn'/ MODEL=NOMODEL parameter.

The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted (or left blank), a CA MICS generation job will allocate a model GDG data set.

If NOMODEL is specified as the model data set name, the use of the model data set in the DCB specification will be bypassed.

DENSITY

This old-style statement is rarely used and is retained for compatibility with older tape devices.

The tape density to be used for the backup and archive tape data sets (3 for 1600 bpi or 4 for 6250 bpi).

NULLSER

The value to be used to signify a dummy tape volume to CA MICS. This value is used for the archive history tape files when the system is initially run. It allows CA MICS to run without error even though the files do not exist. This is accomplished by a trap in the macro routine that recognizes the NULLSER value and generates the appropriate SET statements. The default value for this parameter is NULSER. If your site standards prevent using the NULSER value for this parameter, you must change this value.

An example is in the case of the automatic tape loaders (ATLs). If you are adding a new file to history archive tapes in an ATL, you may need to change your NULLSER value to an actual volume serial number contained in the ATL. This prevents an additional tape device (outside the ATL) from being allocated, as well as MOUNT for the volume serial NULSER. This is imperative for JES3 customers with ATLs because the job will not continue until the mount is satisfied.

NULLUNIT

The generic unit name to be used for tape file allocation for the NULLSER. If not specified, the value of TAPEUNIT is used. The values allowed for this parameter are 2400, 2400-1, 2400-2, 2400-3, 2400-4, 3400-2, 3400-3, 3400-4, 3400-6, 3400-9, and 3480. Specify the parameter "ASIS" if a non-device-dependent name (e.g., TAPE) is used at your site. A sample NULLUNIT statement:

```
NULLUNIT TAPE ASIS
```

The following parameters are used to specify the catalog that contains the high level qualifier in the TAPEPREFIX. Either CVOLUNIT and CVOLSER must be specified and the VCAT parameters omitted (or left blank), or the VCAT parameters must be specified and the CVOL parameters omitted. In addition to building the GDG entries in the catalog specified by these parameters, the CA MICS generation jobs will allocate a model GDG data set on the volume containing the catalog if SYSTEMMODEL has not been specified.

CVOLUNIT

The generic unit name for the device on which the control volume catalog for the CA MICS data set resides. This parameter is mutually exclusive with VCATNAME.

CVOLSER

The volume serial of the volume on which the control volume catalog for the CA MICS data set resides. This parameter is mutually exclusive with VCATNAME.

VCATNAME

The name of the VSAM or ICF catalog in which the CA MICS data sets are to be cataloged, or SYSTEM. The SYSTEM keyword requires that the CA MICS tape prefix be an alias in the master catalog, or that you have update authority against the master catalog. SYSTEM is the default. This parameter is mutually exclusive with CVOLSER.

VCATSER

The volume serial number of the VSAM or ICF catalog in which the CA MICS data sets are to be cataloged. This parameter is mutually exclusive with CVOLSER. This parameter is required if VCATNAME has a value other than SYSTEM. If SYSTEM has been specified, then the value coded here will only be used in allocating the model data set.

VCATUNIT

The generic unit name of the VSAM or ICF catalog in which the CA MICS data sets are to be cataloged. This parameter is required if VCATNAME has a value other than SYSTEM. If SYSTEM has been specified then the value of this parameter will only be used in allocating the model data set. If no value is provided for VCATUNIT then DASDUNIT will be used instead.

VCATOWNER

The owner of the VSAM or ICF catalog in which the CA MICS data sets are to be cataloged. This parameter is optional and mutually exclusive with CVOLSER. This parameter is ignored if VCATNAME has a value of SYSTEM.

VCATPASSWORD

The password of the VSAM or ICF catalog in which the CA MICS data sets are to be cataloged. This password must provide update access, and is only required if needed. It is mutually exclusive with CVOLSER. This parameter is ignored if VCATNAME has a value of SYSTEM.

2.3.3.2.1.6 Database Unit Extended JOB Statement Definitions

The following parameter statements are identical in purpose to those found in sharedprefix.MICS.PARMS(JCLDEFC), described in section 2.3.1.6.5.

```
ACCT        CLASS       GROUP       JOBNAME
JCLLIB      JOB7380     MSGCLASS    MSGLEVEL
NOTIFY      PASSWORD    PERFORM     PROG
PRTY        REGION      USER
```

Extended JOB statement parameters let you individually define each of the supported JOB statement options as listed below. Keep in mind that the JOB statement may be supplied at the beginning of MICS.PARMS(USERJCL), which is described in section 2.3.3.2.2, Job and JES Definitions. If you choose to have CA MICS construct JOB statements, specify these parameters as you require.

CA MICS supports an option to hard-code a JOB statement in MICS.PARMS(USERJCL) instead of having CA MICS generate one from the parameters described above. All the parameters listed above may be used as symbolic variables whose values may be referenced, along with those of certain other CA MICS variables, in JCL that you supply in the USERJCL member in prefix.MICS.PARMS. For details, see section 2.3.3.2.2, Job and JES Definitions (USERJCL).

2.3.3.2.1.7 Database Unit User Defined Symbols

USERDEF

This parameter allows the user to define a symbol and a value that will be substituted for the symbol by JCLGEN. This can be used in a user modification to PROTOLIB to provide user defined symbolic elements. Several USERDEF symbols may be defined. The only restriction is that the symbol on each USERDEF statement must be different from those on all others and different from all symbols currently used by JCLGEN. Also, the value must end before column 73 of the record. For example,

```
   USERDEF DIGITS 0123456789
```

would define a symbol DIGITS. Anywhere this symbol appeared in a PROTOLIB member, JCLGEN would substitute the string 0123456789 in its place.

Note that PROTOLIB is shared by all databases in a complex, so any user symbols which appear in a member of PROTOLIB must be defined in prefix.MICS.PARMS(JCLDEF) for all of the database units which may reference that PROTOLIB member.

2.3.3.2.1.8 JCLGEN Parameter Sharing

JCLGEN Parameter Sharing allows commonly used JCLGEN parameters to be specified in sharedprefix.MICS.PARMS(JCLDEFC), where they then become default parameters for prefix.MICS.PARMS(JCLDEF) at the unit level.

Use of this facility can greatly reduce the amount of coding in the JCLDEF and JCLDEFC members.

JCLGEN parameters eligible for sharing can be organized into the following groups:

```
    SAS Data Set Definitions:
    ------------------------
    SASNAME     - SAS program name
    SASCONFIGx  - SAS configuration member
    SASAUTOS    - SAS auto include library
    SASMSG      - SAS message data set
    SASLIB      - SAS executable library
    SASVER      - SAS version
    SASHELP     - SAS help library

    ISPF Data Set Definitions:
    -------------------------
    ISPTLIBx    - ISPF table data set
    ISPLLIBx    - ISPF load library
    ISPMLIBx    - ISPF message data set
    ISPPLIBx    - ISPF panel data set
    ISPSLIBx    - ISPF skeleton data set

    Miscellaneous Definitions:
    -------------------------
    ASSEMBLER   - IBM Assembler level
    COLIBSPACE  - Size of the temporary MCOLIB
    COMPLEXPARMS- Allow/disallow JCLGEN parameter sharing
    DIRSTATS    - ISPF stats for generated members
    HFSROOT     - Name of the CA MICS HFS path
    JCLLIB      - Include JCLLIB statement with job card
    MICSDMDELIM - Delimiter character for MICSDM/MICSDU procs
    OBJACCESS   - Access mode for MCOLIB and MUOLIB
    OBJWAIT     - Wait time for MCOLIB and MUOLIB allocation
    PROCLIB     - Location of CA MICS PROCs
    SORTLIB     - SORT utility library
    SORTUTILITY - Installation sort utility
    SORTWKNO    - Number of sort work data set allocations
    STEPCPULIMIT- Limit CPU time for any step to 60 minutes
    STEPLIBx    - User-defined STEPLIBs
    SYSTEMMODEL - Name of model data for GDG creation
    USEROBJLIB  - User object definition library
```

JCLGEN Parameter Sharing is always active, but it must be enabled by specifying the following in sharedprefix.MICS.PARMS(JCLDEFC) or prefix.MICS.PARMS(JCLDEF):

```
COMPLEXPARMS YES
```

When specified in JCLDEFC, the value of COMPLEXPARMS is the default for all units. If specified in JCLDEF, COMPLEXPARMS overrides the COMPLEXPARMS action specified in JCLDEFC.

Section 5.5 provides an optional checklist that shows how to activate JCLGEN parameter sharing for all units in the CA MICS complex.

Notes:

o For the ISPF data set and the STEPLIB definitions, specifying the first definition (that is, ISPTLIB1) will automatically blank out the second definition (that is, ISPTLIB2). Therefore, you must always specify these definitions in numeric order.

o COMPLEXPARMS must be the very first statement in JCLDEFC and JCLDEF. If it is not, an error message is generated.

2.3.3.2.2 Job and JES Definitions (USERJCL)

The USERJCL parameter member exists at both the complex (sharedprefix) and unit (prefix) levels. Use the USERJCL parameter member to add JCL comments and/or JES control statements prior to the first EXEC JCL statement of each job generated by JCLGEN. A worksheet for coding USERJCL is provided in Figure 2-23. You may also supply a JOB card to be used by CA MICS rather than have it generate one from the parameters described in the previous section. (You have a limited symbolic-parameter substitution capability in specifying these cards, as described below.)

For example, if your installation supported a special //PROCLIB control card and also a /*PASSWORD JES2 control card, they could both be incorporated in every job generated by JCLGEN by placing the following in MICS.PARMS(USERJCL):

```
    //@ USER DEFINED JCL
    /@PASSWORD XYZ
    //PROCLIB DD DISP=SHR,DSN=&PROCLIB
```

Symbols beginning with an "&" will be substituted immediately by the JCLGEN processor. They will not be present in the final generated job stream. If a symbol is meant to be present in the generated member, it must be prefixed with a question mark (?REGION) which will be translated to an "&" (&REGION) before the member is written.

The JOB card will be created in one of two ways. It can be automatically created by JCLGEN from the Extended Job card Definitions described in the previous section or from the specification you place into MICS.PARMS(USERJCL). If your site has standards defining the order or placement of statements on a job card, the second method of JOB card creation must be used. When the first line in the USERJCL member is "JOBCARD INCLUDED," JCLGEN will not automatically create a job card. This allows you to define the exact order and placement of JOB parameters in the USERJCL member. The "JOBCARD INCLUDED" keywords must start in column 1.

The JOB card can be created with constant data or with a combination of constant data and symbolic references to the values defined in the extended JOB card definitions defined in Section 2.3.3.2.1.6, "Database Unit Extended Jobcard Definitions."

The simplest example of a defined job card would have the minimum required parameters for the installation provided as constants. For example:

```
   JOBCARD INCLUDED
   //&JOBNAME JOB (10,21,44248,SCM),'MURPHY',CLASS=A
   /@JOBPARM LINECT=0,LINES=20,COPIES=1
```

Note the reference to &JOBNAME. This allows CA MICS to supply the JOBNAME appropriate to the job (e.g., DAILY) being generated. The generated name in the symbol "&JOBNAME" will include any JOBNAME prefix you defined in JCLDEF/JCLDEFC.

If you used all of the parameters from the extended JOB card definitions (in a JES2 installation), the USERJCL member could contain:

```
   JOBCARD INCLUDED
   :SET SEQ=&JOB7380
   //&JOBNAME JOB (&ACCOUNT),'&PROG',CLASS=&CLASS,
   // GROUP=&GROUP,MSGCLASS=&MSGCLASS,MSGLEVEL=(&MSGLEVEL),
   // PRTY=&PRTY,NOTIFY=&NOTIFY,PASSWORD=&PASSWORD,
   // PERFORM=&PERFORM,RD=NC,REGION=&REGION,TIME=(&JOBTIME),
   // USER=&USER,RESTART=*
   /@JOBPARM LINECT=0,LINES=&JOBLINES,COPIES=&JOBCOPIES
```

Since some of the extended JOB card definition variables are long, abbreviations have been provided. Using the abbreviations in a JES3 installation, the USERJCL member would contain:

```
   JOBCARD INCLUDED
   :SET SEQ=&JOB7380
   //&JOBNAME JOB (&A),'&PRO',GROUP=&G,
   // MSGCLASS=&MSC,MSGLEVEL=(&MSL),NOTIFY=&N,PASSWORD=&PA,
   // PERFORM=&PE,PRTY=&PRT,RD=NC,REGION=&RE,TIME=(&T),
   // USER=&U,RESTART=*
   //@MAIN LINES=(&LINES,WARNING),CLASS=&C
   //@FORMAT PR,DDNAME=,COPIES=&COPIES
```

Note that if a parameter must be supplied in columns 73 through 80 of the first JOB card, the JOB7380 parameter must be used. Columns 73 through 80 of the USERJCL member are ignored. To be used, the JOB7380 parameter must be assigned to the sequence field (&SEQ) by the user. This is done by including a Generation Control Language statement immediately following the JOBCARD INCLUDED statement. The statement which must be included is:

```
              :SET SEQ=&JOB7380
```

Three of the variables in the above examples (ie. &JOBTIME, &JOBLINES, and &JOBCOPIES) are not from definitions from JCLDEF. These three variables are defined for each job in member JCLINFO (refer to Section 2.3.3.3.2.2, "Job/CLIST Attribute Table").

If you define your own JOB card, remember the following:

1) Sufficient room must be left on each line for the substitution of symbolic variables. The easiest way to do this is to just code one job card parameter on each line, if this is allowed by your JCL standards.

2) The JES statements normally produced by JCLGEN will be also be omitted along with the JOB card; therefore, the generation of these statements becomes your responsibility.

3) Generation Control Language statements may be used in the member USERJCL (GCL is discussed in Section 2.3.3.3.3, "Generation Control Language").

An example of using Generation Control Language statements in USERJCL follows.

An installation requires the MONTHLY job, in their PRIMARY unit database, to have a unique accounting parameter (ACCT) that differs from the normal JCLDEF value.

```
JOBCARD INCLUDED
:SET SEQ=&JOB7380
:IF &DATABASE NE PRIMARY THEN GOTO SKIPA
:IF &MEMBER = MONTHLY THEN
:SET ACCT = &STR(AC01,DP25)
:SKIPA:
//&JOBNAME  JOB  (&ACCT),'&PROG',CLASS=&CLASS,
//  MSGCLASS=&MSGCLASS,PRTY=&PRTY,
//  MSGLEVEL=(&MSGLEVEL),NOTIFY=&NOTIFY
/@JOBPARM LINES=&JOBLINES
//PROCLIB DD DISP=SHR,DSN=&PROCLIB
```

```
+--------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET: User Inserted JCL Definitions        |
|                                                                          |
| PARMS Library Member is USERJCL                                          |
| Reference Section:  2.3.3.2.2                                            |
+--------------------------------------------------------------------------+
|                                                                          |
| This member is only required if the installation's job stream JCL        |
| conventions require special JCL (e.g. PROCLIB, PASSWORD) to be           |
| inserted in each batch job stream generated in the JCL generation        |
| process. If required by installation standards, a JOB card can be        |
| defined in this member.  If this is done, the first statement,           |
| (appearing even before any comment cards),  must be the words            |
| "JOBCARD INCLUDED".  This prevents JCLGEN from creating a JOB card.      |
| In most installations, no JCL statements nor comments may appear         |
| before the jobcard.                                                      |
|                                                                          |
|                                                                          |
| //@ START OF USER DEFINED JCL STATEMENTS                                  |
|                                                                          |
| ________________________________________________________________________ |
|                                                                          |
| ________________________________________________________________________ |
|                                                                          |
| ________________________________________________________________________ |
|                                                                          |
| ________________________________________________________________________ |
|                                                                          |
| ________________________________________________________________________ |
|                                                                          |
| ________________________________________________________________________ |
|                                                                          |
| ________________________________________________________________________ |
|                                                                          |
| ________________________________________________________________________ |
|                                                                          |
| ________________________________________________________________________ |
|                                                                          |
| ________________________________________________________________________ |
|                                                                          |
| ________________________________________________________________________ |
|                                                                          |
| //@ END OF USER DEFINED JCL STATEMENTS                                    |
|                                                                          |
| ....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+--------------------------------------------------------------------------+
```

Figure 2-23. USERJCL Definition Worksheet

2.3.3.2.3 Input DD Statements for SMF (INPUTRDR)

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       |
|                                                                    |
| PARMS 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 PARMS(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=______________________    |
|                                                                    |
|                                                                    |
+--------------------------------------------------------------------+
```

2.3.3.3 JCLGEN Parameters for Special Requirements

If you need to tailor a member provided with or created by CA MICS more than is allowed by the tailoring described in Section 2.3.3.2, Standard JCLGEN Parameters, this section discusses additional tailoring facilities.

Before you undertake the task of further tailoring JCLGEN, you need to understand the logic of the CA MICS JCL generation process. The JCLGEN processes run in batch under control of the TMP (terminal monitor program -- essentially a batch TSO session). A standard CA MICS JCL procedure, MICSDM, is used for all complex-level JCLGEN runs except the first, JCLGEN0. Because MICSDM does not exist before JCLGEN0 is run, it is coded as an instream PROC within the JCLGEN0 job.

At the unit level, JCLGEN processing uses the MICSDUx JCL procedure for the unit. Like at the complex-level, the first unit-level JCLGEN job (JCLGENA) uses an instream PROC within the JCLGENA job because the MICSDUx PROC does not exist until after the JCLGENA job executes.

In all cases, the JCLGEN jobs invoke SAS and read the SAS statements specified in the PDS member identified on the //SYSIN DD statement for the JOB. The specific JCLGEN processing being invoked is defined through the //GENLIST DD statement.

The SAS program JCLGEN01 uses the information provided in two SAS macro variables (COMPLEX and GENLIST), together with parameters from MICS.PARMS and other CA MICS data sets, to perform the JCL generation process.

It is possible for the CA MICS JCL Generator (JCLGEN) to perform a wide range of generation activity in one execution. This activity is different between complex and unit level generations. The control flow of JCLGEN execution is summarized by the steps listed below. A "C" indicates this function is performed for complex level JCLGENs only, while a "U" indicates that the function is only for unit level JCLGENs.

```
    1. Rebuild statement lookup table ($JCLGEN).

    2. Process JCL definition statements

 C      sharedprefix.MICS.PARMS(JCLDEFC)  or
```

```
U           prefix.MICS.PARMS(JCLDEF)

   3. Process complex/unit information in GENLIB(DBTABLE).

U  4. Get name and title from PARMS(EXECDEF) and
      PARMS(SITE).

   5. Process component definitions from GENLIB(COMPTDEF).

U  6. Process prefix.MICS.PARMS(DBSPLIT).

U  7. Build $STEP and $USTEP formats.

   8. Read (shared)prefix.MICS.PARMS(JCLNAMES).

   9. Create name resolution tables for symbolic variables.

U 10. Prepare update for prefix.MICS.USER.SOURCE(PREFIX).

C 11. Process sharedprefix.MICS.PARMS(CPLXDEF).

  12. Process sharedprefix.MICS.PARMS(JCLINFO,JCL$Iccc) and
U     prefix.MICS.PARMS(JCLINFO).

  13. Read list of jobs/CLISTs to be generated.

  14. Convert PROTOLIB/PARMS to SAS files.

  15. Generate GENLIST items using GCL and symbolic
      variables.

  16. Update libraries.

  17. Build special macros in MACAUTOS.
```

The following description of each step in the JCLGEN process expands on the operation of the program.

1. Rebuild the statement lookup table.

Sharedprefix.MICS.SOURCE(JCLGEN02) is included and executes PROC FORMAT to build member $JCLGEN in sharedprefix.MICS.MCOLIB. The format is used to specify the edit checks to be performed on the various statements contained in the JCLDEF or JCLDEFC members.

2. Process the JCL Generator definitions.

The parameter library JCL definition member is examined to obtain the user specifications for JCL planning parameters. For the complex level jobs, this member is

```
    sharedprefix.MICS.PARMS(JCLDEFC)
```

and that member contains control definitions, library definitions, JCL definitions, and extended job card definitions. The contents of this member are completely described in Section 2.3.1.6.

For other JCLGEN jobs (at the database unit level), this member is

```
    prefix.MICS.PARMS(JCLDEF)
```

which contains control definitions, library definitions, JCL definitions, extended job card definitions, and CA MICS option definitions. The contents of this member are completely described in Section 2.3.3.2.1.

If a JCLDEF DD card is present in the JCLGEN job, then any definition specified here will override values in JCLDEFC or JCLDEF.

For either member being processed and the JCLDEF DD (if present), all statements are read and reflected on the MICSLOG. Any semantic or syntax check that can be performed on an individual statement is done as the statement is read. Error and warning messages from these checks are reflected on the MICSLOG immediately after the definition statement.

After all statements have been processed, logic checks are performed. These include making sure required parameters have been specified and checking parameter interdependencies. For example, the maximum PRTY that can be specified for JES3 is 14. Any error or warning messages that result from these checks will be listed on the MICSLOG after all the definition statements are listed. If there are any errors, processing is terminated at this point with a U998 ABEND.

If no errors have occurred, the JCL parameter table in sharedprefix.MICS.ISPTLIB is updated (or created if necessary). This table contains the JCL definitions for use by some online functions. There is one table for complex level definitions (MAFJPT) and one table for each unit (MAFJPTx, where x is the unit identifier).

The final activity in this step of the JCLGEN process only occurs for unit generations. The list of requested components and the list of SMFRECORDING components is sorted, and duplicate definitions are eliminated.

3. Process complex/unit information in GENLIB(DBTABLE).

There is a table of installed database units contained in sharedprefix.MICS.GENLIB(DBTABLE). This member contains two master entries and one entry per database unit.

The first master entry, designated by having "*: " in positions one through three, contains the unit identifier of the primary unit. This entry will only exist if a primary unit has been defined. The second master entry, designated by having "*:: " in positions one through four, contains the sharedprefix and the maintenance level of the JCL generation processor that last updated DBTABLE. The maintenance level indicator is checked by the load process for new component distributions.

For each unit that has been defined in the complex, there is one record in DBTABLE. It contains the unit's prefix, name, identifier, the name of the PARMS library for the unit, the unit type, and the maintenance level of JCLGEN that last updated the record. A sample DBTABLE member, with all items defined, is:

```
  *: S
  *:: CPM981.CMN  MICS.  BAS6000
  DPM981.DB1  CPMPROD  S  DPM981.DB1.MICS.PARMS    P  BAS5075
  DPM981.IMS  IMSPROD  I  DPM981.IMS.MICS.PARMS    U  BAS6000
  DPM981.TEST TEST     T  DPM981.TEST.MICS.PARMS   T  BAS5075
```

DBTABLE is read during each JCLGEN. For complex level generations, the sharedprefix in JCLDEFC is checked against the sharedprefix in the master entries. For unit JCLGENs, the sharedprefix is taken from the master entry.

The unit description records in DBTABLE are used to check for duplicate definitions. The unit prefix, database name, and unit identifier must be unique. In addition, there can be only one PRIMARY database unit in a complex. For JCLGENs other than the first in a unit, checks are made to ensure that the unit prefix, database name, and unit identifier have not changed. If any of these checks fail, the JCLGEN process will be terminated.

When processing DBTABLE, the JCLGEN process produces a report on the MICSLOG describing the unit definitions. If no primary unit has been defined, a warning message is issued that the complex level data libraries will not be backed up.

The unit description records will be sorted by database name and an IEBUPDTE file to replace the DBTABLE member will be written to the FT15F003 work file.

4. Get name and title from PARMS(EXECDEF) and PARMS(SITE).

This step in the JCLGEN process is only performed for unit level generations. Prefix.MICS.PARMS(EXECDEF) is read to find the TITLE statement. Prefix.MICS.PARMS(SITE) is read to find the NAME statement. These values are used to update the ICFDBTDB ISPF table.

5. Process component definitions from GENLIB(COMPTDEF).

Sharedprefix.MICS.GENLIB(COMPTDEF) contains descriptive information on components.

For complex level JCLGENs, the only information that is needed from COMPTDEF is the list of all valid components that might be in the complex.

For unit level JCLGENs, the processing is more complex. The component, information area, and file statements are matched against the list of components being generated in the unit. For those components and the component BAS, the following information is gathered:

o The list of components defined in the unit, which is used for later translation in the GCL generation process

o The components' step names, which are extracted for the CA MICS update processes

o The information area names

o The files defined in the information areas

o The components that are eligible for:

  - Archive Audit
  - Archive Weekly History
  - Archive Monthly History

Prefix.MICS.PARMS(DBCONFIG) contains an extract of the information from GENLIB(COMPTDEF) for components that are defined for the unit, combined with configuration information gathered later in the JCLGEN process from DBSPLIT and the name translation process. The beginning of DBCONFIG is built while processing COMPTDEF by writing an IEBUPDTE stream to FT15F004.

Another process performed while processing DBCONFIG is the creation/update of some of the ISPF tables used by the online processes. The following tables are created/updated:

- MAFDBCx - where x is the unit identifier
- ICFDBTDB
- MAF$DB

MICSLOG messages are produced to give the defined components, requested components, SMF recording components, non-SMF components, and the components that will be in each of the archive processes.

6. Process prefix.MICS.PARMS(DBSPLIT).

The processing of DBSPLIT is performed only for unit level generations. For more information on the specification of DBSPLIT, refer to Section 2.3.3.3.2.4.

To provide the capability to have multiple SAS data libraries for a timespan, CA MICS has a parameter to split a timespan by information area. The definitions are read from prefix.MICS.PARMS(DBSPLIT) and validated syntactically. The DBSPLIT information is then merged with the information area definitions previously extracted from COMPTDEF and matched to ensure that only information areas for components defined in the unit are specified in the DBSPLIT definition.

As the DBSPLIT and information area definitions are being merged, the information area records for PARMS(DBCONFIG) are added to the previous information maintained on FT15F004.

7. Build $STEP and $USTEP formats.

For unit level JCLGENs, the step information extracted from COMPTDEF for the components defined in this unit are sorted to find the predecessor and successor steps for each step in the CA MICS update processes (DAILY, INCRccc, WEEKLY, MONTHLY, YEARLY, BACKUP, and RESTORE). A report is produced via the MICSLOG that provides this information while the $STEP format is built in prefix.MICS.MUOLIB.

Names are built to define the DAILY, INCRccc, WEEKLY, MONTHLY, and YEARLY steps for the symbolic variable resolution process later.

Prefix.MICS.USER.SOURCE(USTEP) is %INCLUDEd to allow you the capability to override the standard CA MICS step names. For more information on this process, refer to Section 2.3.3.3.6.

8. Read (shared)prefix.MICS.PARMS(JCLNAMES).

The parameter library JCL name list member is examined to get the user specifications for data set and procedure name modifications. For the complex level jobs, this member is sharedprefix.MICS.PARMS(JCLNAMES), while for unit level jobs, the member is prefix.MICS.PARMS(JCLNAMES).

That member contains overrides for the default data set and procedure names that JCLGEN would generate. Each keyword parameter in the Generation Control Language that represents a data set name has a default pattern for combining the prefix, CA MICS level, and low-order descriptive nodes to construct the data set name. JCLNAMES normally contains no statements, because all data set names and procedure names have default values. Default values for each data set name, illustrating the effective contents of the JCLNAMES member, are contained in the member ALLNAMES in either PARMS library.

For more information on the modification of data set names and procedure names, see Section 2.3.3.3.2.3 in this guide.

9. Create name resolution tables for symbolic variables.

One of the main functions of JCLGEN is to translate the symbolic names contained in the PROTOLIB and PARMS members that are being generated. A main part of this translation process is done later by looking up symbolic variables in the $KEYWORD SAS format. This step in the JCLGEN process builds that format.

First, shareprefix.MICS.SOURCE(JCLPARM) is read to get the defaults for all symbolic variables. Then a SAS merge is performed of the JCLPARM data, the JCLNAMES information, the parameters from the JCL definition statements in JCLDEF or JCLDEFC, the DD names from DBSPLIT and COMPTDEF, and the component list from COMPTDEF. The SAS merge is done in the order listed so that duplicate definitions will be taken from the item listed last.

The translations for some symbolic variables are taken from other sources. These variables are items such as SYSDATE and SYSTIME, taken from the system; and SYSUID, taken from the name of the GENLIST.

While building the format for symbolic variable lookup, two ISPF tables (ICFDSI and MAFDSNx, where x is the unit identifier) are built that contain the name translation information used by CA MICS online processes. The definition of which names to place in these tables is controlled by sharedprefix.MICS.SOURCE(JCLPARM).

10. Prepare update for prefix.MICS.USER.SOURCE(PREFIX).

If the generation is a unit level JCLGEN, an IEBUPDTE stream is built in FT15F006 that will add/update the member PREFIX to prefix.MICS.USER.SOURCE.

11. Process sharedprefix.MICS.PARMS(CPLXDEF).

For complex level JCL generations, the CPLXDEF member is read and validated. An update for sharedprefix.MICS.MACAUTOS(CPLXDEF) is prepared in FT15F006 to define the CPLXDEF macro that specifies the options for ESSENTIAL and DERIVED support.

12. Process (shared)prefix.MICS.PARMS(JCLINFO) and
    sharedprefix.MICS.PARMS(JCL$Iccc).

Both JCLINFO and all sharedprefix.MICS.PARMS(JCL$Iccc) members are processed to get the component, jobname, time, lines, class, priority, and PROTOLIB member name for each possible generation request. The generic component jobs (___CHECK, ___EVA, ___CGEN,and ___PGEN) in JCL$IBAS are expanded for each possible component. For more information on the format and content of the JCLINFO member, refer to Section 2.3.3.3.2.2.

13. Read list of jobs/CLISTs to be generated.

There are two ways to specify the list of jobs to generate.

First, A list of jobs can be specified on the GENLIST DD.

Second, a GENLIST parameter is passed to the JCL generation process as a SAS macro variable. The GENLIST parameter is used as a member name in the appropriate MICS.PARMS library to read the list of items to be generated. If there are no items to be generated, the generation process is terminated normally and a message is produced to the MICSLOG. When this happens, the only updates that have occurred are the generation of the $STEP and $USTEP formats and the updates to the various ISPF tables.

The list of GENLIST items to be generated is merged with the information from JCLINFO. If no specific line item is available in JCLINFO for the item to be generated, an error message is written on the MICSLOG and processing of the item is bypassed.

For unit level generations, a match is performed against the component specification from JCLINFO and the components to be defined for the unit. If an item is for a specific component (i.e., not BAS) that is not defined in the unit, the generation of that item is ignored.

14. Convert PROTOLIB/PARMS to SAS files.

Sharedprefix.MICS.PROTOLIB and the MICS.PARMS libraries are partitioned data sets. Both of these are read, their noncomment lines are loaded to SAS files, and PROC FORMATS are built to define the starting and ending record numbers in the SAS file for each member.

15. Generate GENLIST items using GCL and symbolic variables.

For each GENLIST item to be generated, the appropriate member of PROTOLIB (as defined in JCLINFO) is processed. Generation Control Language (GCL) statements are interpreted and acted upon. This can cause the processing of additional members of PROTOLIB and/or the appropriate MICS.PARMS library. See Section 2.3.3.3.3 for further information on GCL.

Symbolic variables are resolved during the generation process. Because it is possible for new symbolic variables to be defined during the process and for previously defined symbolic variables to take on new values, a table is maintained for dynamic symbolic variables. Some positions of this table are filled during the initialization of this step of the JCL generation process. While translating symbolic variables, the dynamic table is searched first. If the variable is not found in the dynamic table, the $KEYWORD format (created in step 9 above) is used to resolve symbolic variables. If the symbolic variable is not found in the format, an error message is produced and generation processing terminates with the end of the current member.

GENLIST items are not updated directly into CA MICS libraries. Updates for sharedprefix.MICS.CNTL are written to FT15F001 file in IEBUPDTE format; updates for sharedprefix.MICS.MACAUTOS or prefix.MICS.USER.SOURCE are written to FT15F006; while updates for sharedprefix.MICS.CLIST are written to FT15F002.

16. Update libraries.

If all processes have been successful to this point, the updates to the CA MICS libraries are made. The updates are done by reading the various IEBUPDTE streams (FT15F00x work files) that have been created. The updates listed below will be made as a batch. Batching these updates reduces the time the data sets are reserved.

Work File	Destination	Comments
FT15F001	MICS.CNTL	Jobs
FT15F002	sharedprefix.MICS.CLIST	CLISTs
FT15F003	sharedprefix.MICS.GENLIB	DBTABLE
FT15F004	MICS.PARMS	DBCONFIG
FT15F005	sharedprefix.MICS.SOURCE	
FT15F006	prefix.MICS.USER.SOURCE	unit only
	sharedprefix.MICS.MACAUTOS	complex only

17. Build special macros in MACAUTOS.

Finally, ISPF file tailoring is used to create/update the DBVERFY, DSNAME, DDNx, DDIAMPx, and DSNx macros in sharedprefix.MICS.MACAUTOS, where x is the unit identifier for unit level JCLGENs or not present for complex level JCLGENs. The DSNAME macro is also built with file tailoring.

If you have any questions about the facilities described in this section, please contact CA MICS Product Support.

The following sections describe points at which you can intervene in the JCLGEN process described above to get the outputs you desire.

1 - Prototype Library (PROTOLIB)
2 - JCLGEN Definition Tables
3 - Generation Control Language
4 - OS Data Set Names
5 - OS Job Names
6 - OS Step Names
7 - OS Job Attributes
8 - Writing A User Exit

2.3.3.3.1 Prototype Library (PROTOLIB)

The purpose of modifying this library is to allow your changes to be automatically carried forward during subsequent CA MICS JCL generations. You must make any required changes and extensions to the MICS.PROTOLIB member used to create the member you actually wish to modify, rather than to the generated member itself.

The prototype library contains JCL job streams, JCL procedures, MACROs, and CLISTs members in symbolic form. The JCLGEN process reads these prototype members and, using your definitions from various MICS.PARMS members and CA MICS-provided definition tables, substitutes the appropriate values for the members' symbols. In addition, any Generation Control Language (GCL) statements imbedded in the prototype member will be executed. The completed members are stored in the MICS.CNTL, MICS.MACAUTOS, MICS.USER.SOURCE, or MICS.CLIST libraries.

Figure 2-25 lists the sharedprefix.MICS.PROTOLIB members that are not specified directly by a GENLIST member. They are included by other prototype members or are named in the JCLINFO or JCL$Iccc members. In any case, they do not have a one-to-one correspondence with any member in the target libraries.

CA MICS maintenance to the prototype library will be applied through normal PSP product changes. Any user modifications to the prototype library members must be applied using IEBUPDTE and saved in MICS.LOCALMOD.CNTL so they may be reapplied as needed.

Some prototype members are generated during component generation and stored by the CYCLEGEN job in prefix.MICS.PARMS for the sake of keeping them unit-specific. Any modifications you make to these members will have to be reapplied following the CYCLEGEN job. The generated PARMS members are marked shown in Figure 2-26.

PROTOLIB MEMBER	JOBNAME	DESCRIPTION
CCCCGEN	cccCGEN	Component Generation Jobs
CCCCHECK	cccCHECK	Component CHECK Jobs
CCCEVA	cccEVA	Component Exection Valuation Jobs
CCCPGEN	cccPGEN	Component Parameter Generation Jobs
COMPDATA	DAILY	GCL to Create Daily Work Files
COMPINCL	DAILY	GCL for REPEAT INCLUDE by Component
DBDDS		GCL for Data Base DD Statements
DBTPDDS		GCL for BACKUP Tape DD Statements
DYcccsss	DAILY	Component Step for Each DAILY Job
DYRPT400	DAILYRPT	Component Report Step for DAILY Job
DYUSR500	DAILY	USER Step for DAILY Job
ISPF		GCL to define ISPF environment
ISPSTART		Start SAS in batch TMP environment
JCLGEN	JCLGENx	GCL for Unit Level JCLGEN Jobs
JCLGENC	JCLGENn	GCL for Complex Level JCLGEN Jobs
MICSASM		GCL to create SYS1.PROCLIB member
MICSCOM		GCL to create SYS1.PROCLIB member
MICSDB		GCL to create SYS1.PROCLIB member
MICSDM		GCL to create SYS1.PROCLIB member
MICSDOC		GCL to create SYS1.PROCLIB member
MICSDU		GCL to create SYS1.PROCLIB member
MICSGT		GCL to create SYS1.PROCLIB member
MICSGU		GCL to create SYS1.PROCLIB member
MICSNDB		GCL to create SYS1.PROCLIB member
MICSRPTS		GCL to create SYS1.PROCLIB member
MICSSHR		GCL to create SYS1.PROCLIB member
MNcccsss	MONTHLY	Component Step for Each MONTHLY Job
MNRPT400	MONTHRPT	Component Report Step for MONTHLY Job
MNUSR500	MONTHLY	User Step for MONTHLY Job
SAS		GCL to define SAS/CA MICS environment
STEPLIB		GCL to define STEPLIB DD statement
WKcccsss	WEEKLY	Component Step for Each WEEKLY Job
WKRPT400	WEEKRPT	Component Report Step for WEEKLY Job
WKUSR500	WEEKLY	USER Step for WEEKLY Job
YRcccsss	YEARLY	Component Step for Each YEARLY Job
YRUSR500	YEARLY	USER Step for YEARLY Job

Note: most PROTOLIB member names correspond to the name of the output member generated from this library. Only the exceptional names are listed in this table.

Figure 2-25. Non-GENLIST Members in MICS.PROTOLIB

```
+-----------------------------------------------------------+
| PARMS                                                     |
| MEMBER     JOBNAME         DESCRIPTION                    |
|-----------------------------------------------------------|
| MNccc300   MONTHLY   Tape Allocation for MONTHLY History  |
| WKccc300   WEEKLY    Tape Alloc for WEEKLY Audit/History  |
+-----------------------------------------------------------+
|  Note:  These unit-level PARMS library members are        |
|  generated by a CYCLEGEN job.  Any modifications made to  |
|  them will be lost after the next CYCLEGEN run.           |
+-----------------------------------------------------------+
```

Figure 2-26. Generated Members in prefix.MICS.PARMS

2.3.3.3.1.1 Making a Simple JCL Change in MICS.PROTOLIB

Suppose, for the sake of this example, that you started getting SYSTEM=80A (insufficient virtual region size) abends in the DAYSMF step of the CA MICS DAILY job. You note that an unmodified CA MICS generates the following EXEC card for this job step:

```
     //DAYSMF EXEC MICSDB,RSIZE=750K
```

You decide, perhaps after consultation with the CA MICS Product Support Group, that 1024K is needed in your shop. If you modified prefix.MICS.CNTL(DAILY), the change would have to be repeated after the next JCLGEN, so you decide to take advantage of the flexibility afforded by MICS.PROTOLIB.

From the explanation of the logic of JCLGEN in Section 2.3.3.3, you determine that you must look at the JCL$Iccc members in sharedprefix.MICS.PARMS to find out which PROTOLIB member is used in the generation of prefix.MICS.CNTL(DAILY). In JCL$IBAS you find the following entry:

```
*
*                                    C
*                                 L  O          C
*                              T  I  P          L  P
*                PROTO         I  N  I          A  R
*          ------------------  M  E  E          S  T
* MEMBER   TYPE COMP  MBRNAME  E  S  S UJOBNAME S  Y
*======== ===== ==== ======== == === = ======== = ==
    .       .     .      .     .  .  .
    .       .     .      .     .  .  .
    .       .     .      .     .  .  .
 DAILY     CNTL  BASE       . 30  80 1        . .  .
```

The absence of a MBRNAME specification tells you that the PROTOLIB member to be used is "DAILY." Examining sharedprefix.MICS.PROTOLIB(DAILY), you find the following line:

```
     //DAYSMF EXEC &MICSDB,RSIZE=750K       ...      00064000
```

You prepare a local modification member in sharedprefix.MICS.LOCALMOD.CNTL using the IEBUPDTE utility to update this member of PROTOLIB (thus observing the standard CA MICS modification practice of not using an online editor):

```
//jobname  JOB  ....
//UPDATE   EXEC PGM=IEBUPDTE
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DISP=OLD,DSN=sharedprefix.MICS.PROTOLIB
//SYSUT2   DD   DISP=OLD,DSN=sharedprefix.MICS.PROTOLIB
//SYSIN    DD   DATA,DLM=ZZ
./ CHANGE NAME=DAILY
:* U00001|0592|INCREASE DAYSMF REGION TO 1024K    */ 00002801
//DAYSMF EXEC &MICSDB,RSIZE=1024K,SYSPARM=&SYSPARM   00064000
ZZ
```

The first change is a notation to the user comment block at the beginning of the member.

After running this update job and verifying that it functioned properly, you must regenerate the DAILY job by running the JCLGENU job stream.

Edit prefix.MICS.PARMS(JCLGENU) and enter a line that reads:

```
  DAILY
```

Now submit prefix.MICS.CNTL(JCLGENU) to regenerate the DAILY member in prefix.MICS.CNTL.

2.3.3.3.2 JCLGEN Definition Tables

Standard JCLGEN parameters are contained in the MICS.PARMS members JCLDEFC, JCLDEF, USERJCL and the various INPUTccc and WORKccc members described in prior sections.

Another set of special definitions, provided with the distributed CA MICS system, are contained in definition tables in MICS.PARMS. The generation lists (JCLGENx), JOB information (JCLINFO and JCL$Iccc), data set name override (JCLNAMES), and database location (DBSPLIT) tables do not normally require modification. They do, however, provide extensive tailoring of JCLGEN's output should the standard JCLGEN parameters not address installation-unique requirements.

Each is explained in the following sections.

1 - Job/CLIST Generation Lists
2 - Job/CLIST Attribute Table (JCLINFO/JCL$Iccc)
3 - CA MICS User Names Table (JCLNAMES)
4 - CA MICS Database Split Table (DBSPLIT)

2.3.3.3.2.1 Job/CLIST Generation Lists

The GENLIST parameter specified in the SYSIN data stream of the JCLGEN process, as described in Section 2.3.3.3, specifies a member in MICS.PARMS where JCLGEN will find a list of jobs and/or CLISTs to be created. The following GENLISTs are shipped with CA MICS: members JCLGEN0 through JCLGEN2 and JCLGENUC in sharedprefix.MICS.PARMS and JCLGENA through JCLGENG and JCLGENU in prefix.MICS.PARMS. They are used by jobs stored under the same member names in the corresponding MICS.CNTL library. The contents of the GENLISTs are documented in Section 2.3.3.1.1, "Processing Done by Standard JCLGEN Jobs."

Use sharedprefix.MICS.PARMS(JCLGENUC) to specify which items you want generated at the complex level. Use prefix.MICS.PARMS(JCLGENU) to specify unit level generations.

Use one of these GENLISTs, instead of the standard GENLISTs, to reduce the number of members generated. Only those members modified in MICS.PROTOLIB; or those members that would be affected by changes made in JCLDEFC, JCLDEF, JCLINFO, JCL$Iccc, JCLNAMES, or DBSPLIT; would have to be regenerated. This way you reduce the JCLGEN run time and reduce the number of members that have to be reviewed by your production change control process. For example, to regenerate the DAILY, WEEKLY, and MONTHLY job streams, you do the following:

```
     edit prefix.MICS.PARMS(JCLGENU)

          DAILY
          WEEKLY MONTHLY

     sub 'prefix.MICS.CNTL(JCLGENU)'
```

Specify one or multiple member names on each statement. In the example above, only members DAILY, WEEKLY, and MONTHLY would be generated in prefix.MICS.CNTL.

2.3.3.3.2.2 Job/CLIST Attribute Table (JCLINFO/JCL$Iccc)

The Job/CLIST Attribute Table is split into two types of members. The first have names that begin with JCL$I and are delivered only in sharedprefix.MICS.PARMS. They define the known CA MICS jobs, CLISTs, and macros that may be generated by the JCLGEN process. The second is a member named JCLINFO, which is provided for your use. JCLINFO is found in both sharedprefix and prefix.MICS.PARMS and should be used to hold your local overrides to our definitions as found in the JCL$I members. JCLGEN will use these members to:

- Validate the job, MACRO, or CLIST to be created
- Identify the library to be updated
- Specify the true PROTOLIB member name if different from the GENLIST name
- Define the TIME, CLASS, PRIORITY, and unique JOBNAME to be placed on the job card
- Define the number of LINES and/or COPIES to be requested on the JES parameter cards

You can use this table in a number of ways. For example, CLASS and PRTY can be used to override for individual jobs the overall CLASS and PRTY specifications from JCLDEF or JCLDEFC.

Note: The CA MICS defaults for TIME, defined in the JCL$I members, are based on an IBM 3033. If you run on a faster machine, the TIME entries in this table are adequate.

If you supply the job card and/or JES parameters either via the USERJCL member or the facilities provided by MICS.PROTOLIB, then you are responsible for the proper specification of TIME, LINES, COPIES, and JOBNAME. Refer to Section 2.3.3.2.2, Job and JES Definitions (USERJCL), and Section 2.3.3.3.1, Prototype Library (PROTOLIB), for more information about these facilities.

The Job/CLIST Attribute Tables are free-form but positional; missing fields must be denoted by a period. Comments are specified by starting the line with an asterisk.

Note: Do not make changes to the JCL$I members. Instead, make your changes in the JCLINFO members provided. In this way, JCLINFO will be a record of your local modifications to the delivered CA MICS values.

Changes to JCLINFO in sharedprefix.MICS.PARMS will affect the complex and all unit JCLGENs unless an entry is also specified in the JCLINFO member in prefix.MICS.PARMS. Changes made to prefix.MICS.PARMS(JCLINFO) member will override all values defined in sharedprefix.MICS.PARMS members including JCLINFO.

The information contained in this table includes:

MEMBER - The search argument used to extract information from the table. The MEMBER named "DEFAULT" sets the values to be used when a match is not found in the table. JCLGEN obtains the search argument to match against this field from the "GENLIST" (see Section 2.3.3.3.2.1, Job/CLIST Generation Lists). The member may be specified with three underscores as the first three characters. In this case, the match will be decided based on the remaining characters and the first three characters of a component defined for the unit.

TYPE - The library into which the generated member will be saved. This may be CLIST, MACRO, or CNTL.

COMP - The CA MICS component to which the generated member belongs. This field is used by the JCL generation process to select the jobs to be generated based on the components defined in the COMPONENTS statement in JCLDEF. Since the BAS component is in every unit, a value of "BAS" in the COMP field will always cause this member to be eligible for generation.

MBRNAME - The name of the member in MICS.PROTOLIB that will be used for the generation of this member. If missing, the default name is the same as the MEMBER name.

The following fields are used only if TYPE=CNTL:

TIME - Value to be used for the job card TIME parameter.

LINES - Value to be used for the JES LINES parameter.

COPIES - Value to be used for the JES COPIES parameter.

UJOBNAME - JOBNAME to be used on the job card. If missing, the default is the value specified in MEMBER.

CLASS - Value to be used for CLASS parameter. For JES2, the CLASS parameter is placed on the job card; for JES3, the class parameter is placed on the *MAIN card.

PRTY - Value to be used for the job card PRTY parameter.

Figure 2-27 shows some entries of the Job/CLIST Attribute Table.

```
+---------------------------------------------------------+
| PARMS(JCL$IBAS):                      C                 |
|                                    L  O          C      |
|                                 T  I  P          L  P   |
|                   PROTO         I  N  I          A  R   |
|             ------------------  M  E  E          S  T   |
|    MEMBER   TYPE COMP  MBRNAME  E  S  S UJOBNAME S  Y   |
+---------------------------------------------------------+
|  * GENERIC COMPONENT JOBS                               |
|   ___CHECK  CNTL    . CCCCHECK 15  20 1        . .  .   |
|   ___EVA    CNTL    . CCCEVA   15  25 1        . .  .   |
|   ___CGEN   CNTL    . CCCCGEN  20  30 1        . .  .   |
|   ___PGEN   CNTL    . CCCPGEN  10  10 1        . .  .   |
|  * BASE JOBS                                            |
|   ALLOCATE  CNTL BAS         .  1   5 1        . .  .   |
|   ALLPGEN   CNTL BAS         . 20  10 2        . .  .   |
|   CKPTINIT  CNTL BAS         .  1   5 1        . .  .   |
|   CYCLEGEN  CNTL BAS         .  2   5 1        . .  .   |
|   DAILY     CNTL BAS         . 30  80 1        . .  .   |
|   DAILYRPT  CNTL BAS  DYRPT400 15  19 1        . .  .   |
|   DAYSMFR   CNTL BAS         . 15  19 1        . .  .   |
|   DBSIZE    CNTL BAS         .  2   5 1        . .  .   |
|   DOCUMENT  CNTL BAS         .  2 200 1        . .  .   |
|   EXCFULL   CNTL BAS         . 15  80 1        . .  .   |
|   GDGSGEN   CNTL BAS         .  3   7 1        . .  .   |
|   JCLGEND   CNTL BAS    JCLGEN  5  20 1        . .  .   |
+---------------------------------------------------------+
+---------------------------------------------------------+
| sharedprefix.MICS.PARMS(JCL$IACT):                      |
|                                       C                 |
|                                    L  O          C      |
|                                 T  I  P          L  P   |
|                   PROTO         I  N  I          A  R   |
|             ------------------  M  E  E          S  T   |
|    MEMBER   TYPE COMP  MBRNAME  E  S  S UJOBNAME S  Y   |
+---------------------------------------------------------+
|   ACTDAY1R  CNTL  ACT        . 10   5 1        . .  .   |
|   ACTUPDTB  CNTL  ACT        . 10   5 1        . .  .   |
|   ACTINITA  CNTL  ACT        .  1   1 1        . .  .   |
|   ACTLGRSP  CNTL  ACT        . 10   5 1        . .  .   |
|   ACTPRORT  CNTL  ACT        . 10   5 1        . .  .   |
|   ACTRTSIM  CNTL  ACT        . 10  10 1        . .  .   |
|   BGTCNTL   CNTL  ACT        .  5   5 1        . .  .   |
|   CLOSETBL  CNTL  ACT        .  5   5 1        . .  .   |
|   FINRPTU   CNTL  ACT        . 10  30 1        . .  .   |
|   INVOICEU  CNTL  ACT        . 10  30 1        . .  .   |
+---------------------------------------------------------+
```

Figure 2-27. Sample Job/CLIST Attribute Table

2.3.3.3.2.3 CA MICS User Names Table (JCLNAMES)

The CA MICS User Names Table allows alternatives to be specified for the data set and procedure names used by CA MICS to satisfy two requirements. First, an installation may require names different than the defaults used by CA MICS to satisfy installation standards. For example, CA MICS uses a final qualifier of "LOAD" for data sets that contain executable load modules. JCLNAMES provides the capability to change the name of those data sets to end in "LOADLIB" or anything else the installation may require.

The second requirement satisfied by JCLNAMES is the changing of the CA MICS procedure names. This may be required by installation standards, or to support multiple complexes in an environment where multiple PROCLIBs are not available.

If you do not HAVE to change the data set or procedure names, do not do it. It is your responsibility to ensure that the changes you make in JCLNAMES are reflected in every unit for a complex. Also, JCL distributed for loading tapes for new components or PSP tapes will contain the default data set names and procedure names.

The CA MICS User Names Table consists of a set of statements in member JCLNAMES of sharedprefix.MICS.PARMS and prefix.MICS.PARMS. This table is free-form. Comments are specified by starting the line with an asterisk.

Two fields are defined in the table. The first field is the symbol used by JCLGEN to represent the name. The second field is the value to be assigned. The value may be a constant or may also begin with one of the following combinations of symbolic variables:

```
&SHAREDPREFIX..&SMICS
&SHAREPREFIX..&SMICS
&PREFIX..&PMICS
&TAPEPREFIX..&TMICS
```

As distributed, the User Names Table is empty. This is done to reduce processing time during generation. Figure 2-28 shows a sample of the CA MICS User Names Table with names defined. In actual use, only the names that did not conform to the CA MICS defaults would be in the table. Member ALLNAMES of sharedprefix.MICS.PARMS and prefix.MICS.PARMS contains all of the names that may be specified in JCLNAMES.

If alternate procedure names are specified, be sure to retain the symbolic &PROCIDENTIFIER as shown in the example below. This is to ensure that different procedure names are used for different units. Also, after the PROCJOBn jobs are run to move procedures to your PROCLIB, you must manually change the PROCLIB member name of a procedure with an alternate name.

```
 *   SAMPLE JCLNAMES
 *
 *   SHARED DATA SETS
 *
  ASM              &SHAREDPREFIX..&SMICS.ASM
  BIN              &SHAREDPREFIX..&SMICS.BIN
  CAPACITY         &SHAREDPREFIX..&SMICS.CAPACITY
  DTFINDEX         &SHAREDPREFIX..&SMICS.DTF.INDEX
  CLIST            &SHAREDPREFIX..&SMICS.CLIST
  GENLIB           &SHAREDPREFIX..&SMICS.GENLIB
  HPARMS           &SHAREDPREFIX..&SMICS.HOLD.PARMS
  HUSOURCE         &SHAREDPREFIX..&SMICS.HOLD.USER.SOURCE
  INCLLIB          &SHAREDPREFIX..&SMICS.INCLLIB
  LOAD             &SHAREDPREFIX..&SMICS.LOAD
  LOCKIT           &SHAREDPREFIX..&SMICS.LOCKIT
  MACAUTOS         &SHAREDPREFIX..&SMICS.MACAUTOS
  MACLIB           &SHAREDPREFIX..&SMICS.MACLIB
  MCOLIB           &SHAREDPREFIX..&SMICS.MCOLIB
  MICSHLIB         &SHAREDPREFIX..&SMICS.ISPHLIB
  MICSMLIB         &SHAREDPREFIX..&SMICS.ISPMLIB
  MICSPLIB         &SHAREDPREFIX..&SMICS.ISPPLIB
  MICSSLIB         &SHAREDPREFIX..&SMICS.ISPSLIB
  MICSTLIB         &SHAREDPREFIX..&SMICS.ISPTLIB
  MWFPDS           &SHAREDPREFIX..&SMICS.MWFPDS.DATA
  MWFSAS           &SHAREDPREFIX..&SMICS.MWFSAS.DATA
  OBJ              &SHAREDPREFIX..&SMICS.OBJ
  PROTOLIB         &SHAREDPREFIX..&SMICS.PROTOLIB
  SASFLS           &SHAREDPREFIX..&SMICS.SASFLS
  SCREENS          &SHAREDPREFIX..&SMICS.SCREENS
  SOURCE           &SHAREDPREFIX..&SMICS.SOURCE
  TABLES           &SHAREDPREFIX..&SMICS.TABLES
  DOCDIR           &HFSROOT./documents
  VDIC             &SHAREDPREFIX..&SMICS.DIC.TEXT
  VDOC             &SHAREDPREFIX..&SMICS.DOC.TEXT
  VINC             &SHAREDPREFIX..&SMICS.INC.TEXT
 *
 *   UNIT DATASETS (DASD)
 *
  CHECKPT          &PREFIX..&PMICS.CHECKPT.DATA
  CICDATA          &PREFIX..&PMICS.CIC.DATA
  CNTL             &PREFIX..&PMICS.CNTL
  DETAIL           &PREFIX..&PMICS.DETAIL
  DETAIL2          &PREFIX..&PMICS.DETAIL2
  RESTART          &PREFIX..&PMICS.RESTART.CNTL
  USOURCE          &PREFIX..&PMICS.USER.SOURCE
 *
 *   UNIT DATASETS (TAPE)
 *
  ACCTAUDIT        &TAPEPREFIX..&TMICS.ACCT
  AUDIT            &TAPEPREFIX..&TMICS.AUDIT
  BACKUP           &TAPEPREFIX..&TMICS.BACKUP
  HISTW            &TAPEPREFIX..&TMICS.HISTW
  HISTM            &TAPEPREFIX..&TMICS.HISTM
  MBACKUP          &TAPEPREFIX..&TMICS.MBACKUP
 *
 *   CA MICS PROCEDURE NAMES
 *
  MICSASM          MICSASM
  MICSCOM          MICSCOM
  MICSDB           MICSDB&PROCIDENTIFIER
  MICSDM           MICSDM
  MICSDU           MICSDU&PROCIDENTIFIER
  MICSDOC          MICSDOC
  MICSGT           MICSGT
  MICSGU           MICSGU
  MICSNDB          MICSNDB&PROCIDENTIFIER
  MICSRPTS         MICSRPTS
  MICSSHR          MICSSHR&PROCIDENTIFIER
```

Figure 2-28. Sample CA MICS User Names Table

2.3.3.3.2.4 CA MICS Database Split Table (DBSPLIT)

The CA MICS Online Database is composed of five timespans: DETAIL, DAYS, WEEKS, MONTHS, and YEARS. CA MICS allows the files associated with each timespan to be spread across one to 37 OS files. This table is used to specify which CA MICS Information Areas (a logical grouping of component files) will be in which OS files.

The CA MICS Database Split Table consists of a number of statements in member DBSPLIT in prefix.MICS.PARMS. This table is free-form but positional. Comments are specified by starting the line with an asterisk. The first field is the component's Information Area name (e.g. ADM, TSO, SCP, etc.). The other fields are the DD names where the Information Area files are to be written. The possible DD names are:

DETAIL	DETAILs
DAYS	DAYSs
WEEKS	WEEKSs
MONTHS	MONTHSs
YEARS	YEARSs

where s is the split id 0-9 or A-Z.

NOTE: When a numeric database id is used, split id L cannot be used for DETAIL and S cannot be used for MONTHS.

As distributed, the Database Split Table is empty indicating that all Information Area files are to be written to DETAIL, DAYS, WEEKS, MONTHS, and YEARS DD names. Figure 2-29 shows the CA MICS Database Split Table for a database with the DETAIL timespan split into three additional OS files and the MONTHS timespan split into two additional OS files. In actual use, only the statements for Information Areas that use some DD name other than the standard DETAIL, DAYS, WEEKS, MONTHS, and YEARS would be required in the table, as shown in Figure 2-30.

```
+----------------------------------------------------------+
| INFORMATION AREA   DDNAME LIST                           |
+----------------------------------------------------------+
|                                                          |
|    ADM             DETAIL  DAYS WEEKS MONTHS  YEARS      |
|    SCP             DETAIL1 DAYS WEEKS MONTHS  YEARS      |
|    HAR             DETAIL  DAYS WEEKS MONTHS  YEARS      |
|    BAT             DETAIL2 DAYS WEEKS MONTHS  YEARS      |
|    TSO             DETAIL2 DAYS WEEKS MONTHS  YEARS      |
|    CIC             DETAIL  DAYS WEEKS MONTHS2 YEARS      |
|    IMS             DETAIL3 DAYS WEEKS MONTHS  YEARS      |
|    OPS             DETAIL  DAYS WEEKS MONTHS  YEARS      |
|    SRL             DETAIL3 DAYS WEEKS MONTHSB YEARS      |
|    ACT             DETAIL  DAYS WEEKS MONTHS  YEARS      |
|                                                          |
|                                                          |
+----------------------------------------------------------+
```

Figure 2-29. CA MICS Database Split Table Example 1

```
+----------------------------------------------------------+
| INFORMATION AREA   DDNAME LIST                           |
+----------------------------------------------------------+
|                                                          |
|    BAT              DETAIL2 DAYS WEEKS MONTHSA YEARS     |
|    TSO              DETAIL2 DAYS WEEKS MONTHS  YEARS     |
|    CIC              DETAIL  DAYS WEEKS MONTHS9 YEARS     |
|    IMS              DETAIL3 DAYS WEEKS MONTHS  YEARS     |
|                                                          |
+----------------------------------------------------------+
```

Figure 2-30. CA MICS Database Split Table Example 2

For the installation checklist for the DBSPLIT option, refer to Section 3.8.5.

2.3.3.3.3 Generation Control Language

A series of action statements, called the Generation Control Language (GCL), is provided to increase the flexibility and maintainability of the Prototype Library members. The language consists of a number of statements which may be coded within the members of the Prototype Library and which are acted upon by the JCL generation process to perform conditional generation. Refer to Section 2.3.3.3, "JCLGEN Parameters for Special Requirements", for a discussion of the logic of the CA MICS JCLGEN process.

The GCL language and the symbolic variables it supports are described in the following sections:

1 - Generation Control Language Variables
2 - Generation Control Language Syntax
3 - Using the Generation Control Language

2.3.3.3.3.1 Generation Control Language Variables

When JCLGEN is processing your definitions, it must have a method of placing the information you provide into the records being created. This process of substitution is done via symbolic variables similar to normal IBM CLIST and JCL processing. The prototype statements from MICS.PROTOLIB or MICS.PARMS may contain symbolic variables that JCLGEN replaces with the values you defined.

Symbolic variables begin with an ampersand (&) and are terminated by any of the common separator characters such as a space, parenthesis, another ampersand, equals sign, etc. If the delimiter is a period, it is taken out of further processing. Two ampersands in a row (&&) are replaced with a single ampersand. Substitution of symbolics are repeated on a line until no more symbolics are found.

Besides supporting symbolic variables substitution, the JCLGEN process has a number of built-in functions: &DATATYPE, &LENGTH, &STR, &SUBSTR, and &DLM.

Examples of symbolic substitutions and functions are shown below. For these examples, assume that the values of the symbolic variables are as follows:

```
&CICDATA      = &prefix..MICS.CIC.DATA
&DASDSMSPARMS = STORCLAS=MICSTEMP
&DASDUNIT     = SYSDA
&JOBLINES     = 50
&PREFIX       = XYZ.CICS
```

```
            &PROG         = 'MICS PROGRAMMER'
            &RIP          = CIC
            &SAVEMEMBER   = SMFINITA
            &TAPEUNIT     = 3400-5
```

Standard substitution works by replacing the name of the symbol with the symbol's value. This process repeats until there are no more symbol names present in the prototype statement.

```
  Example 1:
    Before: //WORK   DD  UNIT=&DASDUNIT,
    After:  //WORK   DD  UNIT=SYSDA,
```

The next example shows the use of the &STR function as well as multiple substitutions. &STR returns the value within the parenthesis as is.

```
  Example 2:
    Before: //CICDATA DD DISP=SHR,DSN=&&&RIP&STR(DATA)
    Scan 1: //CICDATA DD DISP=SHR,DSN=&CICDATA
    Scan 2: //CICDATA DD DISP=SHR,DSN=&PREFIX..MICS.CIC.DATA
    After:  //CICDATA DD DISP=SHR,DSN=XYZ.CICS.MICS.CIC.DATA
```

In this case, the &STR function was not needed. The statement could have been written as in the next example and would give the same results.

```
  Example 3:
    Before: //CICDATA DD DISP=SHR,DSN=&&&RIP.DATA
    Scan 1: //CICDATA DD DISP=SHR,DSN=&CICDATA
    Scan 2: //CICDATA DD DISP=SHR,DSN=&PREFIX..MICS.CIC.DATA
    After:  //CICDATA DD DISP=SHR,DSN=XYZ.CICS.MICS.CIC.DATA
```

&DATATYPE(expression) will return a value of NUM or CHAR depending whether the expression is numeric or not.

```
    Before: :IF &DATATYPE(&DASDUNIT) EQ CHAR THEN
    Scan 1: :IF &DATATYPE(SYSDA) EQ CHAR THEN
    After:  :IF CHAR EQ CHAR THEN

    Before: :IF &DATATYPE(&JOBLINES) NE NUM THEN GOTO ERROR
    Scan 1: :IF &DATATYPE(50) NE NUM THEN GOTO ERROR
    After:  :IF NUM NE NUM THEN GOTO ERROR
```

&LENGTH(expression) will return a numeric value equal to the number of characters in the expression up to the last non-blank character. The following example illustrates the use of &LENGTH:

```
    Before: // PGMLEN='&LENGTH(&PROG)',
    Scan 1: // PGMLEN='&LENGTH(MICS PROGRAMMER)',
    After:  // PGMLEN='15',
```

The &SUBSTR function is used to return a portion of a string as a value. It is patterned after the CLIST &SUBSTR function and not the SAS SUBSTR function. Its syntax is &SUBSTR(start position:end position,character string) where start position and end position are relative to the first character in the string being position 1. If start position and end position have the same values, a string of length 1 is returned. In this case, end position need not be specified. Some examples of the use of &SUBSTR follow:

```
  Example 1:
    Before: :SET NAME = &SUBSTR(6:15,&PROG)
    Scan 1: :SET NAME = &SUBSTR(6:15,MICS PROGRAMMER)
    After:  :SET NAME = PROGRAMMER

  Example 2:
    Before: &SUBSTR(1,&DATATYPE(&&&RIP.DATA))
    Scan 1: &SUBSTR(1,&DATATYPE(&CICDATA))
    Scan 2: &SUBSTR(1,&DATATYPE(&PREFIX..MICS.CIC.DATA))
    Scan 3: &SUBSTR(1,&DATATYPE(XYZ.CICS.MICS.CIC.DATA))
    Scan 4: &SUBSTR(1,CHAR)
    After:  C
```

&DLM(expression) inserts a comma before the expression value IF THE VALUE IS NOT NULL. If the expression resolves to a null value, the comma is omitted. &DLM is used in constructing JCL statements with symbols that may have null values.

```
  Example 1:
    Before: //WORK   DD  UNIT=&DASDUNIT&DLM(&DASDSMSPARMS),
    After:  //WORK   DD  UNIT=SYSDA,STORCLAS=MICSTEMP,
                                   |
                                   +--inserted by &DLM
```

This example illustrates the situation where &DASDSMSPARMS has a null value.

```
  Example 2:
    Before: //WORK   DD  UNIT=&DASDUNIT&DLM(&DASDSMSPARMS),
    After:  //WORK   DD  UNIT=SYSDA,
```

A complete list of the variables that are available at the start of the JCL generation process is found in the sharedprefix.MICS.SOURCE(JCLPARM). Some of the more commonly used variables are shown below in Figure 2-31.

SYMBOL	SYNONYMS	DESCRIPTION
&ACCT	&ACCOUNT &A	value of JCLDEF "ACCT"
&AGDG		value of JCLDEF "AUDITGDG"
&ARCHAUD		value of JCLDEF "ARCHIVE AUDIT"
&ARCHMNH		value of JCLDEF "ARCHIVE HISTM"
&ARCHWKH		value of JCLDEF "ARCHIVE HISTW"
&ARET		value of JCLDEF "AUDITGDG" * 7
&AUTOSUBMIT		value of JCLDEF "AUTOSUBMIT"
&BFREQ		value of JCLDEF "BACKUP FREQ"
&BGDG		value of JCLDEF "BCKUPGDG"
&BRET		value of JCLDEF "BCKUPGDG" * 1
&ccc		YES if COMPONENT in unit
&CVOLSER		value of JCLDEF "CVOLSER"
&CVOLUNIT		value of JCLDEF "CVOLUNIT"
&DASDSMSPARMS		value of JCLDEF "DASDSMSPARMS"
&DASDUNIT		value of JCLDEF "DASDUNIT"
&DAYSMFFILES		value of JCLDEF "DAYSMFFILES"
&DATABASE		value of JCLDEF "DATABASE"
&DENSITY		value of JCLDEF "DENSITY"
&DOCSYSOUT		value of JCLDEF "DOCUMENT"
&EXPDT		value of JCLDEF "EXPDT"
&GROUP	&G	value of JCLDEF "GROUP"
&INCLUDELEVEL		internal include nesting level
&ISPxLIBn		value of JCLDEF "ISPxLIBn"
&JES		value of JCLDEF "JES"
&JOBCLASS	&CLASS &C	value of JCLDEF "CLASS" value of JCLINFO "JOBCLASS"
&JOBCOPIES	&COPIES	value of JCLINFO "JOBCOPIES"
&JOBLINES	&LINES	value of JCLINFO "JOBLINES"
&JOBNAME		internal value created from value of JCLDEF "JOBNAME" and value of JCLINFO "UJOBNAME"
&JOBPRTY	&PRTY &PRT	value of JCLDEF "PRTY" value of JCLINFO "JOBPRTY"
&JOBTIME	&TIME &T	value of JCLINFO "JOBTIME"
&JOB7380		value of JCLDEF "JOB7380"
&LBL		internal file sequence number

```
|&LOGS                     | value of JCLDEF "LOGS"         |
|&MGDG                     | value of JCLDEF "HISTMGDG"     |
|&MICSLVL                  | internal Maintenance level     |
+--------------------------+--------------------------------+

+--------------------------+--------------------------------+
|SYMBOL       SYNONYMS     | DESCRIPTION                    |
+--------------------------+--------------------------------+
|&MRET                     | value of JCLDEF "HISTMGDG" * 31|
|&MSGCLASS    &MSC         | value of JCLDEF "MSGCLASS"     |
|&MSGLEVEL    &MSGLEV &MSL | value of JCLDEF "MSGLEVEL"     |
|&NOTIFY      &N           | value of JCLDEF "NOTIFY"       |
|&PASSWORD    &PA          | value of JCLDEF "PASSWORD"     |
|&PERFORM     &PE          | value of JCLDEF "PERFORM"      |
|&PMICS                    | internal value of MICS. or null|
|&PROCLIB                  | value of JCLDEF "PROCLIB"      |
|&PROG        &PRO         | value of JCLDEF "PROG"         |
|&REGION      &RE          | value of JCLDEF "REGION"       |
|&RENUMBER                 | value of JCLDEF "RENUMBER"     |
|&REPORTS                  | value of JCLDEF "REPORTS"      |
|&RIP                      | internal JCLGEN variable that  |
|                          | contains Component for         |
|                          | REPEATINCLUDE statement        |
|&SASHELP                  | value of JCLDEF "SASHELP"      |
|&SASLIB                   | value of JCLDEF "SASLIB"       |
|&SHAREDPREFIX             | value of JCLDEF "SHAREDPREFIX" |
|&SMICS                    | internal value of MICS. or null|
|&SORTCYL                  | value of JCLDEF "SORTCYL"      |
|&SORTLIB                  | value of JCLDEF "SORTLIB"      |
|&SORTSMSPARMS             | value of JCLDEF "SORTSMSPARMS" |
|&SORTUNIT                 | value of JCLDEF "SORTUNIT"     |
|&SYSDATE                  | internal date, DDMMMYY format  |
|&SYSTIME                  | internal time, HH:MM format    |
|&SYSUID                   | same as &GENLIST               |
|&TAPEPREFIX               | value of JCLDEF "TAPEPREFIX"   |
|&TAPEUNIT                 | value of JCLDEF "TAPEUNIT"     |
|&TMICS                    | internal value of MICS. or null|
|&TSMLINK                  | value of JCLDEF "TSMLINK"      |
|&TSMPARMS                 | value of JCLDEF "TSMPARMS"     |
|&UJOBNAME                 | value of JCLINFO "UJOBNAME"    |
|&USER        &U           | value of JCLDEF "USER"         |
|&WGDG                     | value of JCLDEF "HISTWGDG"     |
|&WRET                     | value of JCLDEF "HISTWGDG" * 7 |
+-----------------------------------------------------------+
| NOTE:  ANY OF THE SYMBOLIC DATA SET NAMES MAY ALSO BE     |
|        USED, BUT THEY ARE NOT INCLUDED IN THIS TABLE.     |
|        SEE SECTION 2.3.3.3.2.3, "CA MICS DATA SET NAMES   |
|        TABLE (JCLNAMES)".                                 |
+-----------------------------------------------------------+
```

Figure 2-31. Generation Control Language Variables (Partial)

2.3.3.3.3.2 Generation Control Language Syntax

The Generation Control Language (GCL) has 14 types of statements, each of which must begin with a colon:

FUNCTION	STATEMENT	GENERAL FORMS
Comments	:*	:* any comments
Branching	:GOTO	:GOTO c
Labels	:label:	:c:
Conditional testing	:IF	:IF a = b THEN :IF a = b THEN GOTO c :IF a NE b THEN :IF a NE b THEN GOTO c :IF EXIST a THEN ... :IF NOT_EXIST a THEN
Else conditions	:ELSE	:ELSE :ELSE GOTO c
Setting values	:SET	:SET a = b :SET LBL + 1
	:SETLIST	:SETLIST l = a b c
Prevent JOB card	:NOJOBCARD	:NOJOBCARD
Create JOB card	:JOBCARD	:JOBCARD name
Include PROTO member	:INCLUDE	:INCLUDE member
Dynamically include a job	:GENERATE	:GENERATE job location
Include PARM member	:INCLUDEPARM	:INCLUDEPARM member
Exit PROTO member	:EXIT	:EXIT
Include list of PROTO members	:INCLUDELIST	:INCLUDELIST stp lo hi
Include PROTO member repeatedly	:REPEATINCLUDE	:REPEATLINCLUDE m l

Figure 2-32. GCL Statements

GCL was written for a very specific purpose--JCLGEN. To keep the implementation as simple as possible, the language has very rigid, simple syntax rules. The TSO CLIST language was used as a design guide for GCL; however, GCL does not work just like CLISTs.

GCL has the following general rules:

1. Only one statement can appear on a line. Continuation is not allowed.

2. All GCL statements begin with a colon. The colon may start in column 1 or later; however, nothing may precede it.

3. There cannot be any blanks between the colon and the GCL command name.

COMMENTS (:*   )

The purpose of this statement is to provide documentation in the code. A comment can appear anywhere within the prototype member.

valid example

```
:*  THIS IS A COMMENT
```

invalid example

```
DATA HERE :* ONLY ONE STATEMENT PER LINE
```

The above example is invalid because characters precede the GCL colon.

BRANCHING (:GOTO)

The purpose of this statement is to skip over one or more prototype member statements. You may only branch forward (down towards the last statement in the prototype member). Branching is done by reading and ignoring statements until a label matching the branch label is found. If the end of the member is reached before the label is found, an error message is printed and processing of this member is terminated.

valid example

```
:GOTO B
:A:
//* THIS JCL WILL NOT BE IN THE OUTPUT
```

```
    . . .
    :B:
    //* THIS JCL WILL BE IN THE OUTPUT
```

invalid example

```
    :GOTO :ERROR:
```

The above example is invalid because the branch label must not contain a beginning and/or ending colon.

LABELS (:label:)

The purpose of this statement is to identify a termination point for a branch. All GCL labels begin and end with a colon. Since all branching is downward to a label statement not yet read, an old label name may be "reused." Nothing may follow the ending colon.

valid example

```
    :IF &A = YES THEN GOTO CONTINUE
    //*  SYMBOL A IS NOT EQUAL TO YES
    ...
    :CONTINUE:
    ...
    :IF &B NE 5 THEN GOTO CONTINUE
    //*  SYMBOL B IS EQUAL TO 5
    ...
    :CONTINUE:
```

invalid example

```
    :CONTINUE
```

The above example is invalid because the label does not have an ending colon. JCLGEN will flag this as an unknown GCL statement.

CONDITIONAL TESTING (:IF)

The purpose of this statement is to provide conditional statement execution. The statement in its basic form is written:

```
    :IF a operator b THEN
```

The "a" and "b" represent symbols or constant values. If "a" or "b" contain special characters or blanks, they must be specified with the GCL &STR function (i.e., &STR(3400-4))

The valid comparison operators are:

```
    = ^=  EQ NE
```

The word "THEN" is always required. If the comparison specified is true, the next single statement will be executed. If the comparison is false, the next single statement will be ignored.

valid example

```
    :IF &DATABASE = NORTH THEN
    :INCLUDE DYNORTH
   or
    :IF &TAPEUNIT = &STR(3400-5) THEN
    :SET USRTAPE = &TAPEUNIT
```

invalid example

```
    :IF &SMF = YES AND &TSO = YES THEN
```

The above example is invalid because it contains a compound expression. However, it could have been written as follows:

```
    :IF &SMF&TSO = YESYES THEN
```

In the basic form of the :IF statement, nothing may follow the THEN. A second form of the :IF statement is provided to allow conditional branching. Its special form is:

```
    :IF a operator b THEN GOTO c
```

The special form is an abbreviation for the following basic form:

```
    :IF a operator b THEN
    :GOTO c
```

valid example

```
    :IF &VCA NE YES THEN GOTO SKIPVCA
```

invalid example

```
    :IF &DATABASE = NORTH GOTO NORTHFOUND
```

The above example is invalid because the THEN word is always required.

The special comparison operators EXIST and NOT_EXIST are used to test whether or not a JCLGEN variable has been defined to the JCLGEN process. This can be used to prevent unresolved variable errors that may occur in JCLGEN processing.

The following example will test if the variable a has been defined:

```
    :IF EXIST a THEN
```

The following example will test if the variable a has not been defined.

```
    :IF NOT_EXIST a THEN
```

Note that the actual variable name is used. If an ampersand is used, then variable resolution takes place and the resolved value is tested for existance.

For example:

```
:SET A = SMFID
:IF EXIST &A THEN
```

will test for the existance of variable SMFID.

ELSE CONDITIONS (:ELSE)

The purpose of this statement is to provide conditional execution of a single statement if the previous :IF statement was false.

```
valid example
    :IF &DATABASE = NORTH THEN
    //@ THIS STATEMENT IN NORTH JCL
    :ELSE
    //@ THIS STATEMENT NOT IN NORTH JCL
```

An :ELSE may not follow a conditional :INCLUDE or :INCLUDEPARM. This restriction is necessary since the included member may contain GCL statements.

```
invalid example
    :IF &DATABASE = NORTH THEN
    :INCLUDE DYNORTH
    :ELSE
    :INCLUDE DYSOUTH
```

The above example is invalid because the :ELSE follows a conditional :INCLUDE. The valid way to get the desired result is as follows:

```
:IF &DATABASE = NORTH THEN
:INCLUDE DYNORTH
:IF &DATABASE NE NORTH THEN
:INCLUDE DYSOUTH
```

In the basic form of the :ELSE statement, nothing may follow the :ELSE command name. A second form of the :ELSE statement is provided to allow conditional branching. Its special form is as follows:

```
:ELSE GOTO c
```

The special form is an abbreviation for the following basic form:

```
:ELSE
:GOTO c
```

SETTING VALUES (:SET)

The purpose of this statement is to save the specified value in the named symbolic variable. The symbolic variable name can be up to 31 characters long. You may define new symbolic variables in sharedprefix.MICS.PARMS(JCLDEFC) or in prefix.MICS.PARMS(JCLDEF) via the USERDEF keyword. To define new symbolic variables, start with the three characters "USR" to prevent conflict with existing CA MICS symbols.

The value to which the symbol is set can be another predefined symbolic variables and/or a constant. If the value contains special characters, it should be specified via the &STR function. If the value contains blanks, it must be surrounded by apostrophes within the &STR function.

valid example

```
:SET USRDEVICE = &STR(3400-4)
```

invalid example

```
:SET USRRLSE = 6
:SET USRHEADING = MICS RELEASE &USRRLSE
//@ &USRHEADING
```

The above example is invalid because the second :SET value contains embedded blanks. The value should have been written:

```
:SET USRHEADING = &STR('MICS RELEASE &USRRLSE')
```

A special form of :SET is used to increment the file label sequence number specified in tape DD statements. The special form must be written as shown below. Note the lack of an equal sign.

```
:SET LBL + 1
```

Any attempt to use the + operator with a symbol other than LBL will result in an error.

SETTING VALUES (:SETLIST)

The purpose of this statement is to save the values of 1 or more symbolic variables in the named symbolic variable. Constants can also be introduced, including special characters.

The symbolic variable name can be up to 31 characters long. You may define new symbolic variables in sharedprefix.MICS.PARMS(JCLDEFC) or in prefix.MICS.PARMS(JCLDEF) via the USERDEF keyword. To define new symbolic variables, start with the three characters "USR" to prevent conflict with existing CA MICS symbols.

valid examples

```
:SETLIST USRCOMP = &USR1 &USR2 &USR3
:SETLIST USRCOMP = 4096  &USR2 X Y #Z
```

invalid example

```
:SETLIST USRCOMP &USR1 &USR2 &USR3
```

The above example is invalid because an equal sign was not used to separate the symbol name from the value list.

PREVENT JOB CARD (:NOJOBCARD)

The purpose of this statement is to prevent the automatic creation of a JOB card for a JCL member. This statement must be the first noncomment statement read from the MICS.PROTOLIB member during the creation of the JCL member.

valid example

```
:NOJOBCARD
:IF &AUTOSUBMIT = YES THEN
:JOBCARD BACKUPJ
```

invalid example

```
:IF &AUTOSUBIT = NO THEN
:NOJOBCARD
```

The above example is invalid because the :NOJOBCARD comes too late--the JOB card has already been created.

CREATE JOB CARD (:JOBCARD)

The purpose of this statement is to create a JOB card where specified in the generated member. This statement is intended for jobs that submit other jobs to the JES internal reader. A job name may be specified following the :JOBCARD command name. If the job name is missing, the current member's job name will be used. The name is used to search the JCLINFO and JCL$Iccc tables. If the job name is found, the values specified for TIME, LINES, COPIES, CLASS, and PRIORITY will be used. If the job name is not found, the DEFAULT entry information will be used.

valid example

```
:JOBCARD MONTHLYB
```

invalid example

```
:JOBCARD MONTHLYB1
```

The above example is invalid because the name exceeds 8 characters.

INCLUDE PROTOLIB MEMBER (:INCLUDE)

The purpose of this statement is to process members from the MICS.PROTOLIB library into the member being generated. After reaching the end of the included member, generation continues with the statement following the :INCLUDE statement. This next statement may not be an :ELSE. Up to six nested :INCLUDEs can be defined. Any number of non-nested :INCLUDEs can be in a member. Note that for the purpose of the nesting limit, :INCLUDE and :INCLUDEPARM (described below) must be counted together.

valid example

```
:INCLUDE ABC
```

invalid example

```
     :INCLUDE A

     assume that:
       member A contains - :INCLUDE B
       member B contains - :INCLUDE C
       member C contains - :INCLUDE D
       member D contains - :INCLUDE E
       member E contains - :INCLUDE F
       member F contains - :INCLUDE G
       member G contains - :*
```

The above example is invalid because member F is at the sixth level and may not include a seventh (:INCLUDE G).

INCLUDE PARMLIB MEMBER (:INCLUDEPARM)

The purpose of this statement is to process members from the MICS.PARMS library into the member being generated. After reaching the end of the included member, generation continues with the statement following the :INCLUDEPARM statement. This next statement may not be an :ELSE. Up to six nested :INCLUDEPARMs can be defined. Any number of non-nested :INCLUDEPARMSs can be in a member. Note that for the purpose of the nesting limit, :INCLUDE (described above) and :INCLUDEPARM must be counted together.

valid example

```
    :INCLUDEPARM ABC
```

invalid example

```
    :INCLUDEPARM ABC DEF
```

The above example is invalid because you may not specify more than one member name.

DYNAMICALLY GENERATE JOBS (:GENERATE)

The GENERATE statement is used to request the generation of additional jobs.

Unlike :INCLUDE, :GENERATE actually creates a new member in the target library.

you can also use :GENERATE to define dependencies between jobs. For example, the statement ":GENERATE B CNTL" in the PROTOLIB member for job A requests that whenever JOB

A is generated also generate job B.

The library specification can be the following:

```
CNTL
CLIST
MACRO
PARMS
USOURCE
```

valid example

```
:GENERATE MYPARM PARMS
```

invalid example

```
:GENERATE MYPARM
:GENERATE MYPARM  MACLIB
```

- o The first example is invalid because no destination library has been specified.

- o The second example is invalid because MACLIB is an invalid destination library.

EXIT MEMBER (:EXIT)

The purpose of this statement is to return control to the member which included this member. If this member was specified in the GENLIST member, the generation for this member is finished and the next member specified in the GENLIST is begun.

valid example

```
:IF &AUTOSUBMIT EQ YES THEN
:EXIT
```

invalid example

```
:IF &AUTOSUBMIT EQ YES THEN
:EXIT:
```

The above example is invalid because the ending colon makes this GCL statement into a valid label and processing would continue.

INCLUDE LIST OF PROTOLIB MEMBERS (:INCLUDELIST)

This is a special purpose statement used to include the DAYS, WEEKS, MONTHS, and YEARS job step members from MICS.PROTOLIB when generating the DAILY, WEEKLY, MONTHLY, and YEARLY jobs. The parameters specify the type of job steps to be included and a low and high step suffix number value. Only the step members that fall within the range of these values will be selected. Valid step names types are DYSTEPS, WKSTEPS, MNSTEPS, and YRSTEPS. PROTOLIB members that may be processed are DYcccnnn, WKcccnnn, MNcccnnn, and YRcccnnn where ccc is the identifier of a component defined in the database unit and nnn is the component step number that must fall between the low and high values specified in the :INCLUDELIST statement.

valid example

```
    :INCLUDELIST DYSTEPS ___010 ___199
```

invalid example

```
    :INCLUDELIST &MSGCLASS
```

The above example is invalid because it does not specify a valid job step type or provide a low or high range value.

REPEATEDLY INCLUDE A PROTOLIB MEMBER (:REPEATINCLUDE)

The purpose of this statement is to include the specified MICS.PROTOLIB member multiple times. For each inclusion, the &RIP internal symbol will contain a different value from the specified list. The :REPEATINCLUDE statement cannot be nested.

valid example

```
    :SET KW = &STR(A BB CCC DDDD)
    :REPEATINCLUDE MYMEMBER KW
```

PROTOLIB member MYMEMBER would be included four times. The variable &RIP will contain a value of A for the first include, BB for the second, CCC for the third, and DDDD for the last include.

invalid example

```
    :REPEATINCLUDE SMFCOMPTS
```

The above example is invalid because no repeat value name is specified.

2.3.3.3.3.3 Using the Generation Control Language

We cannot anticipate all of the uses CA MICS users may make of the Generation Control Language. Fortunately, examples of its use abound, because CA MICS uses GCL facilities extensively to implement the JCL tailoring which occurs during the CA MICS installation/generation process.

If you think you have a need to call on the facilities of GCL, look through sharedprefix.MICS.PROTOLIB. It is very likely that you will see an example of GCL usage similar to the one you are considering. As with any CA MICS facility, the CA MICS Product Support Group will be happy to answer any questions you may have.

2.3.3.3.4 OS Data Set Names

All OS data set names in CA MICS are constructed of two parts, an installation defined prefix and a standard descriptive suffix. As part of the installation of CA MICS, the prefixes to be used at the complex and unit database levels must be defined in the PARMS members JCLDEFC and JCLDEF respectively. See Sections 2.3.1.6 and 2.3.3.2.1 for the explanation and examples for defining prefixes for disk and tape data sets.

It is strongly recommended that the descriptive suffix in the names not be changed. The CA MICS documentation contains thousands of references to the standard CA MICS names. The effort required to change all references and to continue to update new documentation is considerable. Changing the names without changing the documentation is sure to result in confusion for CA MICS users. Even if all documentation is changed at the installation, CA MICS education classes, Technical Support, and other CA MICS installations will continue to use the standard names. This will cause a constant communications problem.

Should the standard CA MICS names conflict with installation standards, an attempt should be made to get a waiver for the CA MICS data set names. If there is no alternative to changing the standard names, PARMS member JCLNAMES can be used to assign new names to CA MICS data sets. Section 2.3.3.3.2.3 describes how to specify new data set names in JCLNAMES. PARMS member ALLNAMES is a sample of JCLNAMES with all standard names defined.

For disk data sets, changing JCLNAMES is all that is required to have CA MICS reference the new installation names. Tape data sets are generation data groups. The names specified for tapes in JCLNAMES describes the names for a class of tape files (i.e. ACCT, AUDIT, BACKUP, HISTW, HISTM, or MBACKUP). The class name is combined with a file identifier and generation data group descriptor to create the tape data set name.

BACKUP and MBACKUP tapes have the timespan as the file descriptor, while other tapes use a file name as the descriptor. To change the data set names for tape files, the appropriate PROTOLIB members must be modified. The PROTOLIB members that contain tape data set names are ACTINITA, BACKUP, DBTPDDS, MNACT199, and MONTHLY. As part of unit generation, source member GENPROAR generates the prefix.MICS.PARMS members WKccc300 and MNccc300. You must change GENPROAR and not the generated PARMS members.

2.3.3.3.5 OS Job Names

As distributed, most generated CA MICS jobs will have the same name as the MICS.CNTL member in which they are saved (e.g. the database job will have the name ALLOCATE). These job names will probably not be acceptable at most installations. Three alternatives are available for creating job names for CA MICS jobs.

The easiest way of changing the CA MICS job names is to use the extended jobcard parameter JOBNAME. This parameter allows the specification of a one to eight character prefix to be added to the standard job names. The use of this parameter is described in Section 2.3.3.2.1.6.

If the use of the JOBNAME parameter is not acceptable, two alternatives are available: the job name for each job to be generated can be specified individually in the JCLINFO parameters member, or the USERJCL parameters member can be used to create job cards with job names generated with your GCL statements.

Section 2.3.3.3.2.2 describes the JCLINFO parameters member. One of the optional fields which can be used to provided job attributes is UJOBNAME. This is a user job name which will override the standard CA MICS job name.

The final alternative for creating job names is to create the job card with the USERJCL parameters member. Section 2.3.3.2.2 describes how to define the USERJCL parameters member.

2.3.3.3.6 OS Step Names

Jobs generated with JCLGEN have descriptive job step names. In the CA MICS installation jobs, the job step names can be changed without impact on CA MICS operation. This is NOT true of the CA MICS operational jobs (i.e., DAILY, INCRccc, WEEKLY, MONTHLY, YEARLY, BACKUP, RESTORE, RSTATUS, and SCHEDULE). During the operational jobs, the current job step name is compared with the step which is acceptable to run, based on the status of the database checkpoint file.

CA MICS operational job step names have a strict naming convention. Having operational jobs that adhere to this convention allows the CA MICS Product Support representatives to accurately and quickly isolate operational problems. This helps us provide you with more timely problem resolution should operational job problems occur. Changing the job step names of CA MICS operational jobs always interrupts this initial problem isolation, and usually lengthens problem resolution times.

You may not be able to take full advantage of features provided by CA MICS if you modify the CA MICS operational job step names. CA MICS automatic job submission facilities in the batch SCHEDULE job and in the online Operational Status and Tracking facility create processes that include a series of CA MICS operational jobs. For example, a weekly process would consist of a DAILY, WEEKLY, and BACKUP in a single job stream. Job step names must be unique among all of the operational jobs in order to use the automatic submission facilities.

Changing operational job step names is a process that involves more user modifications than any other tailoring operation in CA MICS. A facility is provided to allow the override of CA MICS operational job step names used in checkpoint analysis. The step names must be manually modified in all of the affected PROTOLIB members, and overridden names must be mapped to standard names. This mapping may change as new products are added to the database.

Because of the maintenance burden this will add, and because modifying operational job step names usually lengthens support interactions, WE HIGHLY RECOMMEND THAT BEFORE THE EFFORT OF CHANGING JOB STEP NAMES IS BEGUN, YOU ATTEMPT TO GET A WAIVER OF THE INSTALLATION JOB STEP NAME STANDARD.

If you find that you must change operational job step names, follow these steps (which are explained in more detail in the text that follows):

MODIFYING PROTOLIB MEMBERS:

- Determine if you will be scheduling CA MICS through a vendor scheduling package or using a CA MICS provided facility.

- Identify the minimum number of job step names that must be changed.

- Identify the location of each job step name in members of sharedprefix.MICS.PROTOLIB.

- Prepare the PROTOLIB modifications.

TRANSLATING STEP NAMES:

- Prepare the PROC FORMAT statements that comprise the $USTEP format. This format is called by operational job checkpoint processing to translate the actual (modified) job and step name of an operational job into the CA MICS standard job and step name.

- Prepare the _USRSTEP macro that performs the opposite translation; given a CA MICS standard operational job step name, return the actual (modified) job step name in which restart processing should occur.

INSTALLING:

- Install the modifications, format, and macro.

- Keep complete written records of all modifications, and have them available for each call to the CA MICS Product Support group.

Here is a detailed description of the operational job step name modification strategy.

MODIFYING PROTOLIB MEMBERS:

1. Determine if you will be scheduling CA MICS through a vendor scheduling package or using a CA MICS provided facility.

   If you plan to use one of the CA MICS scheduling facilities, all operational job step names MUST be unique. If you use the same job step name, for example STEP1, in multiple CA MICS jobs, then you can NOT use the CA MICS batch SCHEDULE job or the Operational Status and Tracking SCHEDULE, DAILY, WEEKLY, MONTHLY, or YEARLY commands.

2. Identify the minimum number of job step names that must be changed.

   The CA MICS operational jobs and their step names are listed in Chapter 4. Thoroughly examine that chapter for an understanding of the structure and job step naming conventions used by CA MICS.

   From the list of possible job step names, select the fewest number of names that must be changed. Fewer name changes mean fewer modifications and less effort.

3. Identify the location of each job step name in members of sharedprefix.MICS.PROTOLIB.

   The PROTOLIB members that cause each operational job to be generated are named for the job: DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, RESTORE, RSTATUS, and SCHEDULE. Note: The INCRccc operational jobs are generated from the cccINCR protolib members.

   The operational update jobs (DAILY, INCRccc, WEEKLY, MONTHLY, and YEARLY) include other PROTOLIB members to complete their generation. The names of these members vary according to the job step which the members prototype.

   For the DAILY, WEEKLY, MONTHLY, and YEARLY job steps that update CA MICS databases, the members have names of the form ttcccsss, where

   ```
      tt = DY for the DAILY job
           WK for the WEEKLY job
           MN for the MONTHLY job
           YR for the YEARLY job
     ccc = the product identifier, and
     sss = the product update step number
   ```

   The INCRccc jobs are in the cccINCR members and have step names of INCRnnn.

   The product identifiers (ccc) and update step numbers (nnn) for each CA MICS data integration product can be found in sharedprefix.MICS.GENLIB(COMPTDEF).

An example of such a member is the CA MICS Batch and Operations (SMF) Analyzer. Its operational update step prototypes can be found in PROTOLIB members DYSMF030, SMFINCR, WKSMF030, MNSMF030, and YRSMF030.

In addition, the DAILY, WEEKLY, and MONTHLY jobs have separate report step prototype members in PROTOLIB. These members are called:

```
    DYRPT400
    WKRPT400
    MNRPT400
```

Finally, the DAILY, WEEKLY, MONTHLY, and YEARLY jobs have separate user step prototype members in PROTOLIB. These members are called:

```
    DYUSR500
    WKUSR500
    MNUSR500
    YRUSR500
```

4. Prepare the PROTOLIB modifications.

   If you need to change step names, to conform to a standard, you would typically build modifications for all the members identified above.

   As an example, we will examine a single modification. The first step of the DAILY job is called DAYALL. Its prototype is in sharedprefix.MICS.PROTOLIB(DAILY). For this example, lets say that the DAYALL step is located on a line with sequence number 00041000. The contents of that line are:

```
col 1                                              col 80
|                                                       |
//DAYALL EXEC &MICSNDB,SYSPARM=&SYSPARM          00041000
```

   To change the name of the step from DAYALL to STPDY000, build an IEBUPDTE input stream that replaces the line with the desired contents:

```
col 1                                              col 80
|                                                       |
./ CHANGE NAME=DAILY
```

```
    //STPDY000 EXEC &MICSNDB,SYSPARM=&SYSPARM        00041000
```

Repeat this procedure for each step and member modified. Be sure to add user maintenance block comments to each member, and save the update in the modification staging library sharedprefix.MICS.LOCALMOD.CNTL.

TRANSLATING STEP NAMES:

5. Prepare the $USTEP FORMAT.

   The $USTEP format is generated by every JCLGEN run in a unit database. The source for this format is contained in prefix.MICS.USER.SOURCE(USTEP) for every unit database.

   For every unit database, enter the step translation information in prefix.MICS.USER.SOURCE(USTEP).

   The distributed USTEP member contains:

```
    PROC FORMAT LIBRARY=TMUOLIB.MICSFMTS PRINT;
         VALUE $USTEP (MIN=16 MAX=16 DEFAULT=16)
    /*                                                   */
    /* INSERT STEP MAPPING DEFINITIONS HERE.             */
    /* FORM OF DEFINITIONS IS:                           */
    /* 'NEW_JOB_NAME_STEP_NAME'='OLD_JOB_NAME_STEP_NAME' */
    /*                                                   */
    /* EXAMPLE:                                          */
    /*  IF JOB "DAILY" STEP "DAY020" HAD BEEN RENAMED TO */
    /*  IF JOB "YLIAD" STEP "Q64A31" THE DEFINITION IS:  */
    /*  'YLIAD   Q64A31  '='DAILY   DAY020  '            */
    /*                                                   */
    /* IT IS VERY IMPORTANT THAT THE JOB AND STEP NAME   */
    /* BE SPECIFIED EXACTLY AS SHOWN. THE JOB MUST BE    */
    /* THE FIRST EIGHT POSITIONS AND THE STEP MUST START */
    /* IN POSITION NINE.                                 */
    /*                                                   */
    /*  NEW_JOB NEW_STEP   OLD_JOB OLD_STEP              */
    /* '1234567812345678'='1234567812345678'             */
    /*                                                   */
    /*                                                   */
    /*  THE FOLLOWING ENTRY REPRESENTS AN ILLEGAL        */
    /*  COMBINATION OF JOB AND STEP NAME.  IT IS         */
    /*  INCLUDED TO ENSURE THAT THIS FORMAT, AS          */
    /*  GENERATED, WILL NOT BE EMPTY.  ADD ANY SITE      */
    /*  DEPENDENT ENTRIES AFTER THIS DUMMY ENTRY.        */
```

```
/*                                                  */
'+               ' = '+              '
; RUN;
```

The translations must be added to this member. Since the CA MICS automatic job submission facilities generate a composite job stream of multiple CA MICS operational jobs, you must provide a mapping of all possible combinations of job and step names. The resulting SAS code in prefix.MICS.USER.SOURCE(USTEP), with comments removed, will look like this:

```
PROC FORMAT LIBRARY=TMUOLIB.MICSFMTS PRINT;
     VALUE $USTEP (MIN=16 MAX=16 DEFAULT=16)
'SYSOPDY STPDY000' = 'DAILY   DAYALL  '
'SYSOPDY STPDY001' = 'DAILY   DAYSMF  '
'SYSOPDY STPDY002' = 'DAILY   DAY010  '
'SYSOPIU STPIU010' = 'INCRTSO INCR010 '
'SYSOPDY STPDY003' = 'DAILY   DAY020  '
'SYSOPIU STPIU020' = 'INCRRMF INCR020 '
'SYSOPDY STPWK001' = 'WEEKLY  WEEK010 '
'SYSOPDY STPWK002' = 'WEEKLY  WEEK020 '
'SYSOPDY STPMN001' = 'MONTHLY MONTH010'
'SYSOPDY STPMN002' = 'MONTHLY MONTH020'
'SYSOPDY STPYR001' = 'YEARLY  YEAR010 '
'SYSOPDT STPYR002' = 'YEARLY  YEAR020 '
      . . .        =      . . .
'SYSOPWK STPWK001' = 'WEEKLY  WEEK010 '
'SYSOPWK STPWK002' = 'WEEKLY  WEEK020 '
      . . .        =      . . .
'SYSOPMN STPMN001' = 'MONTHLY MONTH010'
'SYSOPMN STPMN002' = 'MONTHLY MONTH020'
      . . .        =      . . .
'SYSOPYR STPYR001' = 'YEARLY  YEAR010 '
'SYSOPYR STPYR002' = 'YEARLY  YEAR020 '

              ... etc ...

'+               ' = '+              '
; RUN;
```

Note that each DAILY job step name is mapped with the DAILY, WEEKLY, MONTHLY, and YEARLY job name. WEEKLY, MONTHLY, and YEARLY job step names must also be mapped with the DAILY job name.

If you have duplicate job step names in the CA MICS operational jobs, you will not be able to code the multiple job name mapping. For example, if the first step in the DAILY job is MICS001 and this is also the name of the first step in the WEEKLY job, you cannot correctly map the MICS001 step to DAYALL when the DAILY operational job is part of a composite job stream of DAILY, WEEKLY, and BACKUP (the WEEKLY process). For this reason, you will not be able to use the CA MICS SCHEDULE job or the Operational Status and Tracking SCHEDULE, DAILY, WEEKLY, MONTHLY, or YEARLY commands.

6. Prepare the _USRSTEP macro.

   Remember that the function of the $USTEP macro was to translate an actual job/step name into the equivalent CA MICS standard job and step name. The _USRSTEP macro has the opposite function: to find the name of the step in a failed operational job, which should be specified in a RESTART JCL control parameter. The step name found must be put into a data element called RESTEP.

   The macro could use any of several methods to reassign the new step name into the data element RESTEP. One method that is relatively reliable for a minimum effort is to use the $USTEP format to contain both directions of translation. For example, add to the USTEP member in the above example the opposite translation:

```
PROC FORMAT LIBRARY=TMUOLIB.MICSFMTS PRINT;
     VALUE $USTEP (MIN=16 MAX=16 DEFAULT=16)
'SYSOPDY STPDY000' = 'DAILY   DAYALL  '
'SYSOPDY STPDY001' = 'DAILY   DAYSMF  '
'SYSOPDY STPDY002' = 'DAILY   DAY010  '
'SYSOPIU STPIU010' = 'INCRTSO INCR010 '
'SYSOPDY STPDY003' = 'DAILY   DAY020  '
'SYSOPIU STPIU020' = 'INCRRMF INCR020 '
'SYSOPDY STPWK001' = 'WEEKLY  WEEK010 '
'SYSOPDY STPWK002' = 'WEEKLY  WEEK020 '
'SYSOPDY STPMN001' = 'MONTHLY MONTH010'
'SYSOPDY STPMN002' = 'MONTHLY MONTH020'
'SYSOPDY STPYR001' = 'YEARLY  YEAR010 '
'SYSOPDT STPYR002' = 'YEARLY  YEAR020 '
      . . .        =       . . .
'SYSOPWK STPWK001' = 'WEEKLY  WEEK010 '
'SYSOPWK STPWK002' = 'WEEKLY  WEEK020 '
      . . .        =       . . .
'SYSOPMN STPMN001' = 'MONTHLY MONTH010'
'SYSOPMN STPMN002' = 'MONTHLY MONTH020'
      . . .        =       . . .
'SYSOPYR STPYR001' = 'YEARLY  YEAR010 '
'SYSOPYR STPYR002' = 'YEARLY  YEAR020 '
```

```
              ... etc ...

'DAILY   DAYALL  ' = 'SYSOPDY STPDY000'
'DAILY   DAYSMF  ' = 'SYSOPDY STPDY001'
'DAILY   DAY010  ' = 'SYSOPDY STPDY002'
'DAILY   DAY020  ' = 'SYSOPDY STPDY003'
      . . .        =       . . .
'INCRTSO INCR010 ' = 'SYSOPIU STPIU010'
'INCRRMF INCR020 ' = 'SYSOPIU STPIU020'
      . . .        =       . . .
'WEEKLY  WEEK010 ' = 'SYSOPWK STPWK001'
'WEEKLY  WEEK020 ' = 'SYSOPWK STPWK002'
      . . .        =       . . .
'MONTHLY MONTHL10' = 'SYSOPMN STPMN001'
'MONTHLY MONTH020' = 'SYSOPMN STPMN002'
      . . .        =       . . .
'YEARLY  YEAR010 ' = 'SYSOPYR STPYR001'
'YEARLY  YEAR020 ' = 'SYSOPYR STPYR002'

              ... etc ...

'+               ' = '+               '
; RUN;
```

The standard CA MICS job name can be found easily. The _USRSTEP macro is invoked with the CA MICS restart step name already in data element RESTEP. Thus, on entry to _USRSTEP code, RESTEP will begin with either DAY, INCR, WEEK, MONTH, YEAR, BKUP, or RSTR.

This sample code for the _USRSTEP macro, with the expanded list in the $USTEP macro, may be used:

```
MACRO _USRSTEP
  LENGTH USER16 USER16N $16 USER8 $8;
  IF RESTEP EQ :'DAY'        THEN USER8='DAILY';
  ELSE IF RESTEP EQ :'INCR'  THEN
          USER8 = 'INCR' || SCAN(PUT(RESTEP,$STEP.),1);
  ELSE IF RESTEP EQ :'WEEK'  THEN USER8='WEEKLY';
  ELSE IF RESTEP EQ :'MONTH' THEN USER8='MONTHLY';
  ELSE IF RESTEP EQ :'YEAR'  THEN USER8='YEARLY';
  USER16 = PUT(USER8,$CHAR8.)||PUT(RESTEP,$CHAR8.);
  USER16N = PUT(USER16,$USTEP.);
  IF USER16N NE USER16 AND LENGTH(USER16N) GT 8
   THEN RESTEP = SUBSTR(USER16N,8);
 %
```

Note that the reverse translation does not require double mapping of job step names. You need only list each job step name once for reverse translation.

INSTALLING:

7. Install the modifications, format, and macro.

The most difficult part of installing step name modifications is the coordination of events. There is no unit-level concatenation of the JCL prototype library PROTOLIB, so all changes to it are complex-wide.

If you are installing CA MICS for the first time, do the following:

a. ___ Apply the changes to PROTOLIB by submitting the IEBUPDTE modifications explained above.

b. ___ During unit database installation, after the COPYLIBU job is run, copy the $USTEP format changes to prefix.MICS.USER.SOURCE(USTEP) and install the _USRSTEP exit. Subsequent steps in the normal unit installation process will complete the installation of the step name modifications.

If you have installed CA MICS previously, and have unit databases in production, do the following:

a. ___ Suspend all JCL generation activity in all unit databases. CA MICS has no facility to guarantee this suspension, so you (the CA MICS administrator) must manually ensure this suspension.

b. ___ Apply the changes to PROTOLIB by submitting the IEBUPDTE modifications explained above.

```
*********************************************************
* Perform steps c to f  for every unit database in      *
* your CA MICS complex.                                 *
*********************************************************
```

c. ___ Enter the Operational Status and Tracking SUSPEND command to suspend operational processing in this unit.

d. ___ Copy the $USTEP format changes to prefix.MICS.USER.SOURCE(USTEP), and install the _USRSTEP exit in the unit database.

e. ___ Submit prefix.MICS.CNTL(JCLGEND). This job regenerates all operational JCL.

f. ___ Enter the Operational Status and Tracking RESUME command to resume operational processing in this unit.

```
*********************************************************
* Perform steps c to f  for every unit database in      *
* your CA MICS complex.                                 *
*********************************************************
```

8. Keep complete written records.

   Earlier sections of this book explain the benefits of keeping accurate documentation on all user modifications to CA MICS. Several suggestions for the form and storage of such documentation are given.

   CA MICS operational job step names are not modified often, for the reasons listed at the beginning of this section. If you elect to make such modifications, you will help ensure your continued satisfaction from product support incidents by having complete documentation of your changes at hand when you call.

2.3.3.3.7 OS Job Attributes

There are three ways in which job attributes can be specified. The JCLDEF PARMS member allows attributes to be assigned to all jobs that are generated.

If some job has unique attributes, two methods are available. PARMS member JCLINFO allows specification of the most common job attributes at the individual job level. Should it be necessary to tailor the job attributes in a way not provided by JCLDEF or JCLINFO, then JOB statements can be generated to the exact installation specifications by using the USERJCL PARMS member.

Section 2.3.3.2.1 describes in detail the JCL definitions that can be specified in the JCLDEF member. These include all of the keywords which are allowed on a standard JOB statement.

The JCLINFO PARMS member is described in Section 2.3.3.3.2.2. JCLINO allows specifications of the JOBNAME, TIMES, LINES, CLASS, and PRTY at the individual job level.

Using USERJCL to define the JOB statement gives complete control of its appearance and content (that is, non-standard keywords can be used, and the order on the JOB statement can be defined). Section 2.3.3.2.2 describes and gives examples of defining JOB statements in the USERJCL PARMS member.

2.3.3.3.8 Writing A User Exit

CA MICS provides a user exit facility that enables you to tailor items produced by the CA MICS JCLGEN process. The exit is intended to be used for changes that are not possible with the basic or extended JCLGEN facilities. For detailed information on writing a JCLGEN exit, see Section 4.3.2.3 of the CA MICS System Modification Guide.

2.3.4 Database Space Modeling Facility

A Database Space Modeling Facility is provided to help estimate the amount of DASD space required for a new database. This process takes as input data from about one day and runs a DAILY job with a temporary database. This temporary database is analyzed together with the online data retention requirements defined to determine the required DASD space (in tracks) for allocating the entire CA MICS database. Three reports are produced quantifying the database space requirements. They are the Database Space Estimator Libname Summary, Information Area Summary, and File Summary reports.

Note that this process is only an approximation. You can allocate your database files with large secondary allocations and begin running database updates. Later you can refine your space allocations based on actual space usage.

The steps involved in estimating the database space requirements include:

1. Collect the required input data using the Installation Preparation Worksheets. Complete the Data Retention worksheets for each product in the unit.

2. Using the worksheets, modify the prefix.MICS.PARMS DBMODEL member.

3. Execute the Database Space Modeling Facility to produce the space analysis reports.

4. Evaluate the results and either modify the inputs and redo step 3 or, if satisfied, continue to step 5.

5. Execute the Database Generation Facility in order to implement the database definition (SAS MACRO generation).

Steps 2 through 5 will normally be performed during installation, but the details are included here so that the entire estimation process can be understood more readily.

Note that when changes are made to database retention definitions, the CYCLEGEN job must be executed in order to generate database cycles according to the new definitions.

The following sections explain database space modeling more fully.

1 - Preparing the Modeling Input
2 - Executing the Database Space Modeling Facility
3 - Analyzing DASD Space Estimates

2.3.4.1 Preparing the Modeling Input

The Installation Preparation Worksheet should have been completed for use in building the required inputs for the database space modeling process.

The DBMODEL parameter is distributed as a prototype in sharedprefix.MICS.GENLIB(DBMDLDEF). Component generations adjust DBMDLDEF for database tailoring. The COPYLIBU job for each database unit adjusts the unit's prefix.MICS.PARMS(DBMODEL) for new products and database tailoring during unit modification or installation.

Update your DBMODEL member based on the data collected on the Installation Preparation Worksheets. This updates the default values on the FILE statements for the products that you are installing.

The Database Space Modeling Facility provides a method for estimating the database's DASD space requirements. The modeling process considers one database per run. Depending upon the user's choice of maintaining a single database or multiple database units, multiple modeling runs may be necessary for different databases.

The modeling facility operates from a series of input statements that define the database information areas, the data retention requirements the user has specified for maintaining online and archive data, and input data of about one day.

There are two input statements that are to be specified as input for the modeling analysis. The statements include the following:

DATABASE: Identifies the database being modeled.

FILE: Specifies the online and archive data retention limits for a specific database file.

Figure 2-34 illustrates the order of the input statements.

```
+---------------------------------+
| DATABASE name                   |
+---------------------------------+-----+
    | FILE filename  retention-limits   |
    | FILE filename  retention-limits   |
    +-----------------------------------+
```

Figure 2-34. Statements for Single System Database

The following sections describe the syntax rules and format for the two input statements and provide an example of their use.

1 - DATABASE Statement
2 - Data Retention Specifications (FILE Statement)

Note: We recommend that the retention limits predefined in DBMODEL be used for your first run. To see an example of DBMODEL, refer to your sharedprefix.MICS.GENLIB(DBMDLDEF).

2.3.4.1.1 DATABASE Statement

The DATABASE statement identifies the database being modeled.

```
STATEMENT FORMAT:
  DATABASE name

OPERAND DESCRIPTIONS:

name:   The name to be associated with this database.

Statement Example:
  DATABASE DATA01
```

2.3.4.1.2 Data Retention Specifications (FILE Statement)

The parameter specifications in this section are critical. After making changes to these parameters, you must execute prefix.MICS.CNTL(CYCLEGEN) before your changes take effect. If archive specifications have changed such that a file is either activated or deactivated, you must also execute prefix.MICS.CNTL(JCLGENU) to regenerate the corresponding job or job step.

Data Retention Specifications tell the database how many cycles of each file, in each supported timespan, to save both online and in archive mode. In addition, Data Retention Specifications also control whether or not eligible audit and history archive files are actually created.

FILE statements must directly follow the DATABASE statement The operands in the FILE statement are free-form and positional. All operands are required.

Files that are not supported within a given timespan are indicated by the presence of a 0 retention limit, while those that are supported have a non-zero value.

Except for the archive timespans, a file that is defined as active within a timespan must have a retention limit greater than zero. (See Special Negative/ Hyphen Flag below.) The sharedprefix.MICS.GENLIB(cccGENIN) member, where ccc is the component identifier, contains FILE statements used by the Component Generator to define a file as active or inactive for each timespan. The retention limits defined in the DBMODEL FILE statement should be consistent with the file definitions contained in the cccGENIN FILE statement used to generate the product.

The archive history timespans (wha and mha below) may have a retention limit of zero in order to deactivate a file in any particular unit. The audit archive timespan (aud) has a unique set of specifications.

We recommend using the initial retention values from the Data Retention portion of the Installation Preparation Worksheets for the first space modeling analysis.

```
STATEMENT FORMAT:
  FILE ccc name retx retd retw retm rety rett wha mha aud

OPERAND DESCRIPTIONS:

  ccc   The 3-character component identifier associated
        with this file.

  name  The six-character CA MICS file identifier.

  retx  The number of days of data being retained in the
        DETAIL timespan of the database for this file.

  retd  The number of days of data being retained in the
        DAYS timespan of the database for this file.

  retw  The number of weeks of data being retained in the
        WEEKS timespan of the database for this file.   The
        maximum value is 99.

  retm  The number of months of data being retained in the
        MONTHS timespan of the database for this file.
        The maximum value is 99.

  rety  The number of years of data being retained in the
        YEARS timespan of the database for this file.   The
        maximum value is 99.

        You may exercise further control over the contents
        of the YEARS timespan using the YEARS timespan
        parameter (refer to Section 2.3.2.4, Site
        Characteristics (SITE)).

  rett  The number of levels of data being retained in the
        TABLES Data Area of the database for this file.
        For any given file, the values for rett MUST be the
        same in all database units.  The maximum value is
        99.
```

wha The number of weeks of data being retained in the WEEKS timespan for the weekly history archive.

The weekly history files are created with "oldmaster + updates => newmaster" logic, where the oldmaster is the previous week's archive tape and the updates are this week's WEEKS files, so each generation of this data set has all the weekly archive data accumulated up to the date of its creation, subject to the cutoff that you specify with this parameter. When the new file is created, data older than the cutoff is dropped in order to prevent these files from growing endlessly.

The maximum value is 156. The minimum value is 0, in which case no weekly history archive is created for this file.

Two other parameters, ARCHIVE HISTW and HISTWGDG, relate to the control of this process. Refer to Sections 2.3.3.2.1.3 and 2.3.3.2.1.5 respectively for descriptions of these parameters. In addition, Section 4.3.13.2 also contains information about wha, ARCHIVE HISTW, and HISTWGDG.

mha The number of months of data being retained in the MONTHS timespan for the monthly history archive.

The monthly history files are created with "oldmaster + updates => newmaster" logic, where the oldmaster is the previous month's archive tape and the updates are this month's MONTHS files, so each generation of this data set has all the monthly archive data accumulated up to the date of its creation, subject to the cutoff that you specify with this parameter. When the new file is created, data older than the cutoff specified here is dropped in order to prevent these files from growing endlessly.

The maximum value is 99. The minimum value is 0, in which case no monthly history archive is created for this file.

Two other parameters, ARCHIVE HISTM and HISTMGDG, relate to the control of this process. Refer to Sections 2.3.3.2.1.3 and 2.3.3.2.1.5 for descriptions of these parameters. In addition, Section 4.3.13.2 also contains information about mha, ARCHIVE HISTM, and HISTMGDG.

aud Specifies whether or not an audit archive tape is created for this file.

- AUDIT(NO) deactivates audit archive for this file. This parameter is used to prevent the creation of an audit archive tape for this file.

- AUDIT(YES) activates audit archive for this file. This parameter is commonly used to document that audit archive is enabled for this file.

  An audit archive tape is created from the DETAIL timespan. If the DETAIL timespan is deactivated, the tape is created from the DAYS timespan.

- AUDIT(DAYS) activates audit archive for this file specifically from the DAYS timespan.

  It may be necessary to use this parameter when an audit archive tape is already created from the DAYS timespan of this file and the normally deactivated DETAIL timespan is activated. Specifying AUDIT(DAYS) continues to create the audit archive tape from the DAYS timespan.

- AUDIT(DETAIL) activates audit archive for this file specifically from the DETAIL timespan.

The default for the AUDIT parameter is taken from the cccGENIN member specification in sharedprefix.MICS.GENLIB. If you want to use this default, no specification of the AUDIT parameter is necessary.

Note: To create an audit archive tape, audit archive must be enabled in both cccGENIN and in prefix.MICS.PARMS(JCLDEF). If audit archive is disabled at the complex or unit levels, it cannot be enabled by specifying AUDIT(YES), AUDIT(DAYS), or AUDIT(DETAIL).

Two other parameters, ARCHIVE AUDIT and AUDITGDG, relate to the control of this process. Refer to Sections 2.3.3.2.1.3 and 2.3.3.2.1.5 respectively for descriptions of these parameters. In addition, Section 4.3.13.2 also contains information about aud, ARCHIVE AUDIT, and AUDITGDG.

Sample FILE Statement:

```
FILE TSO TSOTSO 02 33 09 06 01 00 053 024 AUDIT(NO)
```

Refer to the various chapters on planning, installation definitions, or parameters in the individual product guides for information on FILE statements.

After you complete all changes to prefix.MICS.PARMS(DBMODEL), execute prefix.MICS.CNTL(CYCLEGEN) to put your specifications into effect.

Special Negative/Hyphen Flag

Normally each component's DAYnnn job step deletes each file's highest defined DETAIL and DAYS cycles before writing the day's new files into the database. This is done to conserve disk space. However, when only one cycle has been specified for a file, in other words, when the highest cycle is the only cycle, this delete is not done. The reason is that there would no longer be such a file in the database if the job step continued for a long time or if it abended before the new cycle 01 file could be written.

The administrator of a database unit can delete the 01 cycle even if it is the only cycle of a file. This is done by coding a hyphen (negative sign) as a flag before the retx or retd operand values of 1 on the FILE statement.

An example of such a FILE statement follows:

```
FILE DB2 DB2DSP -1 10 05 03 05 00 025 012
```

WARNING! Before using this option, consider the risk involved. The only existing cycle of a database file will be deleted long before the 00 cycle of the file is ready to take its place. If the job abends before the new cycle 01 is written, other jobs or users attempting to use the file will fail until either that file is restored from backup or a restart of the DAILY job step that abended is successfully completed. It might be appropriate to consider and adjust the backup frequency of that file.

Database Data Retention Definitions Worksheet

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 will be 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 may be specified whether or not 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, you must 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 BAS ADMEXC 00 08 00 06 00 00 000 000
    FILE BAS ADMIHL 00 01 00 00 00 00 000 000
    FILE BAS ADMCYC 00 00 00 00 00 00 000 000
    FILE BAS ADMJHL 00 01 00 00 00 00 000 000
    FILE BAS ADMSPC 00 01 00 00 00 00 000 000
```

Note: The ADMIHL, ADMJHL, and ADMSPC specifications must be given exactly as shown.

```
+--------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET: Database Data Retention Definitions  |
|                                                                          |
| PARMS Library Member is DBMODEL                                          |
|                                                                          |
| Reference Section:  2.3.4.1.2                                            |
+------+-----------------------------------------------+-------------------+
|      |            Online Database Retention          |  Archive Cutoff   |
| File |DETAIL    DAYS   WEEKS   MONTHS  YEARS  TABLES | WEEKS    MONTHS   |
| Name | __(NA)  __(NA)  __(NA)  __(NA)   _(NA)  __(NA)|___ (NA)  ___ (NA) |
+------+-----------------------------------------------+-------------------+
|ADMEXC| 00(00)  __(15)  00(00)  __(12)   0(0)  00(00) |000 (000) 000 (000)|
|ADMIHL| 00(00)  01(01)  00(00)  00(00)   0(0)  00(00) |000 (000) 000 (000)|
|ADMCYC| 00(00)  00(00)  00(00)  00(00)   0(0)  00(00) |000 (000) 000 (000)|
|ADMJHL| 00(00)  01(01)  00(00)  00(00)   0(0)  00(00) |000 (000) 000 (000)|
|ADMSPC| 00(00)  01(01)  00(00)  00(00)   0(0)  00(00) |000 (000) 000 (000)|
+------+-----------------------------------------------+-------------------+
```

Figure 2-36. Database Data Retention Worksheet

2.3.4.2 Executing the Database Space Modeling Facility

Two of the jobs provided in the control statement library, prefix.MICS.CNTL, are:

```
DBSIZE   - Database Space Estimation
CYCLEGEN - Database Cycle Generation
```

After the DBMODEL member in prefix.MICS.PARMS has been updated using FILE statement worksheets, the Database Space Modeling Facility may be executed by submitting the DBSIZE job, as shown below:

```
SUB 'prefix.MICS.CNTL(DBSIZE)'
```

This job needs input data of about one day to estimate the required space to allocate on disk. If some input data cannot be provided, the space estimation will be less accurate.

DBSIZE produces three reports, the Database Space Estimator Libname Summary, Information Area Summary, and File Summary. The Libname Summary shows the space allocation of the physical data sets comprising the database. For new databases, the number of tracks shown in the Projected Space column should be taken as the primary allocation for the ALLOCATE job, together with a safe secondary allocation.

Once the analysis determines a database space estimate in line with the data retention requirements and DASD space available, then execute CYCLEGEN.

DBSIZE can also be used to estimate database size after configuration changes, like adding a component to a unit. Since, in this case, the new configuration must be generated before DBSIZE can run, the existing DAILY job may be affected. If the DAILY job has additional data integration steps generated, SYSPARM='NODATA' should be coded on the EXEC JCL card for the corresponding step until you are ready to provide the new data and run it into the database. For example, if you have generated a database unit to contain TSO, RMF, and SMF data and you don't have TSO data available, the EXEC JCL card for the TSO step should look like:

```
//DAY010 EXEC MICSDBx,SYSPARM='NODATA'
```

where MICSDBx is the name of the procedure in the jobstream.

2.3.4.3 Analyzing DASD Space Estimates

The Database Estimator is used to estimate the size of a database. The estimation is based on files in the temporary database created with the DBSIZE job and from information in prefix.MICS.PARMS(DBMODEL). Space is reported in tracks.

The following reports are produced:

- o The Libname Summary Report provides overview information for each timespan in the database.
- o The Information Area Summary Report shows space utilization for each Information Area within a timespan.
- o The File Summary Report shows space utilization for each file in a timespan. Note that all cycles for a file are comprised in a single line entry.

For further information regarding space utilization reporting and the Database Estimator, refer to section 4.4.4.6 of this guide.

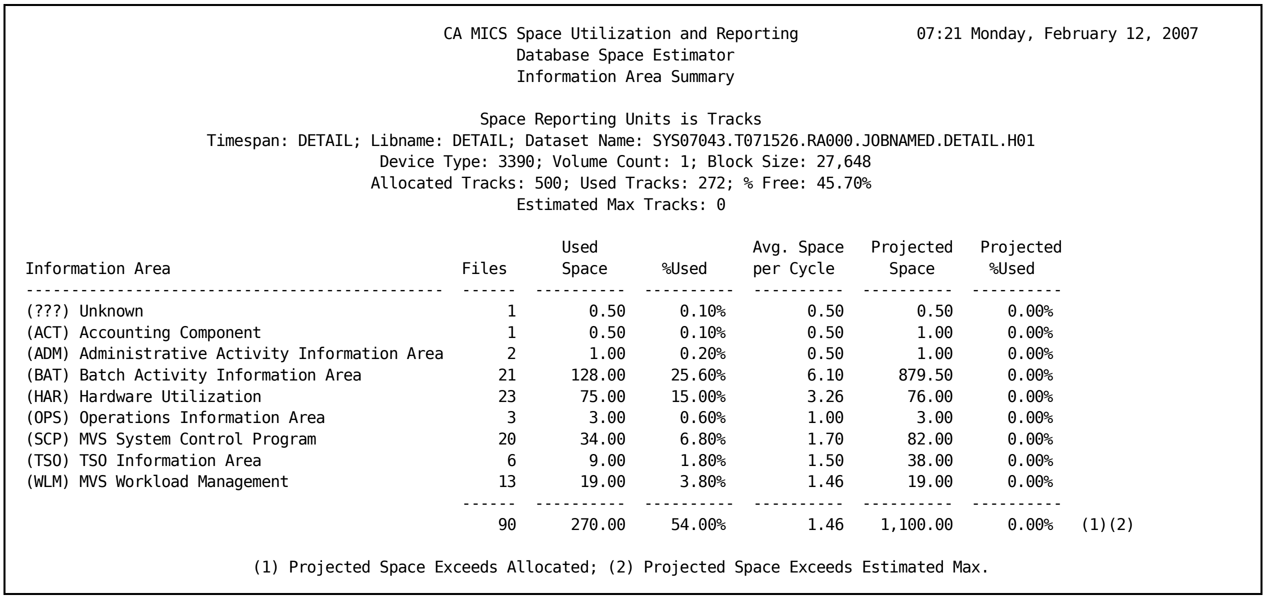
A sample report listing for the Database Estimator follows:

The Libname Summary Report lists space utilizations for each physical data set timespan of the database.

```
                                         CA MICS Space Utilization and Reporting           07:21 Monday, February 12, 2007
                                                 Database Space Estimator
                                                     Libname Summary

                                             Space Reporting Unit is Tracks
                                                                                                            Est. Max
                                                                                Used     Allocated  Projected  Allocated
Libname   Dataset Name                                          Block Size     Space       Space      Space      Space
--------  --------------------------------------------  ----------  ----------  ----------  ----------  ----------
DETAIL    SYS07043.T071526.RA000.JOBNAMED.DETAIL.H01          27,648      271.50      500.00    1,100.00        0.00  (1)(2)
DAYS      SYS07043.T071526.RA000.JOBNAMED.DAYS.H01            27,648      122.50      500.00    1,780.50        0.00  (1)(2)
WEEKS     SYS07043.T071526.RA000.JOBNAMED.WEEKS.H01           27,648      116.50      500.00      723.50        0.00  (1)(2)
MONTHS    SYS07043.T071526.RA000.JOBNAMED.MONTHS.H01          27,648      107.50      500.00      546.00        0.00  (1)(2)
YEARS     SYS07043.T071526.RA000.JOBNAMED.YEARS.H01           27,648        2.00      500.00        0.50        0.00  (2)
                                                                        ----------  ----------  ----------  ----------
                                                                            620.00    2,500.00    4,150.50        0.00

                         (1) Projected Space Exceeds Allocated; (2) Projected Space Exceeds Estimated Max.
```

```
Column Descriptions
-------------------

 Libname: The SAS libname of the timespan.

 Dataset name: The z/OS data set name of the timespan.

 Block size: The DCB block size of the data set.

 Used Space: Space currently used in the timespan.

 Allocated Space: Total space currently allocated to the
 timespan. This is computed as Used + Free space.

 Projected Space: Estimate of used space within a timespan
 when all files are at maximum cycles.

 Estimated Max Allocated Space: Estimate of the maximum size
 of the timespan if the maximum size of each remaining
 secondary extent is used.
```

The Information Area Summary Report shows space utilization for each Information Area within the timespan.



```
                                    CA MICS Space Utilization and Reporting              07:21 Monday, February 12, 2007
                                            Database Space Estimator
                                            Information Area Summary

                                        Space Reporting Units is Tracks
                      Timespan: DETAIL; Libname: DETAIL; Dataset Name: SYS07043.T071526.RA000.JOBNAMED.DETAIL.H01
                                   Device Type: 3390; Volume Count: 1; Block Size: 27,648
                                  Allocated Tracks: 500; Used Tracks: 272; % Free: 45.70%
                                            Estimated Max Tracks: 0

                                                             Used                    Avg. Space   Projected   Projected
Information Area                                Files       Space        %Used       per Cycle       Space       %Used
----------------------------------------------  ------  ----------  ----------  ----------  ----------  ----------
(???) Unknown                                        1        0.50       0.10%        0.50        0.50       0.00%
(ACT) Accounting Component                           1        0.50       0.10%        0.50        1.00       0.00%
(ADM) Administrative Activity Information Area       2        1.00       0.20%        0.50        1.00       0.00%
(BAT) Batch Activity Information Area               21      128.00      25.60%        6.10      879.50       0.00%
(HAR) Hardware Utilization                          23       75.00      15.00%        3.26       76.00       0.00%
(OPS) Operations Information Area                    3        3.00       0.60%        1.00        3.00       0.00%
(SCP) MVS System Control Program                    20       34.00       6.80%        1.70       82.00       0.00%
(TSO) TSO Information Area                           6        9.00       1.80%        1.50       38.00       0.00%
(WLM) MVS Workload Management                       13       19.00       3.80%        1.46       19.00       0.00%
                                                ------  ----------  ----------  ----------  ----------  ----------
                                                    90      270.00      54.00%        1.46    1,100.00       0.00%   (1)(2)

                         (1) Projected Space Exceeds Allocated; (2) Projected Space Exceeds Estimated Max.
```

```
Heading Descriptions
------- -------------

 Timespan: The database timespan.

 Libname: The DDname used to reference the timespan data set.

 Dataset Name: The fully qualified data set name of the
 timespan.

 Device Type: The device type of the data set. Note that the
 device type is the same for all volumes of a multi-volume
 data set.

 Volume Count: Number of volumes of the timespan data set.

 Block Size: The z/OS block size of the data set.

 % Free: Percent of free space in the timespan data set.

 Allocated units: The amount of space currently allocated to
 timespan data set.

 Used units: The amount of space currently used in the
 timespan data set.

 Estimated Max units: The estimated maximum space of the
 timespan data set. This is computed by adding the requested
 size of the remaining secondary extents on the last volume
 of the timespan data set to the allocated space.

Column Descriptions
-------------------

 Information Area: Name the Information Area within the
 timespan.

 Files: Total members in this Information Area. This includes
 valid and invalid (as listed in the Invalid Files report)
 file entries.

 Used Space: Space currently used in the Information Area.
 Note that the space used at the Libname summarization level
 may not equal the allocated space in the timespan.  This
 difference is attributed to internal SAS usage.

 %Used: Percentage of allocated space used by the Information
 Area within the timespan.
```

```
Avg Space/Cycle: Average space used by each cycle of the
file within the timespan.

Projected Space: Space used by this information area when
all cycles of all files are created.

Projected %Used: Percentage of the Estimated Maximum used.
```

The File Summary Report shows space utilization for each file within the timespan.

```
                                    CA MICS Space Utilization and Reporting             07:21 Monday, February 12, 2007
                                            Database Space Estimator
                                                  File Summary

                                          Space Reporting Units is Tracks
                    Timespan: DETAIL; Libname: DETAIL; Dataset Name: SYS07043.T071526.RA000.JOBNAMED.DETAIL.H01
                                    Device Type: 3390; Volume Count: 1; Block Size: 27,648
                                   Allocated Tracks: 500; Used Tracks: 272; % Free: 45.70%
                                            Estimated Max Tracks: 0

Information Area                                             Used                Avg. Space  Def'd  Estimated   Estimated
   File Name                                     Cycles      Space      %Used    per Cycle  Cycles    Space       %Used
-------------------------------------------------  ------  ----------  --------  ----------  ------  ----------  ----------
   ...
(BAT) Batch Activity Information Area
   (_OE) OPEN EDITION/MVS PROGRAM FILE                   1        2.00     0.40%        2.00       0        2.00       0.00%
   (_SA) SYSTEM ADDRESS SPACE ACTIVITY FILE              1        2.00     0.40%        2.00       0        2.00       0.00%
   (_ST) SYSTEM TASK PROGRAM ACTIVITY FILE               1        3.00     0.60%        3.00       0        3.00       0.00%
   (_TP) APPC/MVS TP ACTIVITY FILE                       1        2.00     0.40%        2.00       0        2.00       0.00%
   (_TS) SMF USER TSO ACTIVITY FILE                      1        2.00     0.40%        2.00       0        2.00       0.00%
   (ATP) APPC/MVS TRANSACTION FILE                       1        1.00     0.20%        1.00       0        1.00       0.00%
   (JOB) BATCH USER JOB ACTIVITY FILE                    1        2.00     0.40%        2.00       0        2.00       0.00%
   (MUA) MEASURED USAGE ADDRESS SPACE FILE               1        1.00     0.20%        1.00       0        1.00       0.00%
   (MUG) MEASURED USAGE GLOBAL FILE                      1        1.00     0.20%        1.00      30       30.00       0.00%
   (OEP) OPEN EDITION PROCESS ACTIVITY FILE              1        1.50     0.30%        1.50       0        1.50       0.00%
   (PGM) BATCH USER PROGRAM ACTIVITY FILE                1        3.00     0.60%        3.00       0        3.00       0.00%
   (REN) MULTISYSTEM ENCLAVE ACTIVITY FILE               1        1.00     0.20%        1.00       0        1.00       0.00%
   (SPL) BATCH USER SPOOL ACTIVITY FILE                  1        1.00     0.20%        1.00       0        1.00       0.00%
   (WDA) BATCH WORKLOAD DEVICE ACTIVITY FILE             1        2.00     0.40%        2.00       1        2.00       0.00%
   ----------------------------------------------          ----------  --------                      ----------  ----------
(BAT) Batch Activity Information Area                          24.50     4.90%                           53.50       0.00%
   ...
   ==============================================          ==========  ========                      ==========  ==========
                                                              121.00    24.20%                         1,780.50       0.00%   (1)

                         (1) Projected Space Exceeds Allocated; (2) Projected Space Exceeds Estimated Max.
```

```
Heading Descriptions
------- -------------

 Timespan: The database timespan.

 Libname: The DDname used to reference the timespan data set.

 Dataset Name: The fully qualified data set name of the
 timespan.

 Device Type: The device type of the data set. Note that the
 device type is the same for all volumes of a multi-volume
 data set.

 Volume Count: Number of volumes of the timespan data set.

 Block Size: The z/OS block size of the data set.

 % Free: Percent of free space in the timespan data set.

 Allocated units: The amount of space currently allocated to
 timespan data set.

 Used units: The amount of space currently used in the
 timespan data set.

 Estimated Max units: The estimated maximum space of the
 timespan data set. This is computed by adding the requested
 size of the remaining secondary extents on the last volume
 of the timespan data set to the allocated space.

Column Descriptions
-------------------

 Information Area: Name the information area within the
 timespan.

 File Name: Name the file area within the timespan.

 Cycles: Total iterations (cycles) of this file within the
 timespan.

 Used Space: Space currently used by the file.  Note that the
 space used at the Libname summarization level may not equal
 the allocated space in the timespan.  This difference is
 attributed to internal SAS usage.

 %Used: Percentage of allocated space used by the File within
 the timespan.
```

```
Avg Space/Cycle: Average space used by each cycle within the
timespan.

Def'D Cycles: Number of cycles as defined in
prefix.MICS.PARMS(DBMODEL).

Estimated Space: Space used by this file area when all
cycles are created.

Estimated %Used: Percent of the Estimated Maximum used.
```

2.3.5 Dynamic Execution Options (EXECDEF)

Certain parameters that affect the execution of the standard CA MICS jobs can be altered at any time without regenerating the CA MICS system. These parameters let you specify whether CA MICS will run some, all, or none of the exception modules for the installed system. They also let you specify report options, weekend scheduling options, and certain data retention options.

These parameters are specified by changing the EXECDEF member of MICS.PARMS. A sample EXECDEF member is listed below, followed by an explanation of each parameter statement.

Note: We strongly recommend that you use the default member when CA MICS is first installed. After some experience is gained with the system, the parameters can be adjusted as needed using the worksheet in Figure 2-43. Note that changes made to EXECDEF take effect with the next CA MICS job and do not require any generation process.

```
 *
 *      EXECUTION OPTION DEFINITIONS
 *
  CREATE EXCEPTIONFILES NONE
  REPORT EXCEPTIONS NONE
  REPORT MBODAILY   ALL
  REPORT MBOWEEKLY  ALL
  REPORT MBOMONTHLY ALL
  REPORT WORKSPACE  NO
  SCHEDULE   WEEKEND NO
  INPUTLOGLIMIT 99
  SMFDJHLLIMIT 99
  SPCUTILLIMIT 99
  EXCLUDEORGSYSIDS T168 T033
  TITLE CA MICS PRIMARY DATABASE
```

Figure 2-42. Sample EXECDEF Member

CA MICS supports a separate exception module for each of its products. Exception modules read newly-created cycles, looking for observations that meet exception criteria. Each site defines its own exception criteria. (For detailed information about tailoring exceptions, see the Standard Reports Guide.) When exception modules detect what they consider to be an exceptional condition, they record it in a record in the DAYS.ADMEXC file. The following statement allows you to control this process.

CREATE EXCEPTIONFILES:

Specify NONE, ALL, or a list of one or more CA MICS products to define if the DAILY exception modules should be run. For more information on generating exceptions in the DAILY update job, see Chapter 2, "Exception Reports," in the Standard Reports Guide.

Whether any exceptions that are found are reported is determined by the REPORT EXCEPTIONS parameter below or by your MICF inquiry production reporting definitions. You can choose to create the exception files, but not activate daily reporting in order to have exceptions available for online inquiry using MICF.

NONE - Do not run any exception modules (the DAYS.ADMEXCnn files will exist, but will always be empty).

ALL - Run the exception modules for all installed CA MICS products.

list - Run the exception modules for the listed CA MICS products. Product IDs are separated by blanks.

Note that all exceptions identified by all exception modules go into a single CYCLE of the DAYS.ADMEXC file with one record per exception detected. The INFOAREA field of the records indirectly identifies the product.

REPORT EXCEPTIONS:

Specify NONE or a list of one or more levels of exception reporting that should be done for all CA MICS products whose exception modules were run in the DAILY update job. See the CREATE EXCEPTIONFILES discussion above for more information on creating exceptions.

Specify NONE if you are using MICF inquiry production reporting facilities to generate daily CA MICS exception reports. For more information on executing daily MICF inquiry production reports with the CA MICS DAILY database update, see the MICF Reference Guide.

NONE - Do not produce any Daily Exception Reports.

OVERVIEW - Produce the Daily Exception Report Overview.

SEVERITY - Produce the Daily Exception Report list of

```
                 exceptions by severity level (Critical,
                 Impacting, and Warning).

     MGMTAREA  - Produce the Daily Exception Report list of
                 exceptions by Management Area (Performance,
                 Security, Standards, Workload, and so on).

     DETAIL    - Produce the Daily Exception Report detail
                 list of exceptions.
```

The standard report options control the production of the Daily, Weekly, and Monthly management objective (MBO) reports (see the Standard Reports Guide). Each report option must specify NONE for no reports, ALL for MBO reports from all installed products, or a list of one or more CA MICS product names. For more information, see the COMPONENTS statement in Section 2.3.3.2.1.1, Control Definitions.

Specify NONE if you are using MICF inquiry production reporting facilities to generate Daily, Weekly, and Monthly MBO reports. For more information on executing daily, weekly, and monthly MICF inquiry production reports with the CA MICS daily, weekly, and monthly production processing, see the MICF Reference Guide.

REPORT MBODAILY:

Specify NONE, ALL, or a list of one or more CA MICS products to define if the standard Daily Management Objective Reports should be produced.

```
NONE - Do not produce any Daily Management Objective
       Reports.

ALL  - Produce the Daily Management Objective Reports
       for all installed CA MICS products.

list - Produce the Daily Management Objective Reports
       for the CA MICS products listed.  Product IDs are
       separated by blanks.
```

REPORT MBOWEEKLY:

Specify NONE, ALL, or a list of one or more CA MICS products to define if the standard Weekly Management Objective Reports should be produced.

```
NONE - Do not produce any Weekly Management Objective
       Reports.
```

```
     ALL  - Produce the Weekly Management Objective Reports
            for all installed CA MICS products.

     list - Produce the Weekly Management Objective Reports
            for the CA MICS products listed.  Product IDs
            are separated by blanks.

REPORT MBOMONTHLY:

     Specify NONE, ALL, or a list of one or more CA MICS
     products to define if the standard Monthly Management
     Objective Reports should be produced.

     NONE - Do not produce any Monthly Management Objective
            Reports.

     ALL  - Produce the Monthly Management Objective Reports
            for all installed CA MICS products.

     list - Produce the Monthly Management Objective Reports
            for the CA MICS products listed.  Product IDs are
            separated by blanks.

REPORT WORKSPACE:

     Specify YES or NO to control whether or not the MICS
     Work Space Utilization Report is to be generated.  If
     left blank or unspecified, the default is NO.  For more
     information refer to section 4.3.15.

SCHEDULE WEEKEND:

     A process for automatically scheduling the required
     weekly, monthly, and yearly processes is provided
     through the Job Scheduling Facility.  A specification of
     YES for this parameter allows the Job Scheduling
     Facility to schedule weekly, monthly, or yearly
     operations on a weekend; a specification of NO will
     ensure that THEY ARE NOT SCHEDULED ON THE WEEKEND.

     The rationale for this option is that the weekly,
     monthly, and yearly processes are more complex than the
     daily one, especially with respect to their tape
     processing.  This results in a greater chance that they
     will malfunction and require human intervention, which
     might not be available on a weekend.

INPUT HISTORY LOG RETENTION LIMIT (INPUTLOGLIMIT):
```

The Input History Log file of the Administrative Information Area (DAYS.ADMIHL01) provides the system administrator with an audit trail (by original SYSID and CA MICS product) of the types and volume of data that have been used to update the CA MICS database. The audit trail data is maintained for a given number of days; the number of days is defined through this option. The number of days can range from a minimum of 7 days to a maximum of 99 days. Although, the space required for each cycle of this information is small, we recommend that you use a large number of days for the audit trail data. If this statement is omitted or invalid, a default of 99 days is used.

CA SMF DIRECTOR INPUT HISTORY RETENTION LIMIT (SMFDJHLLIMIT):

The Split Index File of the Administrative Information Area (DAYS.ADMJHL01) is only populated when input is provided using the CA SMF Director interface to CA MICS. This file serves a dual purpose. First, it is input into each CA MICS incremental update and DAILY input processing step to eliminate split files previously processed within the same CA MICS component that would otherwise be considered eligible for input based on standard date/time selection criteria. Second, it provides the system administrator with an audit trail of the split files processed by each CA MICS component to update the CA MICS database. The audit trail data is maintained for a given number of days; the number of days is defined through this option. The number of days can range from a minimum of 7 days to a maximum of 99 days. Since CA MICS uses this file to avoid processing previously processed split files, we recommend that you use a large number of days for the audit trail data. If this statement is omitted or invalid, a default of 99 days is used.

Note: The SMFDJHLLIMIT value should coincide with the CA SMF Director INDEXLIM parameter. If the INDEXLIM value is set higher than the CA MICS SMFDJHLLIMIT, the CA SMF Director index will contain entries that CA MICS has already discarded. This could potentially cause CA MICS to input split files that were previously processed but still remain eligible for input based on the standard date/time selection criteria. While CA MICS has safeguards built-in to eliminate this duplicate data, processing previously processed data will add to the overall processing time and should be avoided.

SPACE UTILIZATION FILE (ADMSPC) RETENTION LIMIT (SPCUTILLIMIT):

The Space Utilization file of the Administrative Information Area (DAYS.ADMSPC01) provides the system administrator with a snapshot (by CA MICS product step, and type) of the space usage of the INCRccc and DAILY jobs. This includes work space as well as unit database space utilization. Data is maintained for the number of days defined through this option. The number of days can range from a minimum of 7 days to a maximum of 99 days. If this statement is omitted or invalid, a default of 99 days is used.

EXCLUDE ORIGINAL SYSIDS (EXCLUDEORGSYSIDS):

This statement can be used to exclude data from selected original SYSIDs from CA MICS processing. Data for any ORGSYSID specified will not be added to the CA MICS database, and will not be checked against the table of original SYSIDs defined to CA MICS in the SYSID (unit-level) or CPLXSID (complex-level) parameter members.

The statement is specified by providing the keyword and a list of one or more original SYSIDs separated by blanks. For example:

```
EXCLUDEORGSYSIDS TST1 TST4 SYST
```

Use the keyword, UNDEFINEDSYSID, to override an UNDEFINEDSYSID ABORT parameter you may have specified or accepted by default in sharedprefix.MICS.PARMS(CPLXSID) or prefix.MICS.PARMS(SYSID). Specify the following statement to automatically exclude data for any ORGSYSID value NOT defined in either prefix.MICS.PARMS(SYSID) or sharedprefix.MICS.PARMS(CPLXSID).

```
EXCLUDEORGSYSIDS UNDEFINEDSYSID
```

Note: The EXCLUDEORGSYSIDS UNDEFINEDSYSID option is intended to address temporary or isolated situations. Use the CPLXSID or SYSID control parameters for "permanent" changes.

TITLE

The TITLE statement will cause a title to be written at the beginning of the SAS log for any processing step that uses #BASMSTR. This feature is primarily intended for identifying jobs using CA MICS special databases, but can be used for any database. If used, the title text must be separated from the TITLE keyword by a blank and must not contain any semicolons. The title does not need to be enclosed in quotes. A TITLE statement must be only one line, but multiple TITLE statements can be specified.

ACCOUNTING

Specify ACTIVE or INACTIVE to define the status of the Accounting product in this database unit. This statement will turn the Accounting product on or off for all files it uses in this unit. This statement does not need to be specified if Accounting is not installed in this database unit.

SD_ACTIVE

Specify YES to activate or NO to disable CA MICS Abend Reporting. When activated, CA MICS will generate call tickets in Service Desk for most abends in CA MICS operational jobs. See Section 4.9 for more information.

USERDEF

This parameter allows you to define a symbol and a value that can be substituted at SAS execution. This substitution can only occur in CA MICS jobs that invoke the SETGBLS SAS routine to input the USERDEF statements in EXECDEF, such as DAILY, WEEKLY, MONTHLY, and YEARLY. The CALL GBLFCN function is used to retrieve the value of the symbol in SAS code and assign the value to a user-supplied SAS element. The USERDEF parameter can be used in a user modification to provide user-supplied parameter values, and the user modification code can perform specific functions based on the USERDEF parameter setting for a specific database unit.

For example, the SMF Analyzer Account Routine in sharedprefix.MICS.PARMS(ACCTRTE) could use different input logic to assign the Account Code (ACCTNOn) elements depending upon the unit database being updated by the DAILY job. The CA MICS database unit being processed in the DAILY update job can be determined by setting a unique value for a specific USERDEF parameter symbol in prefix.MICS.PARMS(EXECDEF) for each database unit that contains the SMF Analyzer.

Several USERDEF symbols can be defined. The only restriction is that the symbol on each USERDEF statement must be different from those on all others and different from all symbols currently used by SETGBLS. Also, the value must end before column 73 of the record. It cannot be continued and must contain no imbedded blanks.

Here is an example:

```
USERDEF CLIENT GROUP1
```

This defines a symbol, CLIENT. This symbol can be accessed using the CALL GBLFCN function to substitute the symbol value for CLIENT as coded in EXECDEF. For more information, see Chapter 4 of the System Modification Guide.

There are additional statements that can be specified in the EXECDEF parameter member. These statements are used for the CA MICS special database facility. They include the COMPONENT, FORCE, SELECT, and ORGSYSID statements. These statements are discussed in Appendix A.3.

```
+-------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET: Dynamic Execution Options          |
| PARMS Library Member is EXECDEF                                         |
| Reference Section: 2.3.5                                                |
+-------------------------------------------------------------------------+
|                                                                         |
|                                                                         |
|                                                                         |
|   CREATE    EXCEPTIONFILES _________ (NONE, ALL, list)                  |
|   REPORT    EXCEPTIONS     _________ (NONE, category)                   |
|                                                                         |
|   REPORT    MBODAILY       _________ (NONE, ALL, list)                  |
|   REPORT    MBOWEEKLY      _________ (NONE, ALL, list)                  |
|   REPORT    MBOMONTHLY     _________ (NONE, ALL, list)                  |
|   REPORT    WORKSPACE      _________ (YES, NO)                          |
|                                                                         |
|   SCHEDULE  WEEKEND        _________ (YES, NO)                          |
|                                                                         |
|   INPUTLOGLIMIT __ (7 - 99)                                             |
|   SMFDJHLLIMIT  __ (7 - 99)                                             |
|   EXCLUDEORGSYSIDS  ____ ____ ____   (UNDEFINEDSYSID, list)             |
|                                                                         |
|   TITLE _______________________________                                 |
|                                                                         |
|   ACCOUNTING ________ (ACTIVE, INACTIVE) (if required)                  |
|                                                                         |
|   SD_ACTIVE  ________ (YES, NO)          (if required)                  |
|                                                                         |
|                                                                         |
+-------------------------------------------------------------------------+
```

Figure 2-43. Dynamic Execution Options Worksheet

2.3.6 Dynamic Allocation Parameter Overrides (//PARMOVRD)

Many CA MICS facilities, including Internal Step Restart and Incremental Update utilize z/OS dynamic data set allocation services for both new and existing DASD data sets. New data sets are allocated according to the data set allocation parameters you specify in sharedprefix.MICS.PARMS(JCLDEFC), prefix.MICS.PARMS(JCLDEF), and/or prefix.MICS.PARMS(cccOPS).

The normal procedure for changing allocation specifications for dynamically allocated data sets is to update the parameter member and then execute the associated JCLGENx and/or cccPGEN jobs. However, in certain situations, such as when recovering from a production job abend, you may need to temporarily override data set allocation parameters for one or more dynamically allocated data sets. CA MICS supports a special ddname where you can specify these execution-time data set allocation parameter overrides.

To enable the dynamic allocation parameter override facility, add a PARMOVRD DD statement to the CA MICS operational job step where the overrides are needed. For example, you might add,

```
//PARMOVRD DD *
```

This DD statement would be followed by the parameter statements containing the overrides you want to use.

The PARMOVRD facility supports execution-time overrides for:

- The prefix.MICS.PARMS(cccOPS) WORK parameter when used in conjunction with Internal Step Restart. When operating with RESTART YES, you can override the data set allocation parameters for the WORKnn DDs allocated for database update processing.

  Note: You can NOT change the number of work files at execution-time. The PARMOVRD facility can ONLY override the data set allocation parameters (for example, to increase the amount of work space allocated for the step).

- The prefix.MICS.PARMS(cccOPS) RESTARTWORK and RESTARTCKPT Internal Step Restart parameters.

- The prefix.MICS.PARMS(cccOPS) INCRDETAIL and INCRDAYS Incremental Update parameters.

- o The DYNAMWAIT parameter. DYNAMWAIT may be specified in prefix.MICS.PARMS(cccOPS), prefix.MICS.PARMS(JCLDEF), and/or sharedprefix.MICS.PARMS(JCLDEFC).

The PARMOVRD parameter syntax is very similar to that used in prefix.MICS.PARMS(cccOPS). Specify the keyword followed by the data set allocation parameter specifications. If multiple lines are required, simply repeat the keyword on subsequent parameter lines. For example,

```
//PARMOVRD DD *
 WORK   SPACE=(CYL,(50,50)) STORCLAS=MICSTEMP
 RESTARTWORK SPACE=(CYL,(50,50))
 RESTARTWORK STORCLAS=MICSTEMP
```

Note: The PARMOVRD data set allocation parameters are NOT validated. The parameters will be used exactly as specified for data set allocations. Syntax errors will be detected when the data set allocation fails. If you have an operational job that is failing with data set allocation errors, be sure to check first for invalid PARMOVRD parameter specifications.

In addition, remember that PARMOVRD specifications completely replace the corresponding cccOPS, JCLDEF, and/or JCLDEFC specifications for the duration of the job step. You must specify a complete set of data set allocation parameters for the associated data sets. Partial specifications will result in data set allocation errors and/or unexpected (and usually undesirable) results.

The remainder of this section describes the parameters available for PARMOVRD specification.

WORK

To override the data set allocation parameters for multiple WORK file support, specify the following WORK statement:

```
WORK  data_set_allocation_parameters
```

data_set_allocation_parameters is one or more data set allocation parameters (for example, STORCLAS or SPACE), separated by spaces.

NOTE: WORK is ignored if Multiple Work File Support is not

```
       active or if Internal Step Restart is not enabled for
       this product.
```

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. 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. If you need multiple lines, repeat the WORK on the continuation line.

Note: DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

```
  WORK  STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss))
```

Example 2 (continuation):

```
  WORK  STORCLAS=MICSTEMP UNIT=SYSDA
  WORK  SPACE=(xxxx,(pppp,ssss),,,ROUND))

  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.

             xxxx is TRK, CYL, or blklen
             pppp is the primary allocation
             ssss 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.
```

See the individual Product Guides for more information on the Multiple Work File Facility, the WORK statement, and any product specific considerations.

RESTARTCKPT

To override data set allocation parameters for the Internal Step Restart checkpoint data set, specify the following

```
RESTARTCKPT statement:

     RESTARTCKPT   data_set_allocation_parameters

       data_set_allocation_parameters is one or more data set
        allocation parameters (for example, STORCLAS or
        SPACE), separated by spaces.

Note:  RESTARTCKPT is ignored if Internal Step Restart is not
       enabled for this product.

Specify data set allocation parameters, separated by blanks,
according to SAS LIBNAME statement syntax.  If you need
multiple lines, repeat the RESTARTCKPT on the continuation
line.

RESTARTCKPT accepts the engine/host options documented in the
"SAS Companion for z/OS", including STORCLAS,
UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Note:  DO NOT SPECIFY THE DISP PARAMETER.


Example 1:

RESTARTCKPT  STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)


Example 2 (multiple lines):

RESTARTCKPT  STORCLAS=MICSTEMP UNIT=SYSDA
RESTARTCKPT  SPACE=(xxxx,(pp,ss),,,ROUND)

  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.

             xxxx is TRK, CYL, or blklen
             pppp is the primary allocation
             ssss 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.
```

See the individual Product Guides for more information on Internal Step Restart, the RESTARTCKPT statement, and any product specific considerations.

RESTARTWORK

To override data set allocation parameters for the Internal Step Restart WORK data set, specify the following RESTARTCKPT statement:

```
     RESTARTWORK   data_set_allocation_parameters

       data_set_allocation_parameters is one or more data
        set allocation parameters (e.g., STORCLAS or SPACE),
        separated by spaces.
```

Note: RESTARTWORK is ignored if Internal Step Restart is not enabled for this product.

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 z/OS", including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Note: DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

```
  RESTARTWORK  STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

Example 2 (multiple lines):

```
  RESTARTWORK  STORCLAS=MICSTEMP UNIT=SYSDA
  RESTARTWORK  SPACE=(xxxx,(pp,ss),,,ROUND)
```

```
  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.

             xxxx is TRK, CYL, or blklen
             pppp is the primary allocation
             ssss 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.


See the individual Product Guides for more information on
Internal Step Restart, the RESTARTWORK statement, and any
product specific considerations.


INCRDETAIL
----------

To override data set allocation parameters for the
incremental update DETAIL data set (IUDETAIL), specify the
following INCRDETAIL statement:

     INCRDETAIL    data_set_allocation_parameters

       data_set_allocation_parameters is one or more data set
        allocation parameters (for example, STORCLAS or
        SPACE), separated by spaces.

Note:  INCRDETAIL is ignored if Incremental Update is not
       enabled for this product or if the INCRDB option is
       PERM.

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.

INCRDETAIL accepts the engine/host options documented in "SAS
Companion for z/OS", including STORCLAS,
UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Note:  DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

INCRDETAIL   STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

Example 2 (multiple lines):

```
INCRDETAIL   STORCLAS=MICSTEMP UNIT=SYSDA
INCRDETAIL   SPACE=(xxxx,(pp,ss),,,ROUND)

  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.

             xxxx is TRK, CYL, or blklen
             pppp is the primary allocation
             ssss 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.
```

See the individual Product Guides for more information on Incremental Update, the INCRDETAIL statement, and any product specific considerations.

INCRDAYS

To override data set allocation parameters for the incremental update DAYS data set (IUDAYS), specify the following INCRDAYS statement:

```
     INCRDAYS      data_set_allocation_parameters

       data_set_allocation_parameters is one or more data set
        allocation parameters (for example, STORCLAS or
        SPACE), separated by spaces.
```

Note: INCRDAYS is ignored if Incremental Update is not enabled for this product or if the INCRDB option is PERM.

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.

```
INCRDAYS accepts the engine/host options documented in "SAS
Companion for z/OS", including STORCLAS,
UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Note:  DO NOT SPECIFY THE DISP PARAMETER.


Example 1:

INCRDAYS     STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)


Example 2 (multiple lines):

INCRDAYS     STORCLAS=MICSTEMP UNIT=SYSDA
INCRDAYS     SPACE=(xxxx,(pp,ss),,,ROUND)

  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, where:

             xxxx is TRK, CYL, or blklen
             pppp is the primary allocation
             ssss 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.


See the individual Product Guides for more information on
Incremental Update, the INCRDAYS statement, and any product
specific considerations.

DYNAMWAIT
---------

If you want to override the amount of time that the DAILY
and/or INCRccc job waits for an unavailable data set, you
must specify the following DYNAMWAIT statement:

     DYNAMWAIT   minutes

     where minutes is the amount of time in minutes to wait.
```

```
See the individual Product Guides for more information on
Internal Step Restart, Incremental Update, the DYNAMWAIT
statement, and any product specific considerations.

SMFD_SELECT
-----------

To specify a date-time range selection for data sets read
from the CA SMF Director duplicate split indices, specify:

SMFD_SELECT starttime endtime

 starttime is the beginning of the range in the standard SAS
 datetime format: ddmonyyyy:hh:mm:ss.hh

 endtime is the end of the range in the standard
 SAS datetime format: ddmonyyyy:hh:mm:ss.hh

For example,
SMFD_SELECT 16JAN2007:08:00:01.00 16JAN2007:20:00:00.00
```

Chapter 3: Installation

This chapter describes the tasks involved in installing the CA MICS system.

As was mentioned in the previous section, the installation should not be undertaken without first comprehensively reviewing the material presented in Chapter 2, Planning for the Installation and Use of CA MICS". Specifically, Section 2.3, "CA MICS Installation Planning and Parameter Specification" should be covered in detail and the Installation Preparation Worksheets should be completed so that the information will be available for the installation process. It is also vital that the contents of the CA MICS System Administrator Guide be checked for any bulletins relevant to the installation process.

Typically, a user will be performing one or more of the five installation functions listed below. Any of these functions can be accomplished by performing some combination of CA MICS installation "stages", as described in section 3.8. A complete checklist is included in the subsections of section 3.8 for each of the five functions.

1. Installing a CA MICS complex for the first time

2. Adding a new product to an existing database unit in an existing database complex

3. Adding a new product to a new database unit in an existing database complex

4. Adding a new database unit in an existing CA MICS complex and not adding new products

5. Splitting an existing database unit

```
Installation is described in the first seven sections listed
below, and the installation checklists are contained in the
last section.
```

This section contains the following topics:

3.1 Installation Preparation

Please review chapter 2, especially the discussion of planning and parameter specification in Section 2.3, complete the Installation Prepartion Worksheets, and review the CA MICS System Administrator Guide for announcements that might pertain to your data center before starting to install the CA MICS system.

To install CA MICS, review the list of considerations and requirements below, review the installation checklist presented at the end of this chapter, contact Technical Support at http://ca.com/support if you have any questions about the considerations or requirements in the list or tasks on the checklist, then complete each step and task on the checklist in the order specified.

For new installations, refer to the CA MICS Resource Management Best Practices Guide, available from CA Support Online, for a streamlined installation methodology.

Considerations and Requirements for Installing CA MICS

o CA MICS CLISTs require that the #, @, $, ?, :, ;, or " (double quote) NOT be used as control characters with the TSO background TMP.

o Not all release levels of the SAS program product in the field are supported for CA MICS database updating. Consult the CA MICS System Administrator Guide to determine the level of SAS to be used for the CA MICS update process. We recommend that a set of SAS libraries be created and used for CA MICS as discussed in Section 2.2.3.

o CA MICS assumes that a standard record format is used for all input data sources. Any local modifications to these record formats may require modification of CA MICS code.

o Special installation procedures will be required if multiple SYSIDs are assigned to the same CPU. Each CPU must have a unique SYSID.

o Frequent adjustment of the time-of-day clock on any of the CPUs for accounting or other reasons will result in predictably bad data.

o The jobs created by CA MICS JCL generation have certain CPU time limits specified for them. You may want to modify these limits as appropriate for your processor speed and data volumes.

- o The CA MICS CLISTs need a LOGON PROC that specifies DYNAMNBR=100 to ensure that all of the CA MICS data sets can be allocated.

- o SAS libraries must be available.

- o ISPF and ISPF/PDF libraries must be available.

SMF Issues

- o A special user modification is required if any accounting data is required from a user-generated SMF record extension.

- o SMF data from multiple CPUs that are loosely-coupled (JES3 or multi-access spool JES2) should be consolidated into a single unit database.

- o CA MICS data integration products process many different SMF record types. Please refer to prefix.MICS.GENLIB(cccGENIN) for SMF records used in each product.

We strongly recommended that the individual installing CA MICS follow the above instructions and discuss any problems or issues with Technical Support at http://ca.com/support before starting the items in the checklist.

3.2 Load Distribution Libraries

This section installs a new CA MICS system. The following sections describe the task to be completed:

1 - Define Catalog Indexes
2 - Ensure that CA Common Services is Operational
3 - Copy the Installation Files to DASD
4 - Install Distribution Tape
5 - Adapt CA MICS Complex to SAS Version

3.2.1 Define Catalog Indexes

Two data set name prefixes, a DASD prefix (sharedprefix) and a tape prefix (tsharedpref), are defined in sharedprefix.MICS.PARMS(JCLDEFC). The sharedprefix is required. The tsharedpref is optional and only used if you have licensed CA MICS Accounting and Chargeback, and if you are going to maintain a complex level audit file.

The data set name prefixes serve several purposes: to give unique names to the CA MICS data sets shared by all database units in the complex, and to provide unique and meaningful qualification for identification and control of the CA MICS shared data sets.

The sharedprefix and tsharedpref must be different from all of the other prefixes and tapeprefixes in the database complex. However, the sharedprefix and the other prefixes may have a common high-level index.

Unless standards at your site prevent it, this step should be used to ensure that the index established for the sharedprefix and tsharedpref point to a catalog other than the master catalog.

In selecting a high-level index for CA MICS data sets, be aware of the following:

1. Most sites have data set naming standards for production data sets. CA MICS should be considered as a production system, and the high-level index chosen accordingly.

2. If a high-level index of a TSO userid is selected, ensure that your site does not delete data sets that remain unreferenced over long periods of time. For example, the YEARS timespan database will not be referenced for output until the MONTHLY job is run.

If the highest level index of the sharedprefix is a TSO userid or an index that already exists, then the required catalog indexes and master catalog pointers are already built and no further action is required in this step.

Special Considerations:

1. No action is required if the high-level index has been defined already in the master catalog.

2. In a multi-CPU environment, it is essential that the high

level index be defined in a VSAM user catalog and not allowed to default to the master catalog. Data sets that are cataloged in the master catalog of one system will not be available from any other system's master catalog unless the data sets are explicitly cataloged on that system.

3.2.2 Ensure that CA Common Services is Operational

CA MICS requires CA LMP (License Management Program), one of the CA Common Services, before you can install or run CA MICS. You must ensure that all CA MICS keys have been added to CA LMP before attempting to load products from the distribution medium (tape or ESD PAX).

Contact CA Support at http://ca.com/support for assistance with CA LMP keys. General information on how to operate CA LMP can be found in the CA Common Services Administrator Guide.

These topics are discussed in detail in the following sections:

1 - LMP Key Syntax Definition
2 - Sample CA MICS LMP Key
3 - Description of LMP Key Certificate
4 - Running CA Common Services
5 - Report CA MICS LMP Key Status

3.2.2.1 LMP Key Syntax Definition

The parameter structure for member KEYS is presented below:

```
PROD(pp) DATE(ddmmmyy) CPU(ttt-mmmm/ssssss)
         LMPCODE(kkkkkkkkkkkkkkkk)
```

where:

pp — Required. The two-character product code. For any given CA LMP software solution, this code agrees with the product code already in use by the CAIRIM initialization parameters for earlier genlevels of CA MICS.

ddmmmyy — The CA LMP licensing agreement expiration date.

tttt-mmmm — Required. The CPU type and model (for example, 3090-600) on which CA LMP software solution is to run. If the CPU type and/or model require less than four characters, blank spaces are inserted for the unused characters.

ssssss — Required. The serial number of the CPU on which CA LMP software solution is to run.

kkkkkkkkkkkkkkkk — Required. The execution key needed to run the CA LMP software solution. This CA LMP execution key is provided on the Key Certificate shipped with each CA LMP software solution.

For a full description of the procedure for defining the CA LMP execution key to the CAIRIM parameters, see installation tasks in the CA Common Services "Getting Started" guide.

3.2.2.2 Sample CA MICS LMP Key

The following is a sample CA MICS LMP key:

```
prod(gi) date(16apr00) cpu(3090-600e/071584)
 lmpcode(19bcsz472zx7rzd7)
```

```
Note:  The actual definition of the LMP key occurs on a
       single statement without wrapping to a second line.
       The sample is only shown on two lines because of text
       formatting limitations.
```

3.2.2.3 Description of LMP Key Certificate

Examine the CA LMP Key Certificate you received with your CA MICS installation or maintenance tape. That certificate contains the following information:

```
Fields               Description
======               ============

Product Name         The trademarked or registered name of
                     CA MICS licensed for the designated
                     site and CPUs.

Product Code         A two-character code that corresponds
                     to a licensed CA MICS product. The
                     codes can be found in the chapter
                     called Running a Report for CA MICS LMP
                     Keys later in this guide.

Supplement           The reference number of your license
                     for the particular CA MICS, in the
                     format nnnnnn-nnn.  This format differs
                     slightly inside and outside North
                     America, and in some cases may not be
                     provided at all.

CPU ID               The code that identifies the specific
                     CPU for which installation is valid.

Execution Key        An encrypted code required by CA LMP
                     for CA MICS initialization.  During
                     installation, it is referred to as the
                     LMP code.

Expiration Date      The day (ddMONyy ie.,15JAN10) your
                     license for CA MICS expires.

Technical Contact    The name of the technical contact at
                     your site who is responsible for the
                     installation and maintenance of the
                     designated CA MICS.  This is the person
                     to whom CA addresses all CA LMP
                     correspondence.

MIS Director         The name of the director of MIS or the
                     person who performs that function for
                     your site. If the title but not the
                     individual's name is indicated on the
                     certificate, you should supply the
```

actual name when correcting and verifying the certificate.

CPU Location — The address of the building where the CPU is installed.

3.2.2.4 Running CA Common Services

After the CA MICS LMP key is defined, make sure CA Common Services is started, prior to executing any of the CA MICS installation or operational jobs or online processes. Attempts to execute without CA Common Services started will result in failures with an S122 ABEND.

3.2.2.5 Report CA MICS LMP Key Status

A report is available that lists all CA MICS products and provides the status of their LMP keys. The report shows those keys that have been installed and those keys that have not been installed.

The JCL and load module required for the report is found in the "tapeload" datasets: sharedprefix.MICS.TAPELOAD.CNTL and LOAD. These datasets are created when the instructions in Section 3.2.3 have been completed.

To run the report, do the following:

1. Edit the JCL in sharedprefix.TAPELOAD.CNTL(PSPLMPKV). Provide a Job statement, complete "sharedprefix" on the STEPLIB DD and submit the job.

2. Review the job's report found in the MICSLOG. Ensure the products appearing under the heading "CA MICS PRODUCT LICENSED ON THIS CPUID" are as you would expect.

The following is a list of CA MICS products and their associated MICS product code and LMP code.

MICS	LMP	
CODE	CODE	PRODUCT
ACT	JP	CA MICS Accounting and Chargeback Option
AST	JM	CA MICS CA ASTEX Option
AUM	JW	CA MICS Analyzer Option for MeasureWare
BAS	GI	CA MICS Base Platform
CAP	J5	CA MICS Capacity Planner Option
CIC	J7	CA MICS Analyzer Option for CICS
DB2	JL	CA MICS Analyzer Option for DB2
DEX	NH	CA MICS Analyzer Option for VAX/VMS
IDM	KS	CA MICS Analyzer Option for CA IDMS
IMS	KU	CA MICS Analyzer Option for IMS
LDE	KX	CA MICS IMS Log Data Extractor Option
MQR	KQ	CA MICS Q&R Workstation Mainframe Server
MQS	KZ	CA MICS Analyzer Option for MQSERIES
PER	LO	CA MICS Performance Manager Option
PWR	LD	CA MICS Analyzer Option for VSE/POWER
RMF	KV	CA MICS Hardware and SCP Analyzer
SMF	KN	CA MICS Batch and Operations Analyzer
SNT	MY	CA MICS Network Analyzer Option
SRL	MZ	CA MICS System Reliability Analyzer Option
STG	M3	CA MICS StorageMate Option

TDM	JS	CA MICS Tandem Option
TAP	JN	CA MICS Tape Analyzer Option (VTS & TLM)
TSO	M4	CA MICS Analyzer for TSO
VCA	J8	CA MICS Space Analyzer Option (VCA & HSM)
VCC	J9	CA MICS Space Collector Option
VMC	M5	CA MICS Analyzer Option for VM/CMS
VMT	KC	CA MICS Data Transfer Option for VM/CMS
WEB	KP	CA MICS WEB Analyzer Option

3.2.3 Copy the Installation Files to DASD

To start your installation, you must create and load two "tapeload" datasets. These datasets can be created from either a tape or an ESD PAX file. These topics are discussed in detail in the following sections:

1 - Create Tapeload Data Sets from ESD PAX
2 - Create Tapeload Data Sets from Tape

3.2.3.1 Create Tapeload Data Sets from ESD PAX

If you are installing from ESD PAX, you must download and unzip the ESD PAX file. Instructions to do this are available from the Download Center on CA Support Online.

During the unzip step, you changed "yourHLQ" to the high-level qualifier of the interim product files. That qualifier is used in the job below.

Create the JCL shown below. Complete the JOB card and parameters.

Note: The sharedprefix.MICS.TAPELOAD.LOAD data set cannot be a PDSE (partitioned data set extended) or we will be unable to write to it.

```
//jobname JOB ..........
//LOAD EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//#CNTL DD DISP=SHR,
//         DSN=yourHLQ.CAI.MICSESD.TAPELOAD.CNTL
//#LOAD DD DISP=SHR,
//         DSN=yourHLQ.CAI.MICSESD.TAPELOAD.LOAD
//CNTL  DD DSN=sharedprefix.MICS.TAPELOAD.CNTL,
//         DISP=(,CATLG,DELETE),UNIT=______,
//         VOL=SER=______,SPACE=(6160,(400,30,20)),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//LOAD  DD DSN=sharedprefix.MICS.TAPELOAD.LOAD,
//         DISP=(,CATLG,DELETE),UNIT=______,
//         VOL=SER=______,SPACE=(6160,(14,7,1)),
//         DCB=(RECFM=U,BLKSIZE=6160)
//SYSIN DD *
   COPY INDD=#CNTL,OUTDD=CNTL
   COPY INDD=#LOAD,OUTDD=LOAD
/*
```

where:

yourHLQ on the #CNTL and #LOAD statements specifies the high-level qualifier of the ESD PAX data sets and should be replaced with the "yourHLQ" used in the ESD PAX UNZIPJCL job.

sharedprefix on the CNTL and LOAD statements completes the name of the installation data sets and should be replaced with your CA MICS high-level qualifier.

UNIT and VOL=SER on the CNTL and LOAD statements specify the unit name of the direct access device and the volume serial number of the volume on which the installation libraries are to be allocated.

Submit the job and make sure that it completes with a condition code of zero.

Multiple members are loaded into the CNTL library; one member is loaded into the LOAD library. These data sets are used when installing your new CA MICS system, installing optional CA MICS products, or refreshing your PSP libraries.

3.2.3.2 Create Tapeload Data Sets from Tape

If you are installing CA MICS from a tape, create the JCL shown below. Complete the JOB card and parameters.

Note: The sharedprefix.MICS.TAPELOAD.LOAD data set cannot be a PDSE (partitioned data set extended) or we will be unable to write to it.

```
        //jobname JOB ..........
        //LOAD EXEC PGM=IEBCOPY
        //SYSPRINT DD SYSOUT=*
        //#CNTL DD DISP=SHR,DSN=CAI.TAPELOAD.CNTL,
        //         UNIT=______,VOL=SER=______,
        //         LABEL=(1,SL,EXPDT=______)
        //#LOAD DD DISP=SHR,DSN=CAI.TAPELOAD.LOAD,
        //         VOL=REF=*.#CNTL,
        //         LABEL=(2,SL,EXPDT=______)
        //CNTL  DD DSN=sharedprefix.MICS.TAPELOAD.CNTL,
        //         DISP=(,CATLG,DELETE),UNIT=______,
        //         VOL=SER=______,SPACE=(6160,(90,30,20)),
        //         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
        //LOAD  DD DSN=sharedprefix.MICS.TAPELOAD.LOAD,
        //         DISP=(,CATLG,DELETE),UNIT=______,
        //         VOL=SER=______,SPACE=(6160,(14,7,1)),
        //         DCB=(RECFM=U,BLKSIZE=6160)
        //SYSIN DD *
           COPY INDD=#CNTL,OUTDD=CNTL
           COPY INDD=#LOAD,OUTDD=LOAD
        /*
```

where:

UNIT, VOL=SER, and EXPDT on the #CNTL and #LOAD statements specify the unit name of the tape device on which the CA MICS distribution tape is to be mounted; the volume serial number of the tape; and, if you are using a tape management system, the tape expiration date.

sharedprefix on the CNTL and LOAD statements completes the name of the CA MICS installation data sets.

UNIT and VOL=SER on the CNTL and LOAD statements specify the unit name of the direct access device and the volume serial number of the volume on which the data sets are to be allocated.

Submit the job and make sure that it completes with a condition code of zero.

Multiple members are loaded into the CNTL library; one member is loaded into the LOAD library. These data sets are used when installing your new CA MICS system, installing optional CA MICS products, or refreshing your PSP libraries.

3.2.4 Install Distribution Libraries

After the "tapeload" libraries are created, you must populate your CA MICS complex libraries. The jobs to do this are contained in sharedprefix.MICS.TAPELOAD.CNTL. The job you select depends on the source of your distribution libraries. These jobs are discussed in detail in the following sections:

1 - Install Distribution Libraries from ESD PAX
2 - Install Distribution Libraries from Tape

3.2.4.1 Install Distribution Libraries from ESD PAX

After the "tapeload" libraries have been copied from the ESD PAX data sets, execute this step to load the remaining CA MICS complex libraries. Edit sharedprefix.MICS.TAPELOAD.CNTL and select the ESDMLD1 member. Review and complete the parameters on the EXEC statement. DO NOT change the parameters on the PROC statement.

The EXEC statement is:

```
  //LOAD EXEC LOAD
  //  RSIZE='4096K',            <-- REGION FOR ALL STEPS
  //  DSP=NEW,                  <-- NEW IF COMPLEX DOES NOT
  //*                                 EXIST, OLD IF COMPLEX
  //*                                 EXISTS
  //  LIBUNIT=,                 <-- LIB DASD UNIT REQUIRED
  //  LIBVOL=,                  <-- DO YOU NEED LIB DASD VOL?
  //  LIBSTORC=,                <-- OPTIONAL LIB STORCLAS=
  //  LIBDATAC=,                <-- OPTIONAL LIB DATACLAS=
  //  LIBMGMTC=,                <-- OPTIONAL LIB MGMTCLAS=
  //  LIBDSNT=,                 <-- OPT., ENTER LIBRARY FOR PDSE
  //  WKUNIT=SYSDA,             <-- IS WORK DASD UNIT OK?
  //  BLOCK=6160,               <-- LIBRARY BLKSIZE?
  //  VBLOCK=6160,              <-- BLKSIZE FOR VB LIBRARIES?
  //  YOURHLQ='______________', <-- REQUIRED --<<
  //  SPREFIX='______________', <-- REQUIRED --<<
  //  SMICS='MICS.',            <-- SHARED MICSLEVEL?
  //  LOADER=IEWLDRGO           <-YOUR SITE'S LOADER NAME
```

where:

RSIZE   is the amount of region storage for steps in this job.

DSP   is the disposition of the libraries being loaded: NEW if a new complex is being loaded; OLD if the complex already exists.

LIBUNIT, LIBVOL, LIBSTORC, LIBDATAC, LIBMGMTC, BLOCK, VBLOCK
are the unit, volume serial number, SMS storage class (STORCLAS=), SMS data class (DATACLAS=), SMS management class (MGMTCLAS=), and block size to be used for the production libraries. Note that the value for BLOCK must be greater than or equal to the block size for the ISPF panel and table libraries (ISPPLIB and ISPTLIB). In addition, VBLOCK is used for the documentation text libraries, which are in RECFM=VB format.

LIBDSNT can be either blank or LIBRARY. Specify LIBRARY to allocate new data sets as partitioned data set extended (PDSE) data sets. Any library ending with .LOAD will be allocated as a PDS.

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

YOURHLQ specifies the high level qualifier of the ESD PAX data sets. It must be replaced with the "yourHLQ" used in the ESD PAX UNZIPJCL job.

SPREFIX is the sharedprefix for the database complex. Note that this is specified with no trailing period (for example, DB1.SP rather than DB1.SP.).

SMICS is the "MICS" level of the standard CA MICS data set names. If you do not want "MICS" as part of your data set names, specify "SMICS=," to set it to null.

LOADER is the program name of the IBM loader at your site. This is typically IEWLDRGO, HEWLDRGO, or LOADER. If you are not sure, contact your system programmer.

After editing the JCL, submit the job and exit sharedprefix.MICS.TAPELOAD.CNTL.

```
*************************************************************
*                                                           *
* NOTE: YOU MUST EXIT TAPELOAD.CNTL AFTER YOU SUBMIT THE    *
*       JOB.                                                *
*                                                           *
*************************************************************
```

Note that the PARMCOPY step may end with a condition code of 4. This is a result of members being selected for a product for which your site is not licensed. This is an acceptable condition for the PARMCOPY step only. All other steps must end with a condition code of 0.

3.2.4.2 Install Distribution Libraries from Tape

After the "tapeload" libraries have been copied from the distribution tape, execute this step to load the remaining CA MICS complex libraries. Edit sharedprefix.MICS.TAPELOAD.CNTL and select the MICSLD1 member. Review and complete the parameters on the EXEC statement. DO NOT change the parameters on the PROC statement.

The EXEC statement is:

```
  //LOAD EXEC LOAD
  //  RSIZE='4096K',             <-- REGION FOR ALL STEPS
  //  DSP=NEW,                   <-- NEW IF COMPLEX DOES NOT
  //*                                 EXIST, OLD IF COMPLEX
  //*                                 EXISTS
  //  TUNIT='______',            <-- REQUIRED --<<
  //  TEXPDT=98000,              <-- IS TAPE EXPDT OK?
  //  TVOL=______,               <-- REQUIRED --<<
  //  LIBUNIT=,                  <-- LIB DASD UNIT REQUIRED
  //  LIBVOL=,                   <-- DO YOU NEED LIB DASD VOL?
  //  LIBSTORC=,                 <-- OPTIONAL LIB STORCLAS=
  //  LIBDATAC=,                 <-- OPTIONAL LIB DATACLAS=
  //  LIBMGMTC=,                 <-- OPTIONAL LIB MGMTCLAS=
  //  LIBDSNT=,                  <-- OPT., ENTER LIBRARY FOR PDSE
  //  WKUNIT=SYSDA,              <-- IS WORK DASD UNIT OK?
  //  BLOCK=6160,                <-- LIBRARY BLKSIZE?
  //  VBLOCK=6160,               <-- BLKSIZE FOR VB LIBRARIES?
  //  SPREFIX='______________', <-- REQUIRED --<<
  //  SMICS='MICS.',             <-- SHARED MICSLEVEL?
  //  LOADER=IEWLDRGO            <-YOUR SITE'S LOADER NAME
```

where:

RSIZE   is the amount of region storage for steps in this job.

DSP   is the disposition of the libraries being loaded: NEW if a new complex is being loaded; OLD if the complex already exists.

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

LIBUNIT, LIBVOL, LIBSTORC, LIBDATAC, LIBMGMTC, BLOCK, VBLOCK are the unit, volume serial number, SMS storage class (STORCLAS=), SMS data class (DATACLAS=), SMS management class (MGMTCLAS=), and block size to be

```
        used for the production libraries.  Note that the
        value for BLOCK must be greater than or equal to the
        block size for the ISPF panel and table libraries
        (ISPPLIB and ISPTLIB).  In addition, VBLOCK is used
        for the documentation text libraries, which are
        RECFM=VB format.

LIBDSNT can be either blank or LIBRARY.  Specify LIBRARY to
        allocate new data sets as partitioned data set
        extended (PDSE) data sets.  Any library ending with
        .LOAD will be allocated as a PDS.

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

SPREFIX is the sharedprefix for the database complex.  Note
        that this is specified with no trailing period (for
        example, DB1.SP rather than DB1.SP.).

SMICS   is the "MICS" level of the standard CA MICS data set
        names.  If you do not want "MICS" as part of your
        data set names, specify "SMICS=," to set it to null.

LOADER  is the program name of the IBM loader at your site.
        This is typically IEWLDRGO, HEWLDRGO, or LOADER.  If
        you are not sure, contact your system programmer.
```

After editing the JCL, submit the job and exit sharedprefix.MICS.TAPELOAD.CNTL.

```
*************************************************************
*                                                           *
* NOTE: YOU MUST EXIT TAPELOAD.CNTL AFTER YOU SUBMIT THE    *
*       JOB.                                                *
*                                                           *
*************************************************************
```

Note that the PARMCOPY step may end with a condition code of 4. This is a result of members being selected for a product for which your site is not licensed. This is an acceptable condition for the PARMCOPY step only. All other steps must end with a condition code of 0.

3.2.5 Adapt CA MICS Complex to SAS Version

CA MICS supports multiple SAS versions and is tailored to the SAS version you use. After successfully installing CA MICS from an installation tape (see section 3.2.4), edit and submit the job contained in sharedprefix.MICS.CNTL(MICSLS1).

Your sharedprefix.MICS.LOAD and sharedprefix.MICS.PARMS libraries contain members that are unique to a SAS version. The MICSLS1 job activates the appropriate member for the SAS version you use. In addition, selected data sets may need conversion from their distributed format to the form compatible with your SAS version.

Review and complete the parameters on the EXEC statement. DO NOT change the parameters on the PROC statement.

Note: The MICSLS1 job dynamically allocates the new sharedprefix.MICS.MCOLIB data set using the allocation parameters of the existing sharedprefix.MICS.MCOLIB data set. If you encounter problems with this allocation, for example, due to installation specific allocation and/or security requirements, you can uncomment the NEWSASX DD of the COPYMCO step and define it according to your requirements.

The EXEC statement is:

```
//CNVTMCO  EXEC CONVERT,SYSPARM=,
// WKUNIT='SYSDA',               <-- Required
// SHRPFX='sharedprefix.MICS',   <-- No Trailing Period
//*
// SASNAME='entry',              <-- SASHOST for V8
//*                                  SAS     for V9
// SASPFX='sas.prefix',          <-- No Trailing Period
// SASCFG1='config',             <-- CNTL(BATCH) IF SAS V8
//*                                  CNTL(BATW0) IF SAS 9.1.3
//*                                  CONFIG(BATCH) IF SAS 9.2
//*                                  For SAS 9.2 uncomment
//*                                  and specify next 3:
//*SASCFG2='CONFIG(COMMON)',         CONFIG(COMMON)
//*SASCFG3='CONFIG(ENW0)',           CONFIG(ENW0)
//*SASCFG4='CONFIG(SITE)',           CONFIG(SITE)
//*
//*AUTOLIB='fixed.block.AUTOLIB',<-- For 9.1.3 only,
//*                                  uncomment and
//*                                  point to FB version
//*                                  of SAS AUTOCALL library
//*
// LANG1='',                     <-- SAS V9 Language
```

```
//*                                    EN = English
//*                                    blank if V8
// LANG2=''                        <-- SAS V9 encoding value
//*                                    W0. = EBCDIC (1047)
//*                                    MUST HAVE TRAILING .
//*                                    blank if V8
```

where the parameters are as follows:

WKUNIT

The unit for the SYSUT3 and SYSUT4 work data sets needed by IEBCOPY.

SHRPFX

The sharedprefix for the database complex. Include the CA MICS level node as needed by your site. Note that this name is specified with no trailing period.

SASNAME

The library member name of the SAS program to be used, also called the entry point name. Normally this is SASHOST for version 8, and SAS for version 9.

SASPFX

The common data set name prefix for your SAS libraries. Note that this name is specified with no trailing period.

SASCFG1-4

The data set name qualifier and member name of your SAS configuration files. Specify only SASCFG1 for SAS versions 8 and 9.1.3. Normally the value is CNTL(BATCH) for SAS version 8 and CNTL(BATW0) for SAS version 9.1.3. For SAS version 9.2 or above, SASCFG1 is normally CONFIG(BATCH). To specify additional configuration files for SAS version 9.2 or above, uncomment SASCFG2, SASCFG3 and SASCFG4. The values are normally CONFIG(COMMON), CONFIG(ENW0), and CONFIG(SITE). Note that the full data set names are built by prefixing the value of SASPFX.

AUTOLIB

The SAS AUTOCALL library that is being converted to fixed block with SAS 9.1.3. It is not required for SAS Version 9.2 and higher.

LANG1

The language code required with SAS versions 9 and higher, for example, EN stands for English. The value is blank for SAS version 8.

LANG2

The encoding value required with SAS version 9 and higher. The value for EBCDIC is W0 and must have a trailing period (.). The value is blank for SAS version 8.

3.3 Define Database Complex

The CA MICS database may contain one or more database units. This section discusses the processes involved in establishing the database complex environment, prior to defining units within the complex. The complex-level JCL is generated, complex library work is completed, and the ISPF environment is defined.

The remainder of this section presents the following topics:

1 - SAS Libraries Used by CA MICS
2 - Generate Complex-level CA MICS Jobs, PROCs, and CLISTs
3 - Move Complex-level CA MICS PROCs to PROCLIB
4 - Move CA MICS FSP Screens and TABLES Files
5 - Define Currency Type Format and Informat
6 - Set Up the CA MICS ISPF Environment
7 - Generate Document Access (DOCAGEN1)

3.3.1 SAS Libraries Used by CA MICS

The following is a list of the SAS libraries required for CA MICS for SAS Versions 8 and 9. The configuration members for the various SAS versions are BATCH for Version 8 and BATW0 for Version 9.1.3. Beginning with SAS 9.2, there are four configuration members, BATCH, COMMON, ENW0, and SITE, usually located in the your.SAS.CONFIG data set.

- your.SAS.LIBRARY
- your.SAS.AUTOLIB
- your.SAS.SASHELP
- your.SAS.CNTL (for SAS 8 and 9.1.3)
- your.SAS.CONFIG (for SAS 9.2 and higher)
- your.SAS.SASMSG

3.3.2 Generate Complex-Level JCL Jobs, PROCs, CLISTs

Section 2.3.1.6, Database Complex JCL Definitions (JCLDEFC), describes the CA MICS parameters that control the process by which the JCL procedures, batch jobs, and TSO CLISTs required for complex-level functions are created.

Before you run JCLGEN0, you must provide some environmental parameter values in sharedprefix.MICS.PARMS, as described below. Note that any changes you make using the facilities discussed in Section 2.3.3.3, JCLGEN Parameters for Special Requirements, must also be complete before JCLGEN0 is submitted.

```
    JCL OPTIONS

        Member:  sharedprefix.MICS.PARMS(JCLDEFC)
        Refer to Section 2.3.1.6


    USER JOB CARD JCL

        Member:  sharedprefix.MICS.PARMS(USERJCL)
        Refer to Section 2.3.3.2.2
```

Review and complete the JOB card and the parameters on the EXEC statement of sharedprefix.MICS.CNTL(JCLGEN0). The EXEC statement is written as follows:

```
//STEP1  EXEC MICSDM,
// ISPTLB1='__________________', <-- REQUIRED
// ISPTLB2='__________________', <-- EQ ISPTLB1 IF NONE
// ISPMLB1='__________________', <-- REQUIRED
// ISPMLB2='__________________', <-- EQ.ISPMLB1 IF NONE
// WKUNIT='SYSDA',                <-- REQUIRED
// SHRPFX='sharedprefix.MICS',    <-- NO TRAILING PERIOD
//*
// SASNAME='entry',               <-- SASHOST for V8
//*                                   SAS     for V9
// SASPFX='sas.prefix',           <-- NO TRAILING PERIOD
// SASCFG1='config',              <-- CNTL(BATCH) IF SAS V8
//*                                   CNTL(BATW0) IF SAS 9.1.3
//*                                   CONFIG(BATCH) IF SAS 9.2
//*                                   For SAS 9.2 uncomment
//*                                   and specify next 3:
//*SASCFG2='CONFIG(COMMON)',          CONFIG(COMMON)
//*SASCFG3='CONFIG(ENW0)',            CONFIG(ENW0)
//*SASCFG4='CONFIG(SITE)',            CONFIG(SITE)
```

```
//*
//*AUTOLIB='fixed.block.AUTOLIB',<-- For 9.1.3 only
//*                                    uncomment and
//*                                    point to FB version
//*                                    of SAS AUTOCALL library
//*
// LANG1='',                       <-- SAS V9 LANGUAGE
//*                                    EN = English
//*                                    blank if V8
// LANG2=''                        <-- SAS V9 encoding value
//*                                    W0. = EBCDIC (1047)
//*                                    MUST HAVE TRAILING .
//*                                    blank if V8
//*
```

JCLGEN Parameter Sharing allows commonly used JCLGEN parameters to be specified in sharedprefix.MICS.PARMS(JCLDEFC), where they then become default parameters for prefix.MICS.PARMS(JCLDEF) at the unit level.

Use of this facility can greatly reduce the amount of coding in the JCLDEF and JCLDEFC members.

JCLGEN parameters eligible for sharing can be organized into the following groups:

```
SAS Data Set Definitions:
------------------------
SASNAME     - SAS program name
SASCONFIGx  - SAS configuration member
SASAUTOS    - SAS auto include library
SASMSG      - SAS message data set
SASLIB      - SAS executable library
SASVER      - SAS version
SASHELP     - SAS help library

ISPF Data Set Definitions:
-------------------------
ISPTLIBx    - ISPF table data set
ISPLLIBx    - ISPF load library
ISPMLIBx    - ISPF message data set
ISPPLIBx    - ISPF panel data set
ISPSLIBx    - ISPF skeleton data set

Miscellaneous Definitions:
-------------------------
ASSEMBLER   - IBM Assembler level
COLIBSPACE  - Size of the temporary MCOLIB
```

```
    COMPLEXPARMS- Allow/disallow JCLGEN parameter sharing
    DIRSTATS    - ISPF stats for generated members
    HFSROOT     - Name of the CA MICS HFS path
    JCLLIB      - Include JCLLIB statement with job card
    MICSDMDELIM - Delimiter character for MICSDM/MICSDU procs
    OBJACCESS   - Access mode for MCOLIB and MUOLIB
    OBJWAIT     - Wait time for MCOLIB and MUOLIB allocation
    PROCLIB     - Location of CA MICS PROCs
    SORTLIB     - SORT utility library
    SORTUTILITY - Installation sort utility
    SORTWKNO    - Number of sort work data set allocations
    STEPCPULIMIT- Limit CPU time for any step to 60 minutes
    STEPLIBx    - User-defined STEPLIBs
    SYSTEMMODEL - Name of model data for GDG creation
    USEROBJLIB  - User object definition library
```

JCLGEN Parameter Sharing is always active, but it must be enabled by specifying the following in sharedprefix.MICS.PARMS(JCLDEFC) or prefix.MICS.PARMS(JCLDEF):

```
COMPLEXPARMS YES
```

When specified in JCLDEFC, the value of COMPLEXPARMS is the default for all units. If specified in JCLDEF, COMPLEXPARMS overrides the COMPLEXPARMS action specified in JCLDEFC.

Section 5.5 provides an optional checklist that shows how to activate JCLGEN parameter sharing for all units in the CA MICS complex.

Notes:

o For the ISPF data set and the STEPLIB definitions, specifying the first definition (that is, ISPTLIB1) will automatically blank out the second definition (that is, ISPTLIB2). Therefore, you must always specify these definitions in numeric order.

o COMPLEXPARMS must be the very first statement in JCLDEFC and JCLDEF. If it is not, an error message is generated.

The ISPxLIBn definitions are used to supply the data set names of your site's IBM ISPF and ISPF/PDF product libraries. The TSO LISTALC command can be used to assist you in determining the correct names to specify for the ISPxLIBn definitions that follow:

ISPTLB1-2

The names of your IBM ISPF Program Product table libraries. ISPTLB1 is required. If ISPF and PDF are combined in one library at your site, you must specify ISPTLB2 with the same name in ISPTLB1.

ISPMLB1-2

The names of your IBM ISPF Program Product message libraries. ISPTLB1 is required. If ISPF and PDF are combined in one library at your site, you must specify ISPMLB2 with the same name in ISPMLB1.

WKUNIT

The unit for the SYSUT3 and SYSUT4 work data sets needed by IEBCOPY.

SHRPFX

The sharedprefix for the database complex. Include the CA MICS level node as needed by your site. Note that this name is specified with no trailing period.

SASNAME

The library member name of the SAS program to be used, also called the entry point name. Normally this is SASHOST for version 8, and SAS for version 9.

SASPFX

The common data set name prefix for your SAS libraries. Note that this name is specified with no trailing period

SASCFG1-4

The data set name qualifier and member name of your SAS configuration files. Specify only SASCFG1 for SAS versions 8 and 9.1.3. Normally the value is CNTL(BATCH) for SAS version 8 and CNTL(BATW0) for SAS version 9.1.3. For SAS version 9.2 or above, SASCFG1 is normally CONFIG(BATCH). To specify additional configuration files for SAS version 9.2 or above, uncomment SASCFG2, SASCFG3 and SASCFG4. The values are normally CONFIG(COMMON), CONFIG(ENW0), and CONFIG(SITE). Note that the full data set names are built by prefixing the value of SASPFX.

AUTOLIB

The SAS AUTOCALL library that is being converted to fixed block with SAS 9.1.3. It is not required for SAS Version 9.2 and higher.

LANG1

The language code required with SAS Versions 9 and higher, for example, EN stands for English. The value is blank for SAS Version 8.

LANG2

The encoding value required with SAS Version 9 and higher. The value for EBCDIC is W0 and must have a trailing period (.). The value is blank for Sas Version 8.

When you have completed editing the job, submit it as follows:

```
SUB 'sharedprefix.MICS.CNTL(JCLGEN0)'
```

There are other JCL generation requirements at the complex level, but they are for creating JCL specific to certain products such as Accounting and Chargeback (JCLGEN1). The guides for these products describe these complex-level generations.

3.3.3 Move Complex-Level CA MICS PROCs to PROCLIB

There are nine PROCs required for complex-level processing by CA MICS which must reside in the procedure library you named on the PROCLIB statement in sharedprefix.MICS.PARMS(JCLDEFC). The job to update the procedure library is contained in the member PROCJOB1 in the sharedprefix.MICS.CNTL library.

1. Review the procedures. If any need to be modified for your installation, you must use the process described in "JCLGEN Parameters for Special Requirements", Section 2.3.3.3.

   The PROCs added by PROCJOB1 are contained in the following members in sharedprefix.MICS.CNTL

MICSDM	- CA MICS Dialog Manager SAS Access
MICSGT	- CA MICS Component Generation and Test
MICSGU	- CA MICS Component Generation and Update
MICSASM	- CA MICS Assembly and Link
MICSCOM	- CA MICS Database Complex Access
MICSDOC	- CA MICS Document Print
MICSDOCH	- CA MICS DocWebVu
MICSRPTS	- CA MICS ISPF Dialog Manager Access
MICSSVR	- CA MICS Q&R Mainframe Server

2. Submit the job to add procedures to your procedure library:

   ```
   SUB 'sharedprefix.MICS.CNTL(PROCJOB1)'
   ```

3.3.4 Move CA MICS FSP Screens and Tables Files

3270 full-screen data entry facilities provided with CA MICS are based on the use of the SAS Full Screen Product (SAS/FSP).

The screen formats that define the layout and data entry rules for the 3270 panels used by SAS/FSP are SAS screen objects stored in the sharedprefix.MICS.SCREENS SAS library.

The Tables files and screen objects are both distributed in the sharedprefix.MICS.SASFLS data set. The following job should be executed to move them to the appropriate tables or screens data set.

```
SUB 'sharedprefix.MICS.CNTL(LDSASFLS)'
```

Note: SAS/FSP is NOT required for normal CA MICS database update and reporting. SAS/FSP data entry is used ONLY within an optional facility of the CA MICS Batch and Operations Analyzer.

3.3.5 Define Currency Type Format and Informat

CA MICS provides a MONEY format for writing monetary values in output reports. All data elements, such as the xxxCOST fields, defined in the CA MICS database have their SAS FORMAT defined as MONEY. The standard format displayed by the MONEY format is a display in U.S. dollars.

If you want to have the monetary values in your database formatted in a currency notation other than U.S. dollars, update 'sharedprefix.MICS.SOURCE(MONEY)'. The changes that are made take effect when BASPGEN or ALLPGEN is executed.

If you update the MONEY member, you should refer to 'sharedprefix.MICS.SOURCE(MONEYSMP)' for sample SAS formats. This member contains the same format as MONEY and also contains a format example which can be easily adapted to represent various European currency notations. You should NOT modify the MONEYSMP member as it is designed to receive maintenance from CA and must therefore be considered static.

SINGLE CURRENCY DISPLAY MODE

In addition to the MONEY format, the MONEY member also contains the MONEYR format, used to print rates in the Accounting Component. The MONEYB format, used to print budget amounts for certain invoice formats in the Accounting Component, is found in sharedprefix.MICS.SOURCE(MONEYB). The three formats differ as follows:

MONEY Used to display charges and certain other monetary amounts. Displays two decimal places.

MONEYB Used to print budget amounts for certain invoice formats. Displays no decimal places.

MONEYR Used to print rates. Displays four decimal places and does not support negative values.

Note: As delivered, each of these formats prints in US dollars. If you alter the MONEY format, you should alter MONEYR and MONEYB so that they reflect the same notation used for MONEY.

DUAL CURRENCY DISPLAY MODE

CA MICS provides euro and dual currency support in the CA MICS Accounting and Chargeback component. To learn more about activating euro support and about using the euro MONEYE formats, see Appendix C of the CA MICS Accounting and Chargeback User Guide.

3.3.6 Set Up the CA MICS ISPF Environment

This section discusses alternatives for setting up the ISPF environment at your installation to accommodate the CA MICS online facilities.

```
*********************************************************
*  Consult the System Requirements chapter of the       *
*  CA MICS System Administrator Guide for information   *
*  regarding supported levels of ISPF and recommended   *
*  maintenance levels.                                  *
*********************************************************
```

DYNAMIC ALLOCATION REQUIREMENTS

One of the critical options in a TSO LOGON procedure is the maximum number of concurrent dynamic allocations (DYNAMNBR) allowed to a TSO user. We recommend you allow at least 100 dynamic allocations. We selected 100 because there is no way to accurately predict the number of concurrent dynamic allocations that might exist.

For example, while the TSO user might be performing limited CA MICS functions on one side of the ISPF split screen, the other side of the split screen could be involved in an application that requires a large number of allocations. Also, dynamic allocation requirements for CA MICS applications fluctuate from one site to another and from one user to another. For example, a MICF inquiry requires a different number of allocations based on its complexity. If the inquiry references multiple CA MICS databases, then the number of allocations will increase as compared to an inquiry that references a single CA MICS database.

Experience will dictate the DYNAMNBR best suited for your site. There is a memory overhead associated with the DYNAMNBR, so your objective is to select a value that satisfies the needs of your users while not being excessively large.

CA MICS AND THE SASCP COMMAND PROCESSOR

CA MICS uses ISPF dialog manager services under the control of the SASCP command processor. The ISPF system programmer uses the ISPTCM table during the installation and customization of ISPF to define the characteristics of commands that operate under ISPF, such as whether or not a command processor can issue ISPF dialog manager services. Please coordinate with your ISPF system programmer about the following points to ensure that CA MICS can operate at your site.

The ISPTCM table is used to define characteristics about commands that will run under ISPF. The characteristics for the SASCP command processor can either be explicitly coded (ISPMTCM ENTRY ...) or assigned by default. To verify that your site's ISPTCM table allows the SASCP command processor to issue ISPF dialog manager services, review and, if necessary, execute the steps that follow.

To determine if the characteristics for the SASCP command processor have been explicitly coded, look in the ISPTCM table for the following line:

```
ISPMTCM ENTRY ENTNAME=SASCP,FLAG=xx...
```

- If the line is present and the optional FLAG=xx keyword is specified, check if it has the required "4" bit on in the first digit. Values that have the "4" bit on are "4x", "5x", "6x", "7x", "Cx", "Dx", "Ex", and "Fx". If it is not, correct the ISPTCM table and reassemble it.

- If the line is present and the optional FLAG=xx keyword is NOT specified, the keyword takes on the default value (FLAG=02) which does not enable the SASCP command processor to use ISPF dialog manager services. Correct the ISPTCM table as described in the paragraph above and reassemble it.

- If the SASCP command processor is not explicitly defined (there is not an ISPMTCM ENTRY ENTNAME=SASCP,FLAG=xx... line in the ISPTCM table), inspect the header entry in the ISPTCM table. The syntax of the header entry is:

  ```
  ISPMTCM HEADER DFFLAG=xx,...
  ```

  - If the optional DFFLAG=xx keyword is not specified, the keyword takes on the default value (DFFLAG=61) which enables the SASCP command processor to use ISPF dialog manager services. No further work is required.

  - If the optional DFFLAG=xx keyword is specified, check if it has the required "4" bit on in the first digit. Values that have the "4" bit on are "4x", "5x", "6x", "7x", "Cx", "Dx", "Ex", and "Fx". If it is not, you must either

    - change the header entry (ISPMTCM HEADER DFFLAG=xx,...)

to specify the optional DFFLAG=xx keyword, where xx is either "4x", "5x", "6x", "7x", "Cx", "Dx", "Ex", or "Fx", and reassemble the table

or

- code an explicit entry for the SASCP command processor (ISPMTCM ENTRY ENTNAME=SASCP,FLAG=xx...), set the optional FLAG=xx keyword to one of the values discussed above, and reassemble the table.

INSTALLING THE CA MICS WORKSTATION FACILITY (MWF)

Adding the CA MICS online facilities to your ISPF environment is a simple two step process. First, you add a CA MICS option to one of your installation's ISPF menus to invoke a CLIST. Second you create this CLIST which allocates CA MICS libraries, connects the libraries to the ISPF environment with the ISPF library definition (LIBDEF) service, and invokes the CA MICS Workstation Facility. With this approach, we minimize modifications to your installation's ISPF environment and we avoid conflicts between CA MICS online facilities and your other ISPF applications. CA MICS libraries are allocated ONLY when needed and ISPF searches CA MICS libraries ONLY when processing CA MICS.

Each installation's ISPF environment and requirements are unique. To meet your unique requirements, CA MICS provides two sample MWF CLISTs you can use as a starting point for creating your own unique MWF CLIST. The JCLGEN0 job generates the sample CLISTs, MWFSMPL1 and MWFSMPL2, into sharedprefix.MICS.CLIST.

The following sections illustrate how to add CA MICS online facilities to your ISPF environment. The first section is an overview of the ISPF LIBDEF service and related issues. The second section is a sample LIBDEF implementation. The third section describes an alternative implementation that may be useful for initial CA MICS testing.

The samples that follow assume familiarity with ISPF Dialog Manager Services. The samples shown in the guide use CA MICS sharedprefix libraries and ISPF default library names. Do not use these samples without verifying these library names; your installation might not use the same library names.
Note that the generated samples in sharedprefix.MICS.CLIST reflect the library name definitions you specified in sharedprefix.MICS.PARMS(JCLDEFC).

The following sections illustrate:

1 - ISPF LIBDEF Service Overview
2 - Sample CA MICS Online Facilities Implementation
3 - Alternative Online Facilities Implementation
4 - ISPF Preprocessed Panels

3.3.6.1 ISPF LIBDEF Service Overview

The ISPF library definition (LIBDEF) Service lets you logically modify the ISPF environment during the ISPF session. This makes it much easier for you to add new applications, such as the CA MICS Workstation Facility, to your installation's ISPF environment.

ISPF uses a series of libraries to access load modules (programs), panels, messages, and tables. These libraries are opened when ISPF is started, and remain open until the ISPF session ends. Thus, all application load modules, panels, messages, and tables must be included in the ISPF environment allocated before ISPF is started. However, the LIBDEF Service lets you logically concatenate additional libraries to the ISPF environment without leaving ISPF and without reallocating the environment.

The LIBDEF service lets you define an application-level set of libraries for ISPF panels, messages, tables, and programs. LIBDEF adds libraries to the ISPF environment to be searched BEFORE the original ISPF libraries. For example, if you use the LIBDEF Service for panels, the library that you indicate is logically concatenated ahead of the original ISPPLIB (panel library) data sets.

The ISPF program library, ISPLLIB, is a special case -- LIBDEF functions differently for the ISPLLIB allocation. In a production environment you will not normally have an ISPLLIB allocation because your ISPF and PDF programs are found in STEPLIB, the SYS1.LPALIB library, or the MVS link list. ISPLLIB is searched first and thus you can allocate a new release of ISPF to ISPLLIB for testing. But when you use the LIBDEF service for ISPLLIB, your LIBDEF ISPLLIB is NOT added to the ISPLLIB concatenation, but instead REPLACES the original ISPLLIB allocation. That means you lose access to the original ISPLLIB allocation. To retain program access you must include your original ISPLLIB data sets in your LIBDEF ISPLLIB allocation. For example, you would allocate both the CA MICS and new ISPF load libraries to your LIBDEF ISPLLIB. This technique is illustrated by sample 2 in Section 3.3.6.2.

3.3.6.2 Sample CA MICS Online Facilities Implementation

```
%-------------------  ISPF MASTER APPLICATION MENU  --------------------
%OPTION ====>_ZCMD                                                     +
%
%   1 +CA MICS     - CA MICS Workstation Facility
%   2 +.           - (Description for option 2)
%   3 +.           - (Description for option 3)
%   4 +.           - (Description for option 4)
%   5 +.           - (Description for option 5)
%   X +EXIT        - Terminate ISPF using list/log defaults

+Enter%END+command to terminate ISPF.
%
)INIT
  .HELP    = ISP00005     /* Help for this master menu             */
  &ZPRIM   = YES          /* This is a primary option menu         */
)PROC
  &ZSEL = TRANS( TRUNC (&ZCMD,'.')
     1,'CMD(%MWF)'
       /*******************************************************/
       /*                                                     */
       /* Add other applications here.                        */
       /*                                                     */
       /*******************************************************/
       /* Following shows how to code an invocation of the    */
       /* ISPF Program Development Facility, where "n" is     */
       /* the desired selection number:                       */
       /*                                                     */
       /*  n,'PANEL(ISR@PRIM) NEWAPPL(ISR)'                   */
       /*                                                     */
       /*******************************************************/
    ' ',' '
      X,'EXIT'
      *,'?' )
)END
```

CA MICS online facilities are accessed through a CLIST which is invoked from an ISPF menu. The CLIST allocates CA MICS ISPF software libraries with the TSO ALLOCATE command, invokes the ISPF library definition (LIBDEF) service, and starts the CA MICS Workstation Facility (MWF) with the ISPF SELECT service.

In this sample, we show the CA MICS option on your ISPF master application panel, ISP@MSTR (see the example above). The panel option executes a CLIST called MWF from your installation's ISPF dialog CLIST library. That is, the MWF clist is placed in a site-dependent library that is allocated to the SYSPROC DDNAME. The member name of this CLIST is MWF. If you use a different name, the selection coded on the menu should be changed accordingly.

In selecting the CLIST from the menu, DO NOT include a NEWAPPL keyword at this point. The NEWAPPL(MWF) keyword must be specified at the point when the CA MICS Workstation Facility is selected from the CLIST, using the ISPEXEC SELECT service. This method is required so that the MWFCMDS table is available from ISPTLIB at the point when the MWF application is started. This also means that the LIBDEF environment is established from the calling application, which should not itself be a LIBDEF environment.
Sample 1, CA MICS invocation with ISPF and PDF in SYS1.LPALIB

This sample assumes that the ISPF and PDF programs are located in STEPLIB, the SYS1.LPALIB library, or the MVS link list, and therefore are not accessed through the ISPLLIB concatenation. See the second sample if you use ISPLLIB to access ISPF and PDF. In this sample, "sasprefix.sas.library" refers to the SAS load library data set name you specified in the sharedprefix.MICS.PARMS(JCLDEFC) SASLIB parameter. See sharedprefix.MICS.CLIST(MWFSMPL1) for the generated version of this sample CLIST.

```
   ISPEXEC TBSTATS MWF$STAT STATUS3(STATUS3)
   IF &STATUS3 NE 1 THEN GOTO NOALLOC
   CONTROL NOMSG
   FREE FI(MICSLIB MWFPLIB MWFMLIB MWFSLIB MWFTLIB)
   CONTROL MSG
   ALLOC FI(MICSLIB) SHR DA('sharedprefix.MICS.LOAD' +
                            'sasprefix.sas.library')
   ALLOC FI(MWFPLIB) SHR DA('sharedprefix.MICS.ISPPLIB' +
                            'sharedprefix.MICS.ISPHLIB')
   ALLOC FI(MWFMLIB) SHR DA('sharedprefix.MICS.ISPMLIB')
   ALLOC FI(MWFSLIB) SHR DA('sharedprefix.MICS.ISPSLIB')
   ALLOC FI(MWFTLIB) SHR DA('sharedprefix.MICS.ISPTLIB')
   NOALLOC: +
   ISPEXEC LIBDEF ISPLLIB LIBRARY ID(MICSLIB)
   ISPEXEC LIBDEF ISPPLIB LIBRARY ID(MWFPLIB)
   ISPEXEC LIBDEF ISPMLIB LIBRARY ID(MWFMLIB)
   ISPEXEC LIBDEF ISPSLIB LIBRARY ID(MWFSLIB)
   ISPEXEC LIBDEF ISPTLIB LIBRARY ID(MWFTLIB)
   ISPEXEC SELECT PGM(MADA) PARM(NEXTFUNC=STARTUP) +
     NEWAPPL(MWF) PASSLIB
   ISPEXEC LIBDEF ISPLLIB ID(MICSLIB)
   ISPEXEC LIBDEF ISPPLIB ID(MWFPLIB)
   ISPEXEC LIBDEF ISPMLIB ID(MWFMLIB)
   ISPEXEC LIBDEF ISPSLIB ID(MWFSLIB)
   ISPEXEC LIBDEF ISPTLIB ID(MWFTLIB)
   ISPEXEC TBSTATS MWF$STAT STATUS3(STATUS3)
   IF &STATUS3 NE 1 THEN GOTO NOFREE
   FREE FI(MICSLIB MWFPLIB MWFMLIB MWFSLIB MWFTLIB)
   NOFREE: +
   EXIT CODE(0)
```

Sample 2, CA MICS invocation with ISPF and PDF in ISPLLIB:

This sample assumes that the ISPF and PDF programs are accessed through the ISPLLIB DDNAME concatenation. Because LIBDEF logically replaces the ISPLLIB allocation, you must include your ISPF and PDF program libraries in the CA MICS ISPLLIB concatenation as shown below. In this sample, "sasprefix.sas.library" refers to the SAS load library data set name you specified in the SASLIB parameter of sharedprefix.MICS.PARMS(JCLDEFC). See the MWFSMPL2 member of sharedprefix.MICS.CLIST for the generated version of this sample CLIST.

```
   ISPEXEC TBSTATS MWF$STAT STATUS3(STATUS3)
   IF &STATUS3 NE 1 THEN GOTO NOALLOC
   CONTROL NOMSG
   FREE FI(MICSLIB MWFPLIB MWFMLIB MWFSLIB MWFTLIB)
   CONTROL MSG
   ALLOC FI(MICSLIB) SHR DA('sharedprefix.MICS.LOAD' +
                           'sasprefix.sas.library' +
                          'ISP.SISPLOAD' +
                          'ISP.SISLPA')
   ALLOC FI(MWFPLIB) SHR DA('sharedprefix.MICS.ISPPLIB' +
                           'sharedprefix.MICS.ISPHLIB')
   ALLOC FI(MWFMLIB) SHR DA('sharedprefix.MICS.ISPMLIB')
   ALLOC FI(MWFSLIB) SHR DA('sharedprefix.MICS.ISPSLIB')
   ALLOC FI(MWFTLIB) SHR DA('sharedprefix.MICS.ISPTLIB')
   NOALLOC: +
   ISPEXEC LIBDEF ISPLLIB LIBRARY ID(MICSLIB)
   ISPEXEC LIBDEF ISPPLIB LIBRARY ID(MWFPLIB)
   ISPEXEC LIBDEF ISPMLIB LIBRARY ID(MWFMLIB)
   ISPEXEC LIBDEF ISPSLIB LIBRARY ID(MWFSLIB)
   ISPEXEC LIBDEF ISPTLIB LIBRARY ID(MWFTLIB)
   ISPEXEC SELECT PGM(MADA) PARM(NEXTFUNC=STARTUP) +
     NEWAPPL(MWF) PASSLIB
   ISPEXEC LIBDEF ISPLLIB ID(MICSLIB)
   ISPEXEC LIBDEF ISPPLIB ID(MWFPLIB)
   ISPEXEC LIBDEF ISPMLIB ID(MWFMLIB)
   ISPEXEC LIBDEF ISPSLIB ID(MWFSLIB)
```

```
ISPEXEC LIBDEF ISPTLIB ID(MWFTLIB)
ISPEXEC TBSTATS MWF$STAT STATUS3(STATUS3)
IF &STATUS3 NE 1 THEN GOTO NOFREE
FREE FI(MICSLIB MWFPLIB MWFMLIB MWFSLIB MWFTLIB)
NOFREE: +
EXIT CODE(0)
```

3.3.6.3 Alternative Online Facilities Implementation

Alternative 1 - Initial Testing

This alternative is intended for use during the initial phases of CA MICS and MWF implementation. This method allows access to MWF without changes to ISPF panels or to TSO CLIST libraries or any other facility that may require TSO system administrator intervention or authorization.

This alternative assumes that the MWF setup detailed in section 3.3.6.2 has been performed, and that the MWF option has not been added to the ISP@MSTR panel.

This option uses the ISPF TSO command panel. To use this option enter the following command in the ISPF primary command field:

If the MWF CLIST is in the SYSPROC concatenation:

```
    =6;%MWF
```

If the MWF CLIST is in a private library:

```
    =6;exec your.clist(MWF)
```

Alternative 2 - ISPF Command Table

This alternative allows MWF to be called as an ISPF command. Add the following command to the ISPCMDS commands table (ISPF option 3.9):

If the MWF CLIST is in the SYSPROC concatenation:

```
    MWF 3 SELECT CMD(%MWF)
```

If the MWF CLIST is in a private library:

```
    MWF 3 SELECT CMD(EXEC 'private.clist(MWF)')
```

NOTE: This alternative can be used in conjunction with the examples in section 3.3.6.2, or with the first alternative.

3.3.6.4 ISPF Preprocessed Panels

ISPF panels are written in an interpretive language that is specific to ISPF. Beginning with ISPF Version 2 Release 2, an option to preprocess, or compile, ISPF panels is available. This provides significant performance improvements for ISPF applications. The ISPF display service normally reads panels from an external library and converts them to an internal format before each display. Preprocessing the panels enhances performance by eliminating the interpretive process from normal display processing.

We recommend using preprocessed panels for the CA MICS Workstation Facility. The remainder of this section describes the steps necessary to accomplish the preprocessing of ISPF panels.

It is described in the IBM manual entitled "ISPF Dialog Developer's Guide", document number SC34-4821, under sections titled "Using Preprocessed Panels".

If you decide to use preprocessed panels for the CA MICS Workstation Facility, use the following steps for implementation:

1. Allocate a library with the same DCB attributes as the sharedprefix.MICS.ISPPLIB library which contains the panel definitions. For the purposes of this discussion, we call this new data set sharedprefix.MICS.PREPROC.ISPPLIB.

2. Invoke the preprocessed panel utility by entering ISPPREP in the primary command field on any ISPF panel.

3. Enter the sharedprefix.MICS.ISPPLIB as the input data set and the new sharedprefix.MICS.PREPROC.ISPPLIB as the output data set. Note that you should enter an asterisk (*) as the input data set member name to specify that all members should be participate. Press ENTER and run the utility.

4. There are some technical reasons that we will not expand on here that prevented some panels from being preprocessed in the previous step. We must copy those panels from the shareprefix.MICS.ISPPLIB to the sharedprefix.MICS.PREPROC.ISPPLIB in order to make a complete and usable library. (We can copy these panels to the new preprocessed library because ISPF

allows the coexistence of preprocessed and non-preprocessed panels in the same data set.)

Select the move/copy utility using PDF option 3.3. Select the copy options and enter the sharedprefix.MICS.ISPPLIB as the "from" data set and the new sharedprefix.MICS.PREPROC.ISPPLIB as the "to" data set. Enter the "from" member name as an asterisk (*) and the "replace like-named members" option as NO. This causes the utility to attempt to copy the entire library, but without replacing members that already exist.

The preprocessed library now has all the panels and can be used instead of the unprocessed (distributed) library.

5. Modify the ISPF allocations used to set up the CA MICS Workstation Facility so that the ISPPLIB allocation is pointing to sharedprefix.MICS.PREPROC.ISPPLIB instead of the actual sharedprefix.MICS.ISPPLIB data set.

   In other words, refer to the previous sections on the allocation of the ISPF environment and substitute the sharedprefix.MICS.PREPROC.ISPPLIB data set name wherever the original sharedprefix.MICS.ISPPLIB data set name is called for.

6. Retain the sharedprefix.MICS.ISPPLIB data set for maintenance purposes. DO NOT delete this data set; it is the library into which all CA MICS panel maintenance is applied. The preprocessing must be redone after applying any product change that changes CA MICS panels.

We do NOT recommend preprocessing the CA MICS tutorial panels (sharedprefix.MICS.ISPHLIB). The performance enhancement from preprocessing tutorial panels is minimal and, since tutorial panels are infrequently displayed, you are unlikely to gain sufficient benefit to justify the additional DASD space required for the preprocessed tutorial panel data set.

If you decide to preprocess the CA MICS tutorial panels, repeat steps 1 through 6 above for the sharedprefix.MICS.ISPHLIB data set. Remember that you must still retain the sharedprefix.MICS.ISPHLIB data set for maintenance purposes -- that is, DO NOT delete sharedprefix.MICS.ISPHLIB even if you preprocess the CA MICS tutorial panels.

3.3.7 Generate Document Access (DOCAGEN1)

CA MICS Document Access uses ISPF tables to provide the displays for Guide Selection, Text Selection, and the Subject Cross Reference selection panels. When installing a new CA MICS system or an additional CA MICS component, the following tables are generated to sharedprefix.MICS.ISPTLIB by the DOCAGEN1 job:

DOCLSTP4	-	Guide list table
DOCPxx	-	Titles table for guide xx
DOCXPa-z,@	-	Subject Cross Reference Phrase Tables
DOCXSa-z,@	-	Subject Cross Reference Section Tables
DOCXWV	-	Subject Cross Reference Keyword Table

These tables are also generated when guides are added or updated in the documentation libraries by other types of product updates. In such cases, the product update documentation includes this step, if necessary.

To generate all the tables for Document Access, perform the following:

```
  Submit 'sharedprefix.MICS.CNTL(DOCAGEN1)'
```

After the job completes, check the output to ensure that the job completed successfully. Review the MICSLOG output file for any error messages. DOCAGEN1 also sends these messages to the ISPF LOG output file if the ISPF LOG is allowed in batch at your installation. If the following message appears:

```
  Docpick Incomplete - Docpick Program Issued Return Code:  x
```

check the return code value of x. If the value is 1 or 2, this is merely an information message (not an error), and the generation should have continued to completion. A higher value indicates that a problem exists, and should be accompanied by a longer diagnostic message to help determine the cause of the problem.

DOCAGEN and DOCAGEN1 Jobs

Two generation jobs are provided for Document Access. The DOCAGEN1 job performs the complete generation process, which includes the Subject Cross Reference tables generation. The DOCAGEN job generates only the Guide and Titles tables, and is used for product updates that do not require regeneration of the Subject Cross Reference tables.

Other tables created by Document Access include:

DOCCAT - Document Management Print Request Table, which is created automatically the first time that any user requests the Document Print option from Document Access.

DOCBMKP4 - User-level Bookmark Table in MICS.USER.TABLES, which is created automatically the first time that each user invokes Document Access.

Document Access Installation Notes:

If default values set in Document Access need to be changed globally for all Document Access users, the following modules should be reviewed and changed accordingly:

o sharedprefix.MICS.ISPSLIB(DOCAG09S)

- If your CA MICS cataloged procedures reside in a user-specified procedure library, there are two ways to specify that library in generated JCL.

  To use the PROCLIB that may have been specified in sharedprefix.MICS.PARMS(JCLDEFC), uncomment the)SEL, //DOCAG09S, and)ENDSEL statements by removing the)CM characters. (The PROCLIB definition statement in JCLDEFC is optional.)

  To ensure that a specific PROCLIB is always used, uncomment only the //DOCAG09S statement and code it to specify that PROCLIB.

- If you do not have lowercase print capability and want all Document Print jobs to translate output to uppercase, uncomment the)SEL DOCACASE statement by removing the)CM characters.

- If you want to print something other than 60 lines per page, set xx to your desired page size. Then uncomment the)SEL DOCASPAR statement by removing the)CM characters.

To illustrate, if all of the above changes were made, the DOCAG09S member would look similar to this:

```
)CM
)SET DOCACASE = 'UCASE'
)SET DOCASPAR = 'NLINES=66'
)SEL @DOCAMPRC NE @Z
//DOCAG09S JCLLIB  ORDER=(@DOCAMPRC)
)ENDSEL
```

o sharedprefix.MICS.ISPPLIB(DOCAI00P)

Sample options are described in the panel module as)PROC section comments. They include options such as overriding the default SYSOUT print destination for text printing, allowing for unusual high-level name qualifiers for the user's browse data sets, and setting all text output to uppercase.

Any one of the comment lines can be uncommented by removing the initial /* and customized according to your needs.

For example, this statement would change the number of lines per page to 66:

```
&DOCASPAR = 'NLINES=66'     /* PRINT LINES PER PAGE */
```

3.4 Define and Generate Components

This phase of the installation defines what components are to be installed in the database complex, the options with which they should be generated, and the method of component generation. The topics covered in the sections that follow are:

1 - Select Components to be Generated
2 - Component Definition Statements (cccGENIN)
3 - Generate the Component (cccCGEN)

3.4.1 Generate the Component (cccCGEN)

This phase of the installation defines what components are to be installed in the database complex.

Define the components, both standard and user-written, that should be generated. As noted in the installation checklists later in this chapter, not all CA MICS products require generation.

If this is the first time that a CA MICS complex is being installed, all components must eventually be generated, but they do not all need to be generated during initial complex activation. Refer to the CA MICS Resource Management Best Practices Guide, available from CA Support Online, for a streamlined installation methodology.

If you are simply adding additional components to an existing complex, or have changed specific component generation statements, only those components affected need to be generated.

Note that component generation takes place on the level of the database complex. This means that each component that will appear in any database unit must be generated using the appropriate cccCGEN job.

3.4.2 Component Definition Statements (cccGENIN)

A generation definition statement member is provided for each product in the sharedprefix.MICS.GENLIB library under the member name cccGENIN, where ccc is the three-character component identifier. The syntax and definitions for the generation definition statements are contained in section 4.2 of the System Modification Guide.

The Database Tailoring facility enables you to tailor each product's database files and each file's contents. This facility is described in the System Modification Guide.

Before the component generation jobs can be run, the complex definitions for ESSENTIAL and DERIVED data element options must be specified, and the parameters generated. The generations take place as a result of coding the sharedprefix.MICS.PARMS(CPLXDEF) parameters and submitting sharedprefix.MICS.CNTL(CPLXGEN). See section 2.3.1.8, Complex Option Definitions, for more information.

We emphatically recommended that NO DATABASE TAILORING be performed the first time that a product is installed.

Please contact CA MICS Product Support to discuss the use of the Database Tailoring facility.

Review the cccGENIN file to ensure the following:

a) There must be a GEN GENFILES statement.

b) Some CA MICS products use an OPTION statement to indicate whether certain groups of data elements (called clusters) are to be kept in the database. Check the default definition for these OPTION statements and change them as appropriate to meet your needs.

   Note: The cccGENIN files contain upper and lower case characters. OPTION statement keywords should be entered in upper case.

c) Some CA MICS products deliver files turned off (data is not collected until the site turns the file on). Refer to the product documentation for information on turning on these files.

3.4.3 Generate Components

The cccCGEN jobs (where ccc is a component's three-letter abbreviation) use the CA MICS Component Generator to produce most of the code which comprises a component. The cccCGEN jobs generate members and entries in the following set of libraries:

```
sharedprefix.MICS.SOURCE
sharedprefix.MICS.USER.SOURCE
sharedprefix.MICS.INCLLIB
sharedprefix.MICS.GENLIB
sharedprefix.MICS.PROTOLIB
sharedprefix.MICS.INC.TEXT
```

The cccCGEN jobs are submitted as follows:

```
sub 'sharedprefix.MICS.CNTL(cccCGEN)'
```

The component generation process is described in the System Modification Guide.

3.5 Generate a Database Unit

A database unit is one physical database within the database complex. In updating and reporting for this physical database, the shared libraries, defined at the database complex-level, are used, as well as a unique set of libraries associated to the database unit.

This phase of the installation ensures that the proper catalog definitions are made, the database unit libraries and database files are allocated, the JCL/CLISTs are generated, the installation PROCLIB is updated, and PARMS have been defined and generated.

The steps in defining, generating, and allocating a database unit are:

1 - Define Catalog Indexes for Prefix and Tapeprefix
2 - Build Database Unit Libraries
3 - Generate JCL/CLIST Procedures
4 - Move CA MICS PROCs to Installation PROCLIB
5 - Generation of Parameter (PARMS) Defined Options
6 - Generate and Allocate the Database
7 - Finalize a Database Unit

3.5.1 Define Catalog Indexes for Prefix and Tapeprefix

A database unit requires two data set name prefixes, a DASD prefix (prefix) and a tape prefix (tapeprefix), which are defined in 'prefix.MICS.PARMS(JCLDEF)', described in section 2.3.3.2.1, "JCL Option Definition (JCLDEF)". These data set name prefixes serve several purposes:

1. They are used to give unique names to the data sets associated with a specific database unit.

2. The high level qualifier of the prefixes may serve as a catalog index for the data sets. This index may provide a pointer to either an OS CVOL or a VSAM user catalog.

3. The prefixes provide meaningful qualification for identification and control of the CA MICS shared data sets.

The prefix and tapeprefix for a database unit must be different from the sharedprefix and from the prefixes and tapeprefixes for all other database units in the database complex. However, the prefix and tapeprefix for a unit may be the same. Also, the sharedprefix and the other prefixes may have a common high level index.

Unless standards at your site prevent it, this step should be used to ensure that the indexes established for the tapeprefix and prefix point to a catalog other than the master catalog. We strongly recommend that the master catalog not be used for cataloging CA MICS data sets, especially in sites where there may be more than one OS/390 system on which CA MICS may be used.

In selecting a high level index for CA MICS data sets, be aware of the following:

1. Most sites have data set naming standards for production data sets. CA MICS should be considered as a production system, and the high level index chosen accordingly.

2. If a high level index of a TSO userid is selected, ensure that your site does not delete data sets that remain unreferenced over long periods of time. For example, the YEARS timespan database will not be referenced for output until the MONTHLY job is run--once a month.

If the high level index of the prefix is a TSO userid or an index that already exists, then the required catalog indexes and master catalog pointers are already built and no further action is required in this step. If this is not the case, then the following steps must be performed.

The choice of Access Method Services (AMS) commands used to define this high level index depends on whether an OS CVOL or VSAM catalog is used, and if VSAM, whether a master or user catalog is to be used.

AMS commands for an OS CVOL index definition (assumes CVOL already exists):

```
    DEFINE ALIAS(NAME('---name---')  -
         RELATE('SYSCTLG.Vxxxxxx'))
```

AMS commands for a VSAM Master Catalog index definition:

No action required, but not recommended.

AMS commands for a VSAM User Catalog index definition (assumes user catalog already exists):

```
    DEFINE ALIAS(NAME('---name---')  -
         RELATE('user-catalog-name'))
```

The xxxxxx field in the OS CVOL example is the name of the volume on which the OS CVOL resides. The '---name---' field must be the high level index of the prefix. This index may be common to all of sharedprefix, prefix(es) and tapeprefix(es).

Special Considerations:

1. No action is required if the high level index has been defined already in the master catalog.

2. In a multi-CPU environment, it is essential that the high level index be defined in an OS CVOL or VSAM user catalog and not allowed to default to the master catalog. Data sets that are cataloged in the master catalog of one system will not be available from any other system's master catalog unless the data sets are explicitly cataloged on that system. The DEFINE ALIAS command should be executed against every system's master catalog where you wish to have access to CA MICS data sets.

For example, if the prefix was DB2.PRE and the tapeprefix was

DB2T.PRE, the following definitions might be made:

If an OS CVOL catalog residing on volume TS0999 was to be used:

```
            DEFINE ALIAS(NAME('DB2')  -
                 REL('SYSCTLG.VTS0999'))
            DEFINE ALIAS(NAME('DB2T')  -
                 REL('SYSCTLG.VTS0999'))
```

If a VSAM Master Catalog index was to be used:

No action required, but not recommended.

If a VSAM User Catalog named SYS.USERCAT was to be used:

```
            DEFINE ALIAS(NAME('DB2')  -
                 REL('SYS.USERCAT'))
            DEFINE ALIAS(NAME('DB2T')  -
                 REL('SYS.USERCAT'))
```

3.5.2 Build Database Unit Libraries

Building the database unit libraries involves two stages.

- o First, the COPYLIBS job allocates and initializes the new database unit libraries by copying initial program, control, and data members from the database complex shared libraries.

- o Then, after the CA MICS unit installation jobs have been generated, the COPYLIBU job populates additional unit library program, control, and data members based on the products being installed in the unit.

These two jobs are discussed in the following sections:

1 - Create Database Unit Libraries (COPYLIBS)
2 - Populate Database Unit Libraries (COPYLIBU)

3.5.2.1 Create Database Unit Libraries (COPYLIBS)

The job in sharedprefix.MICS.CNTL(COPYLIBS) is used to allocate and initialize the database unit libraries by copying initial program, control, and data members from the database complex shared libraries. This job should be edited to review and complete the parameters on the EXEC statement. DO NOT change the parameters on the PROC statement. The EXEC statement follows:

```
//LOAD  EXEC LOAD,DSP=NEW,      <-- NEW IF UNIT DOESN'T EXIST
//*                                 OLD IF UNIT EXISTS
//           DUNIT=MICSDASD,    <-- IS DASD UNIT OK
//           DVOL=,             <-- DO YOU NEED DASD VOL
//           BLOCK=6160,        <-- LIBRARY BLKSIZE
//           PREFIX='_______',  <-- REQUIRED --<<
//           PMICS='MICS.'      <-- PREFIX MICSLEVEL
```

where the parameters are:

DUNIT, DVOL, BLOCK

The unit, volume serial number, and blocksize to be used for the production libraries.

PREFIX

The prefix for the database unit. Note that this is specified with no trailing period (e.g., DB1.SP rather than DB1.SP.).

PMICS

The "MICS" level of the standard CA MICS data set names. If you do not want this as part of your data set names, you may specify PMICS= ,to set it to null. Note that a non-null entry for PMICS must end with a period. Also, if you plan to not use "MICS" as part of your data set names, refer to the discussion of NOMICSLEVEL in section 2.3.1.6.1.

We recommend that the libraries be allocated to a single DASD volume and that the volume be maintained online. By keeping the libraries on the same DASD volume, you reduce the risk of a DASD volume restore back-leveling one or more unit libraries while leaving other unit libraries unchanged.

Note, the database unit libraries are used (and must be online) during database update, production or ad hoc reporting, for MICF or Q&R inquiry, etc.

After editing the job to create the unit database libraries, submit it:

```
             SUB 'sharedprefix.MICS.CNTL(COPYLIBS)'
```

IMPORTANT: The libraries created by this job are not copied from their similarly-named complex level libraries. This means that changes made to the complex level libraries are not carried forward to the database unit libraries. You must NOT modify the sharedprefix.MICS.HOLD.PARMS library, as this library is designed to receive maintenance from CA and must therefore be considered static.

Distribution libraries that are permanently allocated to the database unit are:

```
     prefix.MICS.CNTL
     prefix.MICS.MUOLIB
     prefix.MICS.PARMS
     prefix.MICS.USER.SOURCE
```

These libraries use approximately 10 Megabytes of DASD space if all CA MICS products are installed in the unit.

Data sets that are permanently allocated to the database unit are:

```
     prefix.MICS.CHECKPT.DATA
     prefix.MICS.DAYS
     prefix.MICS.DETAIL
     prefix.MICS.IMSSUS1 (if IMS Analyzer is licensed)
     prefix.MICS.IMSSUS2 (if IMS Analyzer is licensed)
     prefix.MICS.MONTHS
     prefix.MICS.RESTART.CNTL
     prefix.MICS.WEEKS
     prefix.MICS.YEARS
```

COPYLIBS copies initial program, control, and data members from the following CA MICS Platform libraries into the libraries it creates for the new unit:

```
     sharedprefix.MICS.HOLD.PARMS
     sharedprefix.MICS.HOLD.USER.SOURCE
```

3.5.2.2 Populate Database Unit Libraries (COPYLIBU)

Once the JCLDEF member is coded (see Section 3.5.3), the products in the unit are defined. The JCLGENA and JCLGENB jobs generate another installation job, COPYLIBU, which copies more parameter members into the unit libraries previously listed, based on the products installed in the unit.

Note: The JCLGENA job is executed exactly one time, when you create a new unit database. The JCLGENB job is executed when ever you need to recreate the installation jobs generated by JCLGENA and JCLGENB (e.g., to generate an updated COPYLIBU job).

Another function of COPYLIBU is to generate the database modeling parameter library member DBMODEL. When constructed by COPYLIBU, this member contains information for active information areas and files for only the products installed in the unit database.

To finish populating the unit libraries, submit the job:

```
    SUB 'prefix.MICS.CNTL(COPYLIBU)'
```

3.5.3 Generate JCL, CLISTs, and Procedures

A process named JCLGEN is used to create the batch job streams, JCL procedures, MACROS, and TSO CLISTs required by the batch, online, and CA MICS maintenance facilities. This process is executed several times during the installation process. Each JCLGEN execution generates a different group of output members. Execution of JCLGEN is discussed in the following sections:

1 - Defining Parameters for JCL Generation
2 - Executing JCLGENA
3 - Executing JCLGENC Through JCLGENG
4 - Selective JCL Generation Using JCLGENU

3.5.3.1 Define Parameters for JCL Generation

As mentioned above, the JCLGEN processes create the batch jobs, JCL procedures, MACROs, and TSO CLISTs required by the batch, online, and CA MICS maintenance facilities. The process is executed several times during the installation process. Each JCLGEN execution generates a different group of members. Before the JCLGEN jobs are executed, you must complete the parameter members described below.

Note: Any changes that you make using the facilities discussed in Section 2.3.3.3, "JCLGEN Parameters for Special Requirements," must also be completed before the JCLGEN is performed.

For JCLGENA through JCLGENG, and JCLGENU, the following parameter members must be completed:

```
    JCL OPTIONS:
        Member:  'prefix.MICS.PARMS(JCLDEF)'
        See section 2.3.3.2.1

    USER JOB CARD JCL:
        Member:  'prefix.MICS.PARMS(USERJCL)'
        See section 2.3.3.2.2

    SMF LOG INPUT JCL DD STATEMENTS:  (if required)
        Member:  'prefix.MICS.PARMS(INPUTRDR)'
        See section 2.3.3.2.3

    DATABASE COMPONENT INPUT JCL DD STATEMENTS (only if
    required):
        Member:  'prefix.MICS.PARMS(INPUTccc),' where ccc is
        the component identifier.  See section 2.3.3.2, its
        subsections, and to individual component guides.
```

3.5.3.2 Executing JCLGENA

JCLGENA differs from other JCLGEN jobs in that it is generated only once, by JCLGEN0, and copied into prefix.MICS.CNTL. Therefore, it must be edited to fill in the database unit's prefix value and distribution library data set names. All of the other JCLGEN jobs are generated within the unit so no further editing of their JCL need be done.

Before executing JCLGENA, complete the EXEC statement by filling in the CNTL, PARMS, and USOURCE parameters, and complete the data set name (DSN=) parameter on the //ULOCKIT DD statement.

Once JCLGENA has been edited, submit the job as follows:

```
    SUB 'prefix.MICS.CNTL(JCLGENA)'
```

3.5.3.3 Executing JCLGENC Through JCLGENG

After JCLGENA has executed correctly, you should check the generated JCL output. This can be done by examining the member prefix.MICS.CNTL(JCLGENC). If the JCL and JES parameters are not acceptable for your installation, then make the needed changes to prefix.MICS.PARMS members JCLDEF or USERJCL and rerun JCLGENA. When the JCL is acceptable, the other JCLGEN jobs are ready to run.

For example, to generate the unit's operational jobs, execute JCLGEND. This can be done with the following SUBMIT command.

```
     SUB 'prefix.MICS.CNTL(JCLGEND)'
```

The other JCLGEN jobs are submitted similarly and generate the following groups of jobs:

JCLGENA - Database Unit Installation Jobs - Part 1

> Note: JCLGENA is not generated. The COPYLIBS job copies JCLGENA from sharedprefix.MICS.CNTL(JCLGENA) to prefix.MICS.CNTL(JCLGENA) when you first create a new unit database. You generally execute JCLGENA once to initialize the new unit database. Once the unit database is defined, you then execute the JCLGENB job to recreate the "Database Unit Installation Jobs - Part 1."

JCLGENB - Database Unit Installation Jobs - Part 1

> Note: JCLGENB generates the same set of jobs generated by the JCLGENA job. However, the JCLGENA job is intended to be executed only one time, when you first define a new unit database. The JCLGENB job is executed when ever you need to recreate the "Database Unit Installation Jobs - Part 1."

JCLGENC - Database Unit Installation Jobs - Part 2
JCLGEND - Database Unit Operational Jobs
JCLGENE - Jobs Affected by a DB Split After Installation
JCLGENF - SPECIAL Database Unit Installation Jobs
JCLGENG - SPECIAL Database Unit Operational Jobs
JCLGENU - Regenerate Unit Jobs from a User Supplied List

Note: JCLGENE, JCLGENF, JCLGENG, and JCLGENU are not used during a regular CA MICS installation.

3.5.3.4 Selective JCL Generation Using JCLGENU

JCLGENU has no specific set of jobs which it generates. Rather, it is provided for you to use for selective generation or regeneration of any CA MICS prototype member.

The JCLGEN process contains a SAS macro variable, GENLIST, that names a member containing a list of the names of the JCL job streams, JCL procedures, MACROs, and/or TSO CLISTs you are trying to create. The GENLIST names a member in the library pointed to by the PARMS DD statement. For JCLGENU, the list of members to be generated is initially empty. You must enter your own list before running the JCLGENU job.

The names in the GENLIST member can be specified free form, separated by blanks or commas. An asterisk indicates the beginning of a comment. A line of blanks is allowed. Names can appear in position 1-72 but may not be continued between lines.

For example, prefix.MICS.PARMS(JCLGENU) might be used to specify that jobs DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, and DOCUMENT are to be regenerated by JCLGEN. The member would contain the following:

```
* LIST OF MEMBERS TO BE RE-GENERATED BY JCLGEN
DAILY WEEKLY MONTHLY
YEARLY
BACKUP
```

To run JCLGENU for a selective generation, SUBMIT the JCLGENU job.

```
SUB 'prefix.MICS.CNTL(JCLGENU)'
```

The JCLGEN process will use ISPF enqueue mechanisms to ensure that the members being written to are not being used. If it encounters a member being used, the JCLGEN process will terminate abnormally.

If your installation has a security system, such as RACF, installed, the user submitting the job will require authorization to run the TSO CLISTs in background (TMP background process). Contact your installation's security administrator to determine any local restrictions.

3.5.4 Move CA MICS PROCs to Installation PROCLIB

There are three sets of cataloged JCL procedures (PROCs) needed for batch processing by CA MICS. These must reside in one of your site's PROCLIBs.

The job to update the CA MICS complex-level procedures in the library you named on the PROCLIB statement in sharedprefix.MICS.PARMS(JCLDEFC) is generated in sharedprefix.MICS.CNTL(PROCJOB1).

The jobs to update the CA MICS unit-level procedures in the library you named on the PROCLIB statement in prefix.MICS.PARMS(JCLDEF) are generated in members PROCJOB2 and PROCJOB3 in the prefix.MICS.CNTL library.

The CA MICS procedures that are independent of the CA MICS database or that access the entire complex are copied by PROCJOB1. The procedures copied by PROCJOB2 do not access the database while the procedures copied by PROCJOB3 do provide database access. Both are dependent on code generated at the unit database level.

The PROCJOB1, PROCJOB2, and PROCJOB3 jobs execute the IEBCOPY utility to add the CA MICS cataloged procedures to a partitioned data set. Note that some sites may have special security procedures or other JCL requirements for authorization to update a PROCLIB. If this is the case at your site, review the PROCJOB1, PROCJOB2, and PROCJOB3 JCL that CA MICS generates before submitting each job. If any special JCL modifications are needed, make them before submitting the update job.

When executing any of the PROCJOBs you should perform the following steps.

1. Review the procedures that will be copied by

   'sharedprefix.MICS.CNTL(PROCJOB1)'

MICSDM	- CA MICS Dialog Manager SAS Access
MICSGT	- CA MICS Component Generation and Test
MICSGU	- CA MICS Component Generation and Update
MICSASM	- CA MICS Assembly and Link
MICSCOM	- CA MICS Database Complex Access
MICSDOC	- CA MICS Document Print
MICSDOCH	- CA MICS DocWebVu
MICSRPTS	- CA MICS ISPF Dialog Manager Access

```
'prefix.MICS.CNTL(PROCJOB2)'
  MICSNDBx - CA MICS Database Unit No Access
  MICSDUx  - CA MICS Database Unit Dialog Manager SAS
                  Access

'prefix.MICS.CNTL(PROCJOB3)'
  MICSSHRx - CA MICS Database Unit Shared Access
  MICSDBx  - CA MICS Database Unit Exclusive Access
```

If any PROCs need to be modified for your site, follow the guidance in JCLGEN Parameters for Special Requirements, Section 2.3.3.3.

2. Submit the job for execution.

```
  SUB 'sharedprefix.MICS.CNTL(PROCJOB1)'
  SUB 'prefix.MICS.CNTL(PROCJOB2)'
  SUB 'prefix.MICS.CNTL(PROCJOB3)'
```

3.5.5 Generate Parameter-Defined Options (cccPGEN)

The prefix.MICS.PARMS parameters that control CA MICS Database update reporting and online inquiry facilities are implemented via generated SAS MACROs and SAS FORMATs. The use of SAS MACROs (pre-defined sets of SAS code that are included in the source programs at their point of invocation) and SAS FORMATs enables you to control CA MICS operation without changing the source code itself.

The component generation process is performed by the component generation job ALLPGEN. ALLPGEN performs the component generation process for CA MICS Base (BASPGEN) plus the component generation processes for each CA MICS product listed on the COMPONENTS statement in prefix.MICS.PARMS(JCLDEF).

Once you have defined all of the parameters in prefix.MICS.PARMS, submit the ALLPGEN job:

```
    SUB 'prefix.MICS.CNTL(ALLPGEN)'
```

After ALLPGEN has executed, review the output to ensure that it executed correctly. If the output indicates that parameter changes are required, you must make the changes to the appropriate prefix.MICS.PARMS members and rerun ALLPGEN.

ABOUT cccPGEN

The ALLPGEN job runs the individual generation jobs named cccPGEN, where ccc is the component ID. These jobs will generate the macros for just the product to which they pertain. These jobs may be used in place of ALLPGEN if desired. For example, if you have run ALLPGEN and need to make changes to CICS parameters, you could submit the CICPGEN job:

```
    SUB 'prefix.MICS.CNTL(CICPGEN)'
```

Any time you wish to change any CA MICS parameters, you must update the appropriate members in the prefix.MICS.PARMS parameter library and then run ALLPGEN or the correct cccPGEN generation job so that the parameters will take effect.

Note that it may be necessary to regenerate JCL as a result of executing a cccPGEN. For more information about the individual product's parameter generation jobs, refer to the cccPGEN discussion in each data integration product guide.

Generate Base System Code (BASPGEN)

Figure 3-3 identifies each of the system code generation processes for the CA MICS Base Component (BAS), their PARMS member inputs, and the output members, 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.

The following CA MICS system code generation routines are SAS programs executed as part of the job BASPGEN. The applicable input PARMS members that must be completed prior to BASPGEN execution are identified in the parentheses below for each process.

- o Computing System Generation (SYSID)
- o Site Characteristics Generation (SITE)
- o Performance Group Name Generation (PRFGP)
- o Time Zone Generation (ZONE)

System Code Generator	MICS.PARMS Input Member	USER.SOURCE Output Members and SAS MACRO Names		MUOLIB Output SAS FORMATs
SYSIDGEN	SYSID	SYSIDOPU	SYSIDOPU	$SYSID $SYSIDN
SITEGEN	SITE	#BASMSTR	_NAME _YROPTN _WKSTART	YRSTART
		$BASMSTR	NAME YROPTN WKSTART	
		CKPTCNT	CKPTCNT	
PRFGPGEN	PRFGP			$PRFGP
ZONEGEN	ZONE			$ZONEHDB $ZONEHDA SF_ZN $ZONENM $ZONEREG

```
| MONEY     |     -      | These FORMATs are --> MONEY      |
|           |            |   output to the   --> MONEYR     |
|           |            |   Complex Level                  |
| MNTHFMT   |     -      |  (sp.MICS.MCOLIB) --> MNTHFMT    |
|           |            |                    |             |
|           |            |                    |             |
+-----------+------------+--------------------+-------------+
```

3.5.6 Generate and Allocate the Database

This section is the most important phase of the installation process and should be the one in which the greatest care is taken to ensure that the database has been correctly allocated and initialized.

The steps in generating, allocating, and initializing the CA MICS Database are:

1 - Database Macro Generation
2 - CA MICS Database and File Space Allocation
3 - Generation Data Group Catalog Index Creation

3.5.6.1 Database Macro Generation

This step assumes that you have already finalized your data retention parameters in prefix.MICS.PARMS(DBMODEL). If you have not finalized these parameters and run the DBSIZE job to determine the implications of your specifications, see Section 2.3.4.2, "Executing the Database Space Modeling Facility."

```
SUB 'prefix.MICS.CNTL(DBSIZE)'
```

Once the output of DBSIZE has been reviewed and verified, you may generate the SAS MACROs required to define the database by executing the CYCLEGEN job.

```
SUB 'prefix.MICS.CNTL(CYCLEGEN)'
```

Note: CYCLEGEN should only be run when it is mandatory to change a database retention definition. First make sure that the DBMODEL member contains the correct definitions.

3.5.6.2 CA MICS Database and File Space Allocation

The OS data sets that hold your database, and, optionally, certain work files used by the update jobstream, are allocated by running the ALLOCATE job in prefix.MICS.CNTL.

First, however, the ALLOCATE job must be edited to fill in certain JCL parameters, such as space requirements, device types, and if necessary, the specific volume serials for the allocations. You can use the Database Space Estimator Libname Summary Report from your DBSIZE run to estimate how much space is needed for each of the five data sets in the database.

Figure 3-4 illustrates the ALLOCATE job statements. Note that if your DASD requirements are such that any of the five OS data sets (DETAIL, DAYS, WEEKS, MONTHS, and YEARS) must be larger than a single volume, you should use the facilities described in Section 2.3.3.3.2.4, CA MICS Database Split Table (DBSPLIT). Note also that the permanent DAYSMF DD statements are generated only for components that you have installed.

Note: The ALLOCATE job does NOT allocate the incremental update database files or the SPLITSMF job output SMF data files. These data sets are associated with a specific product in the unit database and are allocated by the individual product's cccIUALC job, or are dynamically allocated as required. Data set allocation parameters are specified in the product's prefix.MICS.PARMS(cccOPS) member. See the individual product guides for more information on incremental update database file allocation.

While EDITing the ALLOCATE member in the MICS.CNTL library, you can FIND the statements requiring user-provided information by looking for eye-catching arrows ('<--<<').

```
//BR14 EXEC PGM=IEFBR14
//*
//* DATABASE ALLOCATION :
//* SPECIFY SPACE, VOLSER, AND UNIT TYPE
//* FOR THE DATABASE FILES.
//* USE RECOMMENDED SPACE FROM "DBSIZE" REPORT.
//*
//DETAIL DD DSN=EYC.MICS.DETAIL,
// DISP=(NEW,CATLG,DELETE),DCB=(DSORG=PS,RECFM=FS),
// SPACE=(TRK,(____,___)),UNIT=____,VOL=SER=______    <--<<
//DAYS DD DSN=EYC.MICS.DAYS,
// DISP=(NEW,CATLG,DELETE),DCB=(DSORG=PS,RECFM=FS),
// SPACE=(TRK,(____,___)),UNIT=____,VOL=SER=______    <--<<
//WEEKS DD DSN=EYC.MICS.WEEKS,
// DISP=(NEW,CATLG,DELETE),DCB=(DSORG=PS,RECFM=FS),
// SPACE=(TRK,(____,___)),UNIT=____,VOL=SER=______    <--<<
//MONTHS DD DSN=EYC.MICS.MONTHS,
// DISP=(NEW,CATLG,DELETE),DCB=(DSORG=PS,RECFM=FS),
// SPACE=(TRK,(____,___)),UNIT=____,VOL=SER=______    <--<<
//YEARS DD DSN=EYC.MICS.YEARS,
// DISP=(NEW,CATLG,DELETE),DCB=(DSORG=PS,RECFM=FS),
// SPACE=(TRK,(____,___)),UNIT=____,VOL=SER=______    <--<<
//*
//* CHECKPOINT AND RESTART.CNTL ALLOCATION :
//* SPECIFY VOLSER AND UNIT TYPE FOR THE
//* CHECKPOINT AND RESTART FILES
//*
//CHECKPT DD DSN=EYC.MICS.CHECKPT.DATA,
// DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(5,2)),
// UNIT=____,VOL=SER=______                           <--<<
//RESTART DD DSN=EYC.MICS.RESTART.CNTL,
// SPACE=(TRK,(5,5)),DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),
// DISP=(NEW,CATLG,DELETE),
// UNIT=____,VOL=SER=______                           <--<<
```

Figure 3-4. ALLOCATE Job Contents (Part 1 of 2)

```
//*
//* DAYSMF FILES ALLOCATION :
//* SPECIFY SPACE, VOLSER, AND UNIT TYPE
//* FOR THE PERMANENT DAYSMF FILES
//*
//USER DD DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(____,___)),UNIT=____,VOL=SER=______,   <--<<
// DSN=EYC.MICS.USER.DATA
//*
//* EACH OF THE FOLLOWING ARE ALLOCATED ONLY IF REQUIRED
//*
//TSO DD DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(____,___)),UNIT=____,VOL=SER=______,   <--<<
// DSN=EYC.MICS.TSO.DATA
//RMF DD DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(____,___)),UNIT=____,VOL=SER=______,   <--<<
// DSN=EYC.MICS.RMF.DATA
//SMF DD DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(____,___)),UNIT=____,VOL=SER=______,   <--<<
// DSN=EYC.MICS.SMF.DATA
//CIC DD DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(____,___)),UNIT=____,VOL=SER=______,   <--<<
// DSN=EYC.MICS.CIC.DATA
//IMS DD DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(____,___)),UNIT=____,VOL=SER=______,   <--<<
// DSN=EYC.MICS.IMS.DATA
```

Figure 3-4. ALLOCATE Job Contents (Part 2 of 2)

Database Timespan File Allocation

The Database Space Estimator Libname Report estimates the space requirements in terms of tracks for each of the five database timespans that equate to five OS data set allocations. There is no advantage to cylinder allocation for a SAS file, and the update process is not affected by their being on a single volume.

Modify each of the SPACE, UNIT, and VOLUME parameters to be in accordance with your database allocation requirements. For example, to allocate the database with the following track allocations to the 3390 volume MICS01, the statements would be modified as shown below.

```
 //DETAIL DD DSN=EYC.MICS.DETAIL,
 // DISP=(NEW,CATLG,DELETE),DCB=(DSORG=PS,RECFM=FS),
 // SPACE=(TRK,(3000,150)),UNIT=3390,VOL=SER=MICS01    <--<<
 //DAYS DD DSN=EYC.MICS.DAYS,
 // DISP=(NEW,CATLG,DELETE),DCB=(DSORG=PS,RECFM=FS),
 // SPACE=(TRK,(2000,100)),UNIT=3390,VOL=SER=MICS01    <--<<
 //WEEKS DD DSN=EYC.MICS.WEEKS,
 // DISP=(NEW,CATLG,DELETE),DCB=(DSORG=PS,RECFM=FS),
 // SPACE=(TRK,(1500,075)),UNIT=3390,VOL=SER=MICS01    <--<<
 //MONTHS DD DSN=EYC.MICS.MONTHS,
 // DISP=(NEW,CATLG,DELETE),DCB=(DSORG=PS,RECFM=FS),
 // SPACE=(TRK,(2500,125)),UNIT=3390,VOL=SER=MICS01    <--<<
 //YEARS DD DSN=EYC.MICS.YEARS,
 // DISP=(NEW,CATLG,DELETE),DCB=(DSORG=PS,RECFM=FS),
 // SPACE=(TRK,(1000,250)),UNIT=3390,VOL=SER=MICS01    <--<<
```

Figure 3-5. ALLOCATE Database Example

ALLOCATING CHECKPOINT AND RESTART FILES

Modify each of the UNIT, and VOLUME parameters to be in accordance with unit and volume that are to be used for the Checkpoint and Job Restart Files. For example, to allocate the files to the 3390 volume MICS01, the statements would be modified as shown below.

```
 //CHECKPT DD DSN=EYC.MICS.CHECKPT.DATA,
 // DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(5,2)),
 // UNIT=3390,VOL=SER=MICS01                           <--<<
 //RESTART DD DSN=EYC.MICS.RESTART.CNTL,
 // SPACE=(TRK,(5,5)),DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),
 // DISP=(NEW,CATLG,DELETE),
 // UNIT=3390,VOL=SER=MICS01                           <--<<
```

Figure 3-6. ALLOCATE Checkpoint Example

ALLOCATING WORK FILES

The DAILY job requires a work file for each CA MICS product that takes its input from SMF, if there are two or more of them installed (refer to the discussion of the SMFRECORDING statement in Section 2.3.3.2.1.1, "Control Definitions"). If work files are used, they will have names of the following form:

prefix.MICS.component.DATA

where "component" is any of the three letter component identifiers. These files are used to pass SMF data from the DAYSMF job step to the update processor job step for their product. These data sets can be optionally defined as permanent or temporary. The definition is made in the JCLDEF PARMS member and is described as the DAYSMF option in section 2.3.3.2.1.3, Execution Option Definitions. These work files will not be included in this job if the DAYSMF option described above is set to TEMPORARY.

Note: These data sets apply only to the DAILY job and are populated in the DAYSMF step. If you are using the optional incremental update facility for two or more products, then you may choose to use the SPLITSMF job and associated INCRSPLIT USE option for INCRccc job input data preparation. Data sets created by the SPLITSMF job are allocated dynamically when required, with data set allocation parameters being specified in the prefix.MICS.PARMS(cccOPS) INCRSPLIT USE parameter. Please see the individual product guides for more information on incremental update SPLITSMF job output data set allocation.

Work file space requirements can be estimated by either of the following methods:

1. For each product, use the output of IFASMFDP to obtain the number and average length of each type of record used by the component. Section 6.1.2 of the Batch and Operations Analyzer Guide lists the record types used by each component that takes data from SMF. Estimate the amount of space needed to hold these records.

2. Allocate files temporarily and run the DAYSMFR step of the DAILY job. This step produces the work files. See how much space is actually needed for each file, and use those figures for the permanent allocation.

Modify each of SPACE, UNIT, and VOLUME parameters to be in accordance with the work file requirements. For example, to allocate the work files with the following track allocations to the 3390 volume MICS02, the statements would be modified as shown below.

```
 //USER DD DISP=(NEW,CATLG,DELETE),
 // SPACE=(TRK,(100,050)),UNIT=3390,VOL=SER=MICS02,   <--<<
 // DSN=EYC.MICS.USER.DATA
 //TSO DD DISP=(NEW,CATLG,DELETE),
 // SPACE=(TRK,(300,100)),UNIT=3390,VOL=SER=MICS02,   <--<<
 // DSN=EYC.MICS.TSO.DATA
 //RMF DD DISP=(NEW,CATLG,DELETE),
 // SPACE=(TRK,(200,100)),UNIT=3390,VOL=SER=MICS02,   <--<<
 // DSN=EYC.MICS.RMF.DATA
 //SMF DD DISP=(NEW,CATLG,DELETE),
 // SPACE=(TRK,(700,200)),UNIT=3390,VOL=SER=MICS02,   <--<<
 // DSN=EYC.MICS.SMF.DATA

 //CIC DD DISP=(NEW,CATLG,DELETE),
 // SPACE=(TRK,(300,100)),UNIT=3390,VOL=SER=MICS02,   <--<<
 // DSN=EYC.MICS.CIC.DATA
```

Figure 3-7. ALLOCATE Work File Example

When you have finished EDITing it, you can submit the job:

```
    SUB 'prefix.MICS.CNTL(ALLOCATE)'
```

3.5.6.3 Generation Data Group Catalog Index Creation

The CA MICS backup and archive processes require a number of generation data groups. The actual data set names used are listed in section 4.3.10, Data Set Names.

The use of generation data groups requires that a special index be created in the control volume (or VSAM) catalog. The CA MICS job GDGSGEN will generate the statements to create this index for either type of catalog. Whether the statements are generated for an OS control volume or a VSAM catalog is determined by whether you specify the CVOLSER or VCATSER parameter in the JCLDEF parms member.

Before each index is built, it is deleted. These DLTX (or DELETE) statements will cause an error message if no entry exists. This is done so that the number of entries can be respecified, without the user having to delete each of the index entries.

DLTX and BLDG (or DELETE and DEFINE) will fail if there is a cataloged data set with the same index. IEHPROGM (or IDCAMS) will issue a message and give a return code of 8. This 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, you must 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 uncataloged in step 1.

The job GDGSGEN is submitted for execution to create the generation data group entries for each component. This job will examine the generation data groups defined to the CA MICS system and compare this set against the catalog. An entry will be created for any generation data groups which have been defined to CA MICS that are not found. GDGSGEN may be run at any time without affecting the CA MICS data sets that are already allocated.

Note: The GDGSGEN job does NOT create the index entries for the optional incremental update tape database files or for any optional tape DETAIL files. These GDG indexes are associated with a specific product in the unit database and are created by the individual product's cccIUGDG or cccGDGGN job. Please see the individual product guides for more information on optional incremental update tape database files and/or optional tape DETAIL files.

To run GDGSGEN submit the job as follows:

```
SUB 'prefix.MICS.CNTL(GDGSGEN)'
```

3.5.7 Finalize a Database Unit

After the steps described in the previous sections have each been successfully completed, the system is ready for testing in preparation for production use. It is recommended that the testing be conducted as described below:

1. Review the CLISTs, JCL procedures, and JCL jobs generated during CA MICS installation. If any of them do not meet your needs, you should not change them directly (refer to Section 2.3.3.3, JCLGEN Parameters for Special Requirements).

2. Review the CA MICS operational facilities/procedures in Chapter 4, Operation, of this guide.

   a. Read Section 4.1, Overview, in detail. This is an overview of CA MICS operation and facilities.

   b. Scan Section 4.2, Operational Guidelines. This is a high-level guide or road map to CA MICS operations. Read Section 4.2.1, Getting Started in detail.

   c. Review Section 4.3.4, Operational Status and Tracking. You will use Operational Status and Tracking for initial CA MICS processing.

3. Select one day's data for all the CA MICS products that you have installed. Put this data into the CA MICS input data sets you identified with the DD cards you supplied in the INPUTccc members of prefix.MICS.PARMS (refer to Sections 2.3.3.2 for descriptions of these PARMS members).

4. If you activated incremental update for one or more products in this unit database, run the cccIUALC job for each of these products. The cccIUALC job allocates the product's incremental update checkpoint and database files.

   ```
      SUB 'prefix.MICS.CNTL(cccIUALC)'
   ```

   Also, if you specified
   ```
      INCRDB TAPE
   ```
   for one or more of these products, run the product's cccIUGDG job to create GDG indexes for the incremental update tape database files.

```
SUB 'prefix.MICS.CNTL(cccIUGDG)'
```

5. Initialize the checkpoint file, 'prefix.MICS.CHECKPT.DATA':

   ```
   SUB 'prefix.MICS.CNTL(CKPTINIT)'
   ```

6. Invoke the CA MICS Workstation Facility (MWF), select option 5, CA MICS Administrator Facility (MAF), and then select option 1, Operational Status and Tracking. The Operational Status and Tracking display should list each active unit Database.

7. Run a DAILY job.

```
*************************************************************
*  Note that jobs which use the MICSDU PROC (for example    *
*  DAYALL, DAYRSR, RESTORE, and BACKUP) include a MICSABND  *
*  procstep which flushes under normal operation.  All      *
*  other jobsteps in the CA MICS operational jobs should    *
*  have return codes of zero.                               *
*************************************************************
```

   a. Enter the Operational Status and Tracking DAILY command in the Cmd column for the unit database.

   b. The DAILY command will execute the DAILY job and the BACKUP job. See Section 4.3.3 for more information on DAILY processing.

   c. After DAILY and BACKUP complete, use Operational Status and Tracking to review processing status.

      o Enter REFRESH command to update the display.

      o Enter STATUS command to review checkpoint status.

      o Enter CHECKPT command to review the checkpoint update date/time ranges.

      o Enter HISTORY command to review the input data that was read, processed, and/or dropped.

      o Enter RSTATUS command to test the RSTATUS job.

8. If this is a test unit database, manually execute the WEEKLY, MONTHLY, and YEARLY jobs to check out the JCL (Refer to Section 4.3.3). If this is not a test

unit, skip this step.

```
SUB 'prefix.MICS.CNTL(WEEKLY)'
SUB 'prefix.MICS.CNTL(MONTHLY)'
SUB 'prefix.MICS.CNTL(YEARLY)'
```

Do NOT use the Operational Status and Tracking for this exercise. The WEEKLY, MONTHLY, and YEARLY commands execute DAILY processing in addition to the weekly/monthly/yearly cycle close-out. DAILY processing would reject all input as duplicate data and terminate with a U300 abend because you have not yet provided a second day's input data.

NOTE: If you specified any of the following parameter combinations,

```
ARCHIVE AUDIT YES JOB

ARCHIVE HISTW YES JOB

ARCHIVE HISTM YES JOB
```

in prefix.MICS.PARMS(JCLDEF) then you may also need to execute the stand-alone AUDIT, HISTW, and/or HISTM jobs to perform archive tape processing.

If you specified the AUTOSUBMIT option on the ARCHIVE statements, then the AUDIT, HISTW, and/or HISTM jobs will be submitted automatically by the WEEKLY job WEEK300 step and/or MONTHLY job MONTH300 step. Otherwise, you will need to submit the AUDIT, HISTW, and/or HISTM job manually.

9. The contents of the database should be examined to ensure that the definitions and generations have all been properly applied.

   Run the database check jobs (cccCHECK) which will produce PROC PRINT listings of the first 10 records from the 01 cycle of all files in the DETAIL timespan and PROC FREQ TABLES LISTs of the 01 cycle of all files in DAYS and the 00 cycle of all files in the WEEKS and MONTHS timespans.

   These listings should be used to verify the account code, cost center code, job group, application unit, etc., definitions.

If this job completes with a non-zero return code, usually 8, then one or more component database files (for example, CICCSUnn) that should be present are not. This is a condition that was not expected and must be investigated since you have just run a DAILY job in step 7 above. Check the SASLOG to determine the files involved and the reason why, and then decide what action to take.

10. Run a database BACKUP operation--Enter Operational Status and Tracking BACKUP command.

11. Run a database RESTORE operation--Enter Operational Status and Tracking RESTORE command.

12. If you need to reset the database to an empty state, perform the following steps:

    - Delete the database and checkpoint files.

      ```
      prefix.MICS.CHECKPT.DATA
      prefix.MICS.RESTART.CNTL
      prefix.MICS.DETAIL
      prefix.MICS.DAYS
      prefix.MICS.WEEKS
      prefix.MICS.MONTHS
      prefix.MICS.YEARS
      ```

      (The incremental update checkpoint and database files, if allocated, were cleared by the last successful DAILY job and need not be deleted to reset the database.)

    - Rerun the ALLOCATE job.

    - Rerun the CKPTINIT job.

      (The following steps are optional, and need not be done if you have no GDGs in the test database, or do not care about the extra generations which will remain from the previous tests.)

    - Uncatalog the generation data sets for the backup, audit, and history files.

    - Rerun GDGSGEN.

13. Set-up to run CA MICS in test mode on a daily basis (see Section 4.2.1, Getting Started).

14. Use the procedures discussed in the next section to gradually move the CA MICS system to production status.

3.6 Finalize Database Complex

Once all of the database units are created and/or regenerated, the remaining tasks are list the contents of the formats in the CA MICS load library, print documentation for your users, and resolve any remaining administrative issues. The following sections describe these activities in greater detail.

1 - Produce CA MICS Documentation
2 - Resolve Administrative Items

3.6.1 Produce CA MICS Documentation

CA MICS generates printed documents from MWF Document Access or from a batch job. We recommend producing documents from Document Access, an ISPF panel-driven facility for selecting documents to print. Document Access includes full tutorial support.

Activate the CA MICS Workstation Facility as explained in Section 3.3.6 of this guide and select Option 1 for Document Access.

CA MICS also provides backup methods for printing documents, including jobs to assist you in producing your own copies of the CA MICS documentation. These facilities can produce all of the CA MICS guides or any individual guide. The jobs, described below, are available for use after the CA MICS database complex is fully defined.

If you need to produce CA MICS documentation prior to installing the database complex, the EARLYDOC job in sharedprefix.MICS.CNTL can be used after the CA MICS production libraries have been loaded from the distribution tape. Documentation produced before products are generated reflects the default component definitions shipped with the system.

IMPORTANT NOTES:

If you do not have upper and lower case printing capability, a translation feature is available as part of the document production facility. It is discussed in the section on the MICSDOC Procedure that follows.

The DOCUMENT print program assumes a page size of 60 lines. If the paper you use will not hold 60 lines on a page, there is an option to change the number of lines per page. It is discussed in the section on the MICSDOC Procedure that follows.

Guides for optional CA MICS products, and chapters or sections of guides for these optional products, are not available for printing unless you have the products installed.

The sharedprefix.MICS.CNTL(DOCUMENT) job can be used to print CA MICS guides. The DOCUMENT job uses the MICSDOC procedure described below. You may need to adjust this job's JES parameters to ensure that it will print all of the lines it produces. In this case, a permanent change should be made to the JCLINFO member of sharedprefix.MICS.PARMS before you regenerate the DOCUMENT job.

MICSDOC Procedure:

To produce a single manual, build and submit the following job:

```
//jobname  JOB
//S1       EXEC MICSDOC,MANUAL=x
```

where "x" must be a one or two character value corresponding to one the CA MICS documents listed below:

AH	- CA MICS Tandem Option Guide
AI	- CA MICS CA ASTEX Option Guide
BE	- CA MICS How to Use the PSP
BS	- CA MICS System Administrator Guide
C	- CA MICS Network Analyzer Option Guide
CK	- CA MICS Space Collector Option Guide
CL	- CA MICS IMS Log Data Extractor Option Guide
CM	- CA MICS Data Transfer Option for VM/CMS Guide
FB	- CA MICS Analyzer for TSO Guide
FC	- CA MICS Hardware and SCP Analyzer Guide
FD	- CA MICS Batch and Operations Analyzer Guide
FE	- CA MICS Analyzer Option for CICS Guide
FG	- CA MICS Analyzer Option for IMS Guide
FH	- CA MICS System Reliability Analyzer Option Guide
FI	- CA MICS Analyzer Option for VM/CMS Guide
FJ	- CA MICS Space Analyzer Option Guide
FM	- CA MICS Analyzer Option for MeasureWare Guide
FN	- CA MICS Analyzer Option for DB2 Guide
FO	- CA MICS Analyzer Option for MQSeries Guide
FQ	- CA MICS Analyzer Option for VSE/Power Guide
FT	- CA MICS Tape Analyzer Option Guide
FV	- CA MICS Analyzer Option for VAX/VMS Guide
FW	- CA MICS Web Analyzer Option Guide
H	- CA MICS Standard Reports Guide
I	- CA MICS Database Structure and Content Guide
JD	- CA MICS Document Access Guide
JR	- CA MICS MICF Reference Guide
JU	- CA MICS MICF User Guide
L	- CA MICS Planning, Installation, Operation, and Maintenance Guide
M	- CA MICS System Modification Guide
PA	- CA MICS Capacity Planner Option Guide

PD	-	CA MICS Performance Manager Option Guide
PG	-	CA MICS StorageMate Option Guide
PK	-	CA MICS Accounting and Chargeback Option Concepts and Overview Guide
PL	-	CA MICS Accounting and Chargeback Option User Guide
W	-	CA MICS Analyzer Option for CA IDMS Guide

To produce several documents, build and submit a multistep job like the following:

```
//jobname  JOB
//S1       EXEC MICSDOC,MANUAL=x
//S2       EXEC MICSDOC,MANUAL=y
```

where "x" and "y" are any of the one or two character manual ID values shown above.

To produce only selected chapters of a document, build and submit a job as above, but qualify the MANUAL= value with a chapter number. For example, to produce the chapters discussing the Hardware and SCP information areas in the Hardware and SCP Analyzer Guide, the following job would be built:

```
//jobname  JOB
//*
//S1       EXEC MICSDOC,MANUAL=FC5   Files for both areas
//S2       EXEC MICSDOC,MANUAL=FCXD  Data Dictionary
```

To request multiple copies of the printout from MICSDOC, use the COPIES parameter as in the following example.

```
//S1       EXEC MICSDOC,MANUAL=x,COPIES=2
```

If you do not have upper and lowercase printing at your site, request the translation of all text to all uppercase characters as follows:

```
//jobname  JOB
//S1       EXEC MICSDOC,MANUAL=x,CASE=UCASE
```

where "x" is any of the values shown above.

In addition to the lower to uppercase translation, this option translates certain special characters to other characters more commonly found on printers that do not support lowercase printing. This translation includes translating vertical bar to "I" and underscore to ".".

If you use paper that will not hold 60 lines of output per page, change the number of lines of output per page by using the NLINES option. For example, to request 55 lines per page, code:

```
    //jobname  JOB
    //S1       EXEC MICSDOC,MANUAL=x,SYSPARM='NLINES=55'
```

The value for NLINES ranges from 20 to 60.

Much of the document text is formatted assuming 60 lines per page (e.g., full page figures of report output). For this reason, NLINES should not be used unless required, and if used, the value for NLINES should be as close to 60 as possible.

When printing CA MICS documentation, use the SNUM option if you wish to print Table of Contents entries without document section numbers. For instance, this section would normally be listed as follows in the Detail Table of Contents.

```
     3.6.1 Produce CA MICS Documentation . . . . . . page #
```

Code the SNUM option in the SYSPARM field to suppress text section numbering.

```
    //jobname  JOB
    //S1       EXEC MICSDOC,MANUAL=x,SYSPARM='SNUM'
```

By default, MICSDOC prints text section numbers in the table of contents.

For printing the text of the manuals, the MICSDOC procedure uses the SYSOUT specification you gave on the DOCUMENT parameter in sharedprefix.MICS.PARMS(JCLDEFC) when the CA MICS system JCL was generated.

EARLYDOC Job:

The EARLYDOC job is used to print a CA MICS guide or guide chapter once the CA MICS production libraries have been loaded from the distribution tape. This job requires the use of SAS, but not necessarily the CA MICS copy.

Review and complete the job card and the parameters on the EXEC statement of sharedprefix.MICS.CNTL(EARLYDOC). The EXEC statement is as follows:

```
//DOC  EXEC MICSDOC,
// MANUAL='__',                   <-- REQUIRED
// WKUNIT='SYSDA',                <-- REQUIRED
// SHRPFX='sharedprefix.MICS',    <-- NO TRAILING PERIOD
//*
// SASNAME='entry',               <-- SASHOST for V8
//*                                   SAS     for V9
// SASPFX='sas.prefix',           <-- NO TRAILING PERIOD
// SASCFG1='config',              <-- CNTL(BATCH) IF SAS V8
//*                                   CNTL(BATW0) IF SAS 9.1.3
//*                                   CONFIG(BATCH) IF SAS 9.2
//*                                   For SAS 9.2 uncomment
//*                                   and specify next 3:
//*SASCFG2='CONFIG(COMMON)',          CONFIG(COMMON)
//*SASCFG3='CONFIG(ENW0)',            CONFIG(ENW0)
//*SASCFG4='CONFIG(SITE)',            CONFIG(SITE)
//*
//*AUTOLIB='fixed.block.AUTOLIB',<-- For 9.1.3 only
//*                                   uncomment and
//*                                   point to FB version
//*                                   of SAS AUTOCALL library
//*
// LANG1='',                      <-- SAS V9 LANGUAGE
//*                                   EN = English
//*                                   blank if V8
// LANG2='',                      <-- SAS V9 encoding value
//*                                   W0. = EBCDIC (1047)
//*                                   MUST HAVE TRAILING .
//*                                   blank if V8
//*
// CASE=,                         <-- UPPER CASE ? (UCASE)
// DOCOUT='A',                    <-- OUTPUT SPEC FOR GUIDE
// SYSPARM=' '                    <-- DOCUMENT OPTIONS
```

EARLYDOC uses an instream version of the standard MICSDOC procedure. Therefore, use of the MANUAL, CASE, and SYSPARM parameters is the same as described above for the DOCUMENT job.

The parameters required to execute the MICSDOC instream procedure in EARLYDOC follow. You must specify the values for these parameters. The defaults provided for all other parameters are normally adequate, though you should verify that these values are valid for your installation before submitting the job.

WKUNIT

The unit for the SYSUT3 and SYSUT4 work data sets needed by IEBCOPY.

SHRPFX

The sharedprefix for the database complex. Include the CA MICS level node as needed by your site. Note that this name is specified with no trailing period.

SASNAME

The library member name of the SAS program to be used, also called the entry point name. Normally this is SASHOST for version 8, and SAS for version 9.

SASPFX

The common data set name prefix for your SAS libraries. Note that this name is specified with no trailing period.

SASCFG1-4

The data set name qualifier and member name of your SAS configuration files. Specify only SASCFG1 for SAS versions 8 and 9.1.3. Normally the value is CNTL(BATCH) for SAS version 8 and CNTL(BATW0) for SAS version 9.1.3. For SAS version 9.2 or above, SASCFG1 is normally CONFIG(BATCH). To specify additional configuration files for SAS version 9.2 or above, uncomment SASCFG2, SASCFG3 and SASCFG4. The values are normally CONFIG(COMMON), CONFIG(ENW0), and CONFIG(SITE). Note that the full data set names are built by prefixing the value of SASPFX.

AUTOLIB

The SAS AUTOCALL library that is being converted to fixed block with SAS 9.1.3. It is not required for SAS Version 9.2 and higher.

LANG1

The language code required with SAS versions 9 and higher, for example, EN stands for English. The value is blank for SAS version 8.

LANG2

The encoding value required with SAS version 9 and higher. The value for EBCDIC is W0 and must have a trailing period (.). The value is blank for SAS version

8.

When you have completed editing the job, submit it by coding the following to print a CA MICS guide or chapter using the default document text shipped with the system.

```
SUB 'sharedprefix.MICS.CNTL(EARLYDOC)'
```

3.6.2 Resolve Administrative Items

There are several administrative items that should be resolved to ensure a smooth transition into production operation.

DASD Space Management Systems

If a DASD space management system is installed, data sets might be deleted automatically if the data sets are not defined to a master file table. Please ensure that all of the CA MICS complex and unit data sets and the work data sets used in each of the daily jobs are added to this master file table. This prevents the DASD space management system from deleting CA MICS data sets.

Tape Management Systems

If a tape management system is installed, data sets might be automatically scratched if they are not properly defined (e.g., GDG tape data sets and TMS). Please ensure that all of the CA MICS tape files (backups and archives) have been properly defined to the tape management system.

System Backup

Ensure that backups of the CA MICS libraries are taken and are updated when system modifications are applied. It is prudent to have a duplicate copy of the CA MICS system stored offsite as a backup.

3.7 Production System Implementation

Once you have completed the instructions presented in section 2.3, "CA MICS Installation Planning and Parameter Specification", and the actual installation of the CA MICS system, CA MICS should be considered at the pre-production level. In other words, all of the libraries, code, JCL, etc., have been properly set up, but you must now gain practical experience in running the system, possibly revise system parameters, work out any problems that may have been encountered, and adjust report parameters, before CA MICS can be considered in full production status.

Once all of the above have been completed and you have gained confidence in your handling of the system, you are ready for production turnover. In other words, the status of CA MICS may move from pre-production to full production.

The following are tasks that should be done prior to activating CA MICS on a production basis.

1 - System Activation
2 - Exception Process Review and Adjustment
3 - Management Objective Report Tailoring
4 - Report Distribution Definition
5 - Operational Procedure Documentation

3.7.1 System Activation

This section recommends a series of actions to ensure a full system check-out and successful pre-production activation.

1. List the JCL streams in the job control library (MICS.CNTL) named DAILY, WEEKLY, MONTHLY, and YEARLY. Read the Operations chapter thoroughly and review the JCL streams just listed to gain a good understanding of the operational processes. Verify in this review that the streams are in full accordance with installation standards.

2. Verify that the Checkpoint File is correctly initialized for system activation. Use the Operational Status and Tracking STATUS command to display Checkpoint File status line information. Last Completed Step should be 900 for all CA MICS jobs (DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, and RESTORE). Last Completed Date should be the date CKPTINIT was executed, the first date of the current week, the first date of the current month, and the first date of the current year respectively for the DAILY, WEEKLY, MONTHLY, and YEARLY jobs.

3. List the catalog for the prefix(es) to which all of the CA MICS data sets were appended. Ensure that all of the data sets that are referenced in the JCL procedures listed in step 1 are allocated and cataloged.

4. Determine how the installation's normal SMF dumping and accumulation process can be used to provide the files necessary for the DAILY process.

5. Use the Operational Status and Tracking DAILY command to execute the DAILY process for the next three evenings and check for correct operation.

6. Each morning check Operational Status and Tracking for CA MICS DAILY processing results, review report outputs to gain familiarity with the data, monitor the run times and archive process, and in general look for possible problems.

7. Using the documentation in chapter 4, "Operations", and listings of the JCL streams, work through how a restart would be handled. Also, test the restoring of the database from the backup files.

8. After several nights of running experience, begin using CA MICS facilities for automatic scheduling of DAILY, WEEKLY, MONTHLY, and YEARLY processing. Use the SCHEDULE

command instead of DAILY from the fourth day on.

9. Plan on running the system in this pre-production mode, under close monitoring and supervision, at least until you have scheduled and executed (successfully) both a WEEKLY and MONTHLY job stream at the required times.

10. During the pre-production period, begin implementing full production operation for the CA MICS system. See section 4.2.2, Production Operations, for guidelines on implementing CA MICS production processing with your installation's production batch processing facilities.

3.7.2 Exception Process Review and Adjustment

The running of CA MICS Exception Analyzer routines is controlled by the CREATE EXCEPTIONFILES parameter you specify in prefix.MICS.PARMS(EXECDEF) (refer to Section 2.3.5, Dynamic Execution Options, for a description of this parameter). As the CA MICS system is shipped, this parameter is specified as NONE, causing CA MICS to not run the Exception Analyzer routines as part of the CA MICS DAILY job. If you are going to run some or all of the Exception Analyzers, you must review these programs and change the values used for identifying exceptions to be appropriate for your installation's operating environment.

Follow the steps below to analyze and adjust the exception values:

1. Review Section 2.4, Exception Values, in the Standard Reports Guide. This section describes the Exception Value Analyzer, which enables you to adjust exception values to address your installation's requirements.

2. List the source code used for setting the exceptions from the source library (sharedprefix.MICS.SOURCE). You need only examine the modules for those components you listed on your CREATE EXCEPTIONFILES parameter statement. The Exception Analyzers shipped with the standard set of CA MICS products are:

```
    DYSMFEXC - Batch Exceptions
    DYRMFEXC - Hardware Utilization and System Control
               Program Exceptions
    DYTSOEXC - TSO Exceptions
```

while the Exception Analyzers shipped with optional CA MICS products are:

```
    DYcccEXC - ccc Exceptions
```

where ccc is the three character product identifier. These will only be in sharedprefix.MICS.SOURCE if your site licenses these products.

Review the listings to gain a basic understanding of how an exception is identified by the code.

3. Execute the Exception Value Analyzers referenced in Step 1. The output of these reports provides an initial set of values to be used for exception testing that are based on

data from your installation. Review the recommendations, which are listed by data element within exception number for quick reference, and decide whether to use the recommended value or your own judgment (which enables you to consider organizational and political factors).

4. Modify the exception analyzer routines to reflect the adjustments that should be made in tailoring each of the exception values to address the installation's requirements.

5. Activate the modified programs by changing the CREATE EXCEPTIONFILES statement which is contained in member EXECDEF of prefix.MICS.PARMS. Note that if you want the CA MICS Exception Reports to be printed for some or all of the exceptions found by the Analyzers, you will have to name them on the REPORT EXCEPTIONS parameter statement in the EXECDEF member (or use the MICF inquiry production reporting facilities, documented under MICF Administration in the MICF Reference Guide, to generate daily CA MICS exception reports). Both of these changes will take effect for your next day's processing cycle with no further action. Check the output of the next daily run to ensure that no operational problems were introduced. Carefully verify that all functions are still operative after the modifications were implemented.

6. You will find that the exceptions thresholds will require periodic adjustment using the above process.

3.7.3 Management Objective Report Tailoring

Many management objective reports compare observed system activity to objectives for that activity. You should examine the objectives shipped with the CA MICS system and change them if they are not appropriate for your operating environment and reporting needs.

The REPORT MBODAILY, REPORT MBOWEEKLY, and REPORT MBOMONTHLY statements in prefix.MICS.PARMS(EXECDEF) control the production of management objective reports. EXECDEF parameters are described in Section 2.3.5, Dynamic Execution Options (EXECDEF). Alternatively, you can use MICF inquiry production reporting facilities documented under MICF Administration in the MICF Reference Guide to generate the management objective reports. Executing the MBO reports within a Production Reporting Job Stream provides the benefit of multiple unit access. In addition, separating the reporting from the Operational Job releases the exclusive enqueue on the database, shortening the time the database is unavailable for additional reporting.

Refer to chapter 3 of the individual product guides for more information on standard reports included with each product.

3.7.4 Report Distribution Definition

This section suggests a plan for distributing and using exception and management objective reports. Use this information to help you develop a report distribution plan that meets your organization's needs. For more information about daily exception reports and the daily, weekly, and monthly management objective reports, refer to the Standard Reports Guide and the individual product guides.

If you are using MICF inquiry production reporting facilities for daily exception reports and/or daily, weekly, or monthly management objective reports, printed reports are reviewed online with the MICF Reports and Graphics option rather than being distributed on paper. See MICF Administration in the MICF Reference Guide and the MICF Tutorial in the MICF User Guide for a discussion of MICF inquiry production reporting and report security considerations.

The daily Management Objective Reports provide day-to-day management of systems with an hourly quantification of service, availability, and load. The reports may be distributed to System Performance, Systems Programming, Operations, and Time-Sharing Administration.

The Exception Management Overview should be made available to the data center managers having responsibility for system availability, operation, or performance.

The Severity Level Exception Summary should be generated specifically for use by the System Performance Department. This report contains a prioritized list of all critical, impacting, and warning exceptions. A copy of the critical exceptions report should be distributed to the same managers receiving the Exception Management Overview.

The Management Area Exception Summary should be separated by area reported (e.g., availability) and distributed as illustrated in Figure 3-8.

Consider the following in defining the report distribution plan:

1 - Itemize the reports to be produced daily, weekly, and monthly.
2 - Identify the individuals who will receive copies of the reports and when (daily, weekly, or monthly).

Management Area Exceptions	Department Distribution List
Availability Exceptions	Operations Quality Control Systems Programming System Performance
Performance Exceptions	Systems Programming System Performance
Productivity Exceptions	Applications Development Standards Control System Administrators
Service Exceptions	Data Center Management Operations Systems Programming System Performance
Standards Exceptions	System Administrators Standards Control Applications Development
Workload Exceptions	System Administrators Systems Programming System Performance
Security Exceptions	Data Center Management Security Control System Administrators

Figure 3-8. Management Area Exception Report Distribution

3.7.5 Operational Procedure Documentation

In preparing to turn the CA MICS system over to full production status, one of the most important steps is providing the documentation required for your installation's scheduling, operations, and production control departments.

It is recommended that the documentation procedures for CA MICS be written to be consistent with your organization's standards for documenting production systems.

The basic operation of the DAILY, INCRccc (optional incremental update for product ccc), SPLITSMF (optional SMF data split for INCRccc job input), WEEKLY, MONTHLY, YEARLY and BACKUP jobs should be described as to their purpose, scheduling considerations, and resource requirements. Your installation's standards may also require documentation for the SCHEDULE job which submits the other jobs if the CA MICS Job Scheduling Facility is used. Lastly, the CA MICS DAILY, WEEKLY, and MONTHLY jobs may "spin off" a corresponding xxxxxRPT job if you have set AUTOSUBMIT YES in prefix.MICS.PARMS(JCLDEF) (refer to Section 2.3.3.2.1.4, Database Unit JCL Definitions). The chapter on Operation provides descriptive material that may be used directly or modified for this purpose.

The procedure to be used for restart should be explained and the list of names and phone numbers for operations to contact in case of database problems should be documented.

The report distribution procedures should be provided to enable the production control department to properly distribute the reports to the appropriate individuals and departments in a timely manner.

The above points describe the CA MICS system as being handled similarly to other production systems. Such treatment will increase the overall productivity of the personnel assigned to CA MICS by enabling them to focus their time on the use of the CA MICS system.

3.8 Installation Checklists

Typically, a user is performing one or more installation functions. A checklist for each of these functions is contained in the following sections.

1. Installing a CA MICS complex for the first time.

2. Adding a new component to an existing database unit in an existing Database Complex.

3. Adding a new component to a new database unit in an existing Database Complex.

4. Creating a new database unit in an existing CA MICS complex and not adding new components.

5. Splitting an existing database unit.

6. Loading a newly licensed optional product from the encrypted CA MICS distribution medium.

```
*** IMPORTANT *** IMPORTANT *** IMPORTANT *** IMPORTANT ***
*                                                         *
*  In order to install CA MICS with a minimum of          *
*  difficulty, you must use the information provided      *
*  in the references for the steps of the checklist.      *
*                                                         *
*** IMPORTANT *** IMPORTANT *** IMPORTANT *** IMPORTANT ***
```

The checklists are constructed from a combination of CA MICS installation "stages." There are six stages:

1. Unload the Distribution Tape.
2. Define the Database Complex.
3. Define and Generate Component(s).
4. Reconfigure Existing Database Unit(s).
5. Generate New Database Unit(s).
6. Finalize the Database Complex.

The checklist for installing a CA MICS complex for the first time, for example, consists of the following stages:

1. Unload the Distribution Tape.
2. Define the Database Complex.

3. Define and Generate Component(s).
4. Generate New Database Unit(s).
5. Finalize the Database Complex.

3.8.1 Installing a CA MICS Complex

This checklist defines the tasks required to install a CA MICS complex for the first time or any subsequent re-installation. The checklist tasks are organized into the following groups:

1 - Unload the Encrypted Distribution Tape
2 - Define the Database Complex
3 - Define and Generate Products
4 - Generate the New Database Unit(s)
5 - Finalize the Database Complex

3.8.1.1 Unload the Distribution Medium

```
__ 1. Define Catalog Index Sharedprefixes

      Define the catalog index for the sharedprefix.

      No action is required if the high-level node-name
      already has been defined as an alias in the master
      catalog.

      +---------------------------------------------------+
      | Reference(s): Section 3.2.1                       |
      +---------------------------------------------------+

__ 2. Install CA Common Services and LMP Keys

      Ensure that CA Common Services is operational.

      CA License Management Program (LMP), one of the CA
      Common Services, is required before you can install
      or run CA MICS.

      To run CA MICS r11 or higher, you must define the
      keys needed for CA LMP.

      +---------------------------------------------------+
      | Reference(s): Section 3.2.2                       |
      +---------------------------------------------------+

__ 3. Review Cover Letter and PMLs

      Review the cover letter and associated product
      maintenance letters (PML) for the latest important
      technical information.

__ 4. Copy Load Files

      Copy the files which load the CA MICS system.

      To start the installation, two "tapeload" data sets
      must be created. They contain JCL and load modules
      that are required to complete the installation.

      To create the data sets:

        - See Section 3.2.3.1 if installing from ESD PAX.
        - See Section 3.2.3.2 if installing from tape.

      +---------------------------------------------------+
      | References: Sections 3.2.3, 3.2.3.1, 3.2.3.2      |
      +---------------------------------------------------+
```

__ 5. Load Production Libraries

Load the CA MICS production libraries using ESD PAX or tape. They both contain all code and documentation for the entire CA MICS product line in a compressed format. The installation job uses the CA LMP keys to identify and decode the modules that belong to your licensed CA MICS products.

Modules are written to disk by executing the program named LPEBLDD. It runs as a "nonauthorized" program. Some mainframe security systems, such as CA Top Secret, have a feature that restricts nonauthorized programs from writing authorized load modules. (An authorized load module is link-edited with an attribute of AC(1).) If your site has this security feature activated and is licensed for the CA MICS DASD Space Collector (VCC), then the load job will fail with an S913 abend while trying to write an authorized VCC load module. To prevent this abend, request your security administrator to permit LPEBLDD to write authorized load modules.

To load the production libraries:

- See Section 3.2.4.1 if installing from ESD PAX.
- See Section 3.2.4.2 If installing from tape.

```
+---------------------------------------------------+
| References: Sections 3.2.4, 3.2.4.1, 3.2.4.2      |
+---------------------------------------------------+
```

__ 6. Retain Distribution Tape and Load Libraries

Retain the "tapeload" libraries for future optional product installation. They are:

o sharedprefix.MICS.TAPELOAD.CNTL
o sharedprefix.MICS.TAPELOAD.LOAD

In addition, if you installed CA MICS from a tape, retain the tape for future optional product installation.

If you installed from ESD PAX, there is no need to retain the interim libraries since they can be downloaded from Support Online when they are needed.

__ 7. (CONDITIONAL) Conversion to SAS 9.1.3 ONLY

If you are converting to SAS 9.1.3, the SAS Autocall library (sasprefix.W0.AUTOLIB) must be converted to a fixed block data set to address issues documented in SAS Usage Note SN-013990. Review this SAS Usage Note before proceeding with the rest of this conversion checklist.

This data set should be used only for CA MICS, and is only necessary for SAS 9.1.3. The fixed-block version of the SAS Autocall library must be created with an LRECL of 80 and a BLKSIZE that is a multiple of 80.

Note that standard IBM utilities, such as IEBCOPY, will not copy variable blocked data sets to fixed block.

The following is a sample job using the SAS provided VBGT72 macro, referenced in SN-013990 to copy the SAS Autocall library. VBGT72 must be downloaded from the SAS institute support site before it can be used. In this example, VBGT72 was downloaded and then saved as member VBGT72 in sasprefix.VBGT72.LIBRARY:

```
//    ....JOB....
//VBGT72   EXEC SAS
//VBGT72   DD DISP=SHR,DSN=sasprefix.VBGT72.LIBRARY
//ISASAUTO DD DISP=SHR,DSN=sasprefix.W0.AUTOLIB
//AUTOOUT  DD DISP=(NEW,CATLG,DELETE),
//    DSN=micssasprefix.FB.W0.AUTOLIB,
//    UNIT=unit,VOL=SER=volume,
//    DCB=(LRECL=80,RECFM=FB,BLKSIZE=6160),
//    SPACE=(CYL,(1,1,45))
//SYSIN    DD *
 %INCLUDE VBGT72(VBGT72);
 %VBGT72(indd=ISASAUTO);
//
```

The JCL will need to be modified before submission to suit your site requirements.

Alternatively, the copy utility provided with ISPF (Option 3.3) can be used.

3.8.1.2 Define the Database Complex

```
__  1. Execute MICSLS1

       Activate the LOAD and PARMS libraries' release
       dependent modules for the SAS version in use.

       __ Edit sharedprefix.MICS.CNTL(MICSLS1) and fill in
          the parameters on the CNVTMCO EXEC statement.  Do
          not enter parameters on the PROC statement.

       __ Submit the following job:

            'sharedprefix.MICS.CNTL(MICSLS1)'

       +-------------------------------------------------+
       | Reference(s): Section 3.2.5                     |
       +-------------------------------------------------+


       NOTE: For SAS 9.1.3, use the fixed block library
             created in the previous section.

__  2. (OPTIONAL) Execute EARLYDOC

       Print the CA MICS documentation guides.

       If you choose to print guides before the database
       complex is defined, they will only reflect the default
       definitions shipped with the system.

       To print CA MICS documentation guides or chapters
       before defining the database complex:

       __ Edit sharedprefix.MICS.CNTL(EARLYDOC) and fill in
          the job card and the parameters on the EXEC
          statement at the end of the member.  DO NOT enter
          parameter values on the PROC statement.

       __ Submit the following job:

            'sharedprefix.MICS.CNTL(EARLYDOC)'

       +---------------------------------------------------+
       | Reference(s): Section 3.6.1                       |
       +---------------------------------------------------+

__  3. Define Complex Databases
```

Define the primary, unit, and test databases that are to comprise your installation's database complex. Note that the identity, contents, and options for each database unit must be completely defined.

```
+-----------------------------------------------------+
| Reference(s):  Section 2.3.1.4                      |
+-----------------------------------------------------+
```

__ 4. Apply Required SAS Maintenance

Apply required maintenance to the SAS program library. Ensure that a supported SAS release is being used.

Refer to the System Requirements section of the CA MICS System Administrator Guide for any special considerations and for the recommended maintenance for your CA MICS copy of SAS. This section describes modifications that are to be applied to the level of SAS in use. Make sure that any of the zaps defined are properly applied before proceeding.

__ 5. Review SAS Libraries

Review individual SAS libraries for CA MICS.

Refer to the System Requirements section of the CA MICS System Administrator Guide for the levels of SAS supported by CA MICS. Create a special set of SAS libraries for CA MICS use only. You provide the name of these libraries to CA MICS using the parameters in sharedprefix.MICS.PARMS(JCLDEFC) and also in the prefix.MICS.PARMS(JCLDEF) member for each database unit.

NOTE: All versions of SAS supported for CA MICS require the use of the SAS LIBRARY, AUTOLIB, SASCNTL, SASHELP, and SASMSG data sets. Create a copy of these data sets for CA MICS use only. You provide the name of these data sets to CA MICS: SASLIB, SASAUTOS, SASCONFIG, SASHELP, and SASMSG, respectively.

```
+-----------------------------------------------------+
| Reference(s):  Section 2.3.1.6 and                  |
|                Section 6.6, System Administrator    |
|                  Guide, (System Requirements)       |
+-----------------------------------------------------+
__  6. Customize JCLDEFC

       Customize the sharedprefix.MICS.PARMS(JCLDEFC) member
       to reflect your organization's standards.

       +---------------------------------------------------+
       | Reference(s): Section 2.3.1.6                     |
       +---------------------------------------------------+

__  7. Verify Parameters in JCL$Iccc, JCLNAMES, and JCLINFO

       Review the contents of the JCL$Iccc, JCLNAMES, and
       JCLINFO members in sharedprefix.MICS.PARMS.

       Parameters in JCL$Iccc, where ccc is the three
       character component identifier, may require
       modification to properly set job attributes such as
       TIME=, COPIES=, JOBNAME=, CLASS=, PRTY=, and LINES=.

       Make your modifications in the JCLINFO member of
       sharedprefix.MICS.PARMS.

       JCLNAMES may require modification if the default PROC
       or data set names used by CA MICS are not acceptable
       at your installation.

       +---------------------------------------------------+
       | Reference(s): Section 2.3.3.3.2.2 (JCLINFO)       |
       |               Section 2.3.3.3.2.2 (JCL$Iccc)      |
       |               Section 2.3.3.3.2.3 (JCLNAMES)      |
       +---------------------------------------------------+

__  8. Customize USERJCL

       Customize the sharedprefix.MICS.PARMS(USERJCL) member
       to reflect organization standards.

       +---------------------------------------------------+
       | Reference(s):  Section 2.3.3.2.2                  |
       +---------------------------------------------------+

__  9. Verify ISPF Environment

       Customize sharedprefix.MICS.PARMS to ensure the
       correct ISPF environment is defined and that it
       conforms with conventions at your site.  Make updates
       if needed.

       This review must include the names and blocksizes for
```

standard ISPF data sets and the naming conventions for data sets allocated by the CA MICS Workstation Facility (MWF) for each user. Note that, except for unusual requirements, the options provided in the members JCLDEFC and JCLNAMES of sharedprefix.MICS.PARMS should accommodate most installations' standards.

```
+-----------------------------------------------------+
| Reference(s):  Section 3.3.6.1                      |
|                Section 2.3.1.6 (JCLDEFC)            |
|                Section 2.3.3.3.2.3 (JCLNAMES)       |
+-----------------------------------------------------+
```

__ 10. Execute JCLGEN0

Generate the database complex-level JCL.

__ Edit sharedprefix.MICS.CNTL(JCLGEN0) and define the parameters on the EXEC statement. DO NOT enter parameter values on the PROC statement.

__ Submit the following job:

```
'sharedprefix.MICS.CNTL(JCLGEN0)'
```

```
+-----------------------------------------------------+
| Reference(s):  Section 3.3.2                        |
+-----------------------------------------------------+
```

__ 11. Move PROCs by Executing PROCJOB1

Move CA MICS PROCs to the installation PROCLIB.

__ Check the JCL generated by JCLGEN0 by examining the following sharedprefix.MICS.CNTL members:

```
PROCJOB1  MICSASM  MICSCOM  MICSDM    MICSDOC
MICSDOCH  MICSGT   MICSGU   MICSRPTS
```

__ (CONDITIONAL) You must determine whether the JCL and JES parameters are acceptable at your installation. If they are not, update the sharedprefix.MICS.PARMS members JCLDEFC or USERJCL and repeat the "Customize USERJCL" step.

OR

__ If the JCL is acceptable, move CA MICS PROCs to the installation PROCLIB by submitting the following job:

```
'sharedprefix.MICS.CNTL(PROCJOB1)'
```

```
+---------------------------------------------------+
| Reference(s):  Section 3.3.3                      |
+---------------------------------------------------+
```

__ 12. Customize Money Format

Customize monetary formats used with CA MICS if desired. CA MICS provides the MONEY and MONEYR formats for writing monetary values in output reports. A corresponding informat, @MONEY, is used for reading monetary values. If you use some currency notation other than U.S. dollars, edit sharedprefix.MICS.SOURCE alter the format or the informat for use with your currency notation.

```
+---------------------------------------------------+
| Reference(s):  Section 3.3.5                      |
+---------------------------------------------------+
```

__ 13. Execute DOCAGEN1

Build the ISPF tables needed by CA MICS Document Access. Submit the following job:

```
'sharedprefix.MICS.CNTL(DOCAGEN1)'
```

```
+---------------------------------------------------+
| Reference(s):  Section 3.3.7                      |
+---------------------------------------------------+
```

__ 14. Activate MWF

Activate the CA MICS Workstation Facility (MWF).

Read the referenced sections listed below to set up the CA MICS ISPF environment. Enter the CA MICS online environment and specify your site's parameters. Note, you may only enter the CA MICS Administrator Facility automatically the first time you enter the environment.

```
+---------------------------------------------------+
| Reference(s): Sections 3.3.6 and 4.4              |
+---------------------------------------------------+
__ 15. (OPTIONAL) Activate MWF User Authorization


       Activate the MWF User Authorization feature.

       If you do not wish to limit access to CA MICS, skip
       this step.

       +----------------------------------------------------+
       | Reference(s): Section 4.4.3                        |
       +----------------------------------------------------+


__ 16. Execute LDSASFLS


       Load the SAS Screen Files and CA MICS Tables Data
       Area.  Submit the following job:

         'sharedprefix.MICS.CNTL(LDSASFLS)'

       +----------------------------------------------------+
       | Reference(s): Section 3.3.4                        |
       +----------------------------------------------------+
```

3.8.1.3 Define and Generate Products

```
__  1. Review Data Element Option Indicators

      Review the default options for Essential and Derived
      data elements, and optionally, override complex-level
      defaults for the YEARS time-span, week start, and
      fiscal calendar definitions.  Changes to the option
      values are a major database tailoring action.
      Carefully read the referenced sections listed below
      before changing the option values.

      The complex definition options are located in
      sharedprefix.MICS.PARMS(CPLXDEF).

      +-----------------------------------------------------+
      | Reference(s):  Sections 2.3.1.8 and 3.4.2           |
      +-----------------------------------------------------+

__  2. (OPTIONAL) Specify Complex Computing System Parameter
       Definitions

      Specify the complex-level computing system parameter
      definitions.  Complex-level computing system
      definitions are located in
      sharedprefix.MICS.PARMS(CPLXSID).  Carefully read the
      referenced section listed below before specifying the
      option values.

      +-----------------------------------------------------+
      | Reference(s):  Section 2.3.1.9                      |
      +-----------------------------------------------------+

__  3. (OPTIONAL) Specify Complex Time Zone Definition

      Specify the complex-level time zone definition.  The
      complex-level time zone specification is located in
      sharedprefix.MICS.PARMS(CPLXZONE).  Carefully read the
      referenced section listed below before specifying the
      option values.

      +-----------------------------------------------------+
      | Reference(s):  Section 2.3.1.10                     |
      +-----------------------------------------------------+

__  4. Execute CPLXGEN

      Run the CPLXGEN job to generate complex option
      definition global switches, system and time zone
      definitions, default YEARS time-span, week start, and
      fiscal calendar parameters.  Submit the following job:

        'sharedprefix.MICS.CNTL(CPLXGEN)'

      +----------------------------------------------------+
      | Reference(s):  Section 2.3.1.8 (CPLXDEF)           |
      |                Section 2.3.1.9 (CPLXSID)           |
      |                Section 2.3.1.10(CPLXZONE)          |
      |                Section 3.4.2 (cccGENIN)            |
      +----------------------------------------------------+

__  5. Define Products to be Generated

      Check off the products that require generation:

      Note: CAP, PER, STG, LDE, VCC, and VMT do not require
      generation.

      Standard Products

      ___ BAS - CA MICS base platform
      ___ ACT - Accounting and Chargeback Option
      ___ AST - CA ASTEX Option
      ___ AUM - Analyzer Option for MeasureWare
      ___ CIC - Analyzer Option for CICS
      ___ DB2 - Analyzer Option for DB2
      ___ DEX - Analyzer Option for VAX/VMS
      ___ HSM - VCA support for DFHSM
      ___ IDM - Analyzer Option for CA IDMS
      ___ IMS - Analyzer Option for IMS
      ___ MQS - Analyzer Option for MQSeries
      ___ PWR - Analyzer Option for VSE/Power
      ___ RMF - Hardware and SCP Analyzer
      ___ SMF - Batch and Operations Analyzer
      ___ SNT - Network Analyzer Option
      ___ SRL - System Reliability Analyzer Option
      ___ TDM - Tandem Option
      ___ TLM - Tape Analyzer Option, TLM Component
      ___ TSO - Analyzer for TSO
      ___ VCA - Space Analyzer Option
      ___ VMC - Analyzer Option for VM/CMS
      ___ VTS - Tape Analyzer Option, VTS Component
      ___ WEB - Web Analyzer Option

      Field Developed Applications (Including MSI and MTI)

      ___ (___) - ____________________________
      ___ (___) - ____________________________
      ___ (___) - ____________________________

      +---------------------------------------------------+
      | Reference(s):  Section 3.4.1                      |
      +---------------------------------------------------+


************************************************************
*  Repeat Steps 6 through 9 in this section for all        *
*  products that have been checked off for generation.     *
************************************************************

__  6. Review Component Generation Options

      Review component generation options in Chapter 7 of
      the product guides for the data integration
      applications (DIAs) you are installing.  Some products
      provide global element tailoring options that you may
      wish to review.  Products that process user SMF data
      sources also provide an option for record selection.

      +---------------------------------------------------+
      |Reference(s):Section 3.4.2                         |
      |             Section 4.4                           |
      |             Section 4.2, System Modification Guide|
      |             Chapter 7, ccc Analyzer Guide(s)      |
      +---------------------------------------------------+

__  7. (CONDITIONAL) Copy Load Module for CICS Analyzer

      Copy the appropriate load modules for the CICS
      Analyzer to the sharedprefix.MICS.LOAD library based
      on your installation's needs.

      INPUT SOURCE                      Load Module
      ------------                  -----------
      The Monitor
          Versions 8.0/8.1, 2.0/2.1       $CRCPRS
          Versions 7.0/7.1                TMV611

      EPILOG 1000                         BGUDCMP

      +----------------------------------------------------+
      | Reference(s):  Section 6.1, CICS Analyzer Guide   |
      +----------------------------------------------------+

__  8. (CONDITIONAL) Define Appropriate PARMS Members


      Customize the PARMS members that define account
      codes, cost center codes, and the exit routines with
      which they are associated.

      For the ccc Analyzer and Other Standard Products
      (Including FDAs, MSI and MTI):

          cccACCT  - Account Code Specification
          cccACRT  - Account Code Exit Routine

      For Batch and Operations Analyzer (SMF):

          ACCOUNT  - Batch Accounting Structure
          ACCTRTE  - Batch Accounting Exit Routine
      For Accounting and Chargeback (ACT):

          COSTCTR  - Accounting Cost Center Structure
          COSTRTE  - Accounting Cost Center Exit Routine

      For VAX/VMS Analyzer (DEX):

          DEXACRTD - DEX Disk Usage Accounting Exit Routine
          +-----------------------------------------------------+
          | Reference(s):  Section 2.3.1.7, all subsections    |
          |                Section 4.1, Accounting and         |
          |                             Chargeback User Guide |
          |                Chapter 7, ccc Analyzer Guide(s)    |
          +-----------------------------------------------------+

    __  9. Execute cccCGEN for Each Product

          Generate the product.

          Run the component generation process for each product,
          where ccc is the component identifier.

          Submit the following job:

'           sharedprefix.MICS.CNTL(cccCGEN)'

          Ensure that there are no error messages in ISPLOG
          and MICSLOG; that MICSLOG contains the normal termination
          message, BAS60199I; and that the job completes with a
          condition code of zero.

          Check off each component generation as it is done,
          running the BASCGEN first:

             __ BASCGEN

             __ ACTCGEN __ ASTCGEN __ AUMCGEN __ CICCGEN
             __ DB2CGEN __ DEXCGEN __ HSMCGEN __ IDMCGEN
             __ IMSCGEN __ MQSCGEN __ PWRCGEN __ RMFCGEN
             __ SMFCGEN __ SNTCGEN __ SRLCGEN __ TLMCGEN
             __ TDMCGEN __ TSOCGEN __ VCACGEN __ VMCCGEN
             __ VTSCGEN __ WEBCGEN

             __ cccCGEN for FDA, MSI, and MTI (if applicable)


          +-----------------------------------------------------+
          | Reference(s):  Section 3.4.3                        |
          +-----------------------------------------------------+

    ************************************************************
    *  Repeat Steps 6 through 9 in this section for all        *
    *  products that have been checked off for generation.     *
    ************************************************************

    __ 10. Execute MDBGEN
```

Run the Meta Database Build process. Submit the following job:

```
'sharedprefix.MICS.CNTL(MDBGEN)'
```

__ 11. (CONDITIONAL) Activate MTI

If applicable, activate the CA MICS NSM Interface (MTI).

MTI, part of the CA MICS platform, provides technology and tools for creating a custom component to read, process, and analyze performance management data from CA NSM.

__ Generate the product. Submit the following job:

```
'sharedprefix.MICS.CNTL(MTICGEN)'
```

Ensure that there are no error messages in ISPLOG and MICSLOG, that MICSLOG contains the normal termination messages BAS20999I, and that the job completes with a condition code of zero.

__ Initialize the dictionary. Submit the following job:

```
'sharedprefix.MICS.CNTL(DICINIT)'
```

Ensure that there are no error messages in the MICSLOG, that the MICSLOG contains the normal termination message MTI02399I, and that the job completes with a condition code of zero.

```
+----------------------------------------------------+
|Reference(s): Chapter 8, System Modification Guide |
+----------------------------------------------------+
```

__ 12. Execute cccCGEN for FDAs

Generate Field Developed Applications (FDAs)

Run the component generation process for each FDA, where ccc is the component identifier.

Check off each component generation as it is done:

```
         User Components

         -- -------   -- -------   -- ------- -- -------
         -- -------   -- -------   -- ------- -- -------

      +-----------------------------------------------------+
      |Reference(s): Chapter 7, System Modification Guide |
      +-----------------------------------------------------+

******************** IMPORTANT NOTICE *********************
*                                                         *
*  The following products do not require generation.      *
*                                                         *
*  CA MICS Capacity Planner (CAP) - If you are licensed    *
*  for CAP, use Document Access to print the guide and     *
*  follow the instructions in Chapter 12.                  *
*                                                         *
*  CA MICS Space Collector (VCC) - If you are licensed     *
*  for VCC, use Document Access to print the guide and     *
*  follow the instructions in Chapter 4.                   *
*                                                         *
*  CA MICS IMS Log Data Extractor (LDE) - If you are       *
*  licensed for LDE, use Document Access to print the      *
*  guide and follow the instructions in Chapter 4.         *
*                                                         *
*  CA MICS Performance Manager (PER) - If you are          *
*  licensed for PER, use Document Access to print the      *
*  guide and follow the instructions in Chapter 1.         *
*                                                         *
*  CA MICS StorageMate (STG) - If you are licensed for     *
*  STG, use Document Access to print the guide and follow  *
*  the instructions in Appendix B.                         *
*                                                         *
*  CA MICS Data Transfer Option for VM/CMS (VMT) - If you  *
*  are licensed for VMT, use Document Access to print the  *
*  guide and follow the instructions in Chapters 7 and 8.  *
*                                                         *
***********************************************************
```

3.8.1.4 Generate the New Database Unit(s)

```
 ************************************************************
 *  Repeat the steps in this section for each new database  *
 *  unit in the CA MICS complex.                            *
 ************************************************************

__  1. Define Catalog Index Prefixes/Tapeprefixes

       Define the catalog index for the prefix and the
       tapeprefix.

       No action is required if the high-level node name
       has already been defined as an alias in the master
       catalog.

       +-----------------------------------------------------+
       | Reference(s): Section 3.5.1                         |
       +-----------------------------------------------------+


__  2. (OPTIONAL) Worksheet Preparation

       Complete the set of Installation Preparation
       Worksheets for the database unit being defined.

       +-----------------------------------------------------+
       | Reference(s): Section 2.3, all subsections          |
       |               Chapter 7, ccc Analyzer Guide(s)      |
       +-----------------------------------------------------+


__  3. Execute COPYLIBS

       Build the database unit libraries.

       __ Edit sharedprefix.MICS.CNTL(COPYLIBS) and fill in
          the parameters on the EXEC statement.  DO NOT enter
          parameter values on the PROC statement.

       __ Submit the following job:

           'sharedprefix.MICS.CNTL(COPYLIBS)'

       +-----------------------------------------------------+
       | Reference(s): Section 3.5.2                         |
       +-----------------------------------------------------+
```

__ 4. Define JCL Parameters

Define all of the members in the prefix.MICS.PARMS library that are required by JCLGEN:

```
prefix.MICS.PARMS(JCLDEF)
prefix.MICS.PARMS(USERJCL)
prefix.MICS.PARMS(INPUTRDR or INPUTccc) if an SMF
  source component will be installed in this database
  unit
prefix.MICS.PARMS(JCLNAMES) if the default PROC or
  data set names should not be used for this unit
```

```
+----------------------------------------------------+
| Reference(s): Section 2.3.3.2.1 (JCLDEF)           |
|               Section 2.3.3.2.2 (USERJCL)          |
|               Section 2.3.3.2.3 (INPUTRDR)         |
|               Section 2.3.3.3.2.3 (JCLNAMES)       |
|               Chapter 7, ccc Analyzer Guide(s)     |
+----------------------------------------------------+
```

__ 5. Verify JCL and CLIST Attributes

Verify the JCL and CLIST attributes for each CA MICS product by reviewing sharedprefix.MICS.PARMS(JCL$Iccc), where ccc is the three character component identifier. If changes are required, make the changes in prefix.MICS.PARMS(JCLINFO).

```
+----------------------------------------------------+
| Reference(s): Section 2.3.3.3.2.2 (JCLINFO)        |
+----------------------------------------------------+
```

__ 6. Generate Unit Installation Jobs

Generate the CA MICS Unit Installation Jobs.

__ Edit prefix.MICS.CNTL(JCLGENA). Enter the parameters on the EXEC statement and the data set name (DSN=) parameter on the //ULOCKIT DD statement.

__ Submit the following job:

```
'prefix.MICS.CNTL(JCLGENA)'
```

```
       +---------------------------------------------------+
       | Reference(s): Section 3.5.3.2                     |
       +---------------------------------------------------+


__  7. Review JCL in PROCJOB2

       Check the JCL generated by JCLGENA by examining
       the following members in prefix.MICS.CNTL:

         PROCJOB2   MICSNDB   MICSDU

       If the JCL and JES parameters do not meet your
       installation's standards, update the
       prefix.MICS.PARMS members JCLDEF or USERJCL and
       resubmit JCLGENA.

       +---------------------------------------------------+
       | Reference(s): Section 3.3.3                       |
       +---------------------------------------------------+


__  8. Execute PROCJOB2

       Move the first group of CA MICS cataloged JCL
       procedures to the production PROC library by
       submitting the following job:

         'prefix.MICS.CNTL(PROCJOB2)'

       +---------------------------------------------------+
       | Reference(s): Section 3.5.4                       |
       +---------------------------------------------------+


__  9. Execute COPYLIBU

       Move the CA MICS parameter library members that are
       dependent on the products installed in this database
       unit.  Submit the following job:

         'prefix.MICS.CNTL(COPYLIBU)'

       +---------------------------------------------------+
       | Reference(s): Section 3.5.2.2                     |
       +---------------------------------------------------+


__ 10. Define Required Parameters
```

```
Define the required PARMS members.

__ CA MICS base platform (BAS):

    'prefix.MICS.PARMS(SYSID)'
    'prefix.MICS.PARMS(ZONE)'
    'prefix.MICS.PARMS(SITE)'

Update the members in prefix.MICS.PARMS for the
appropriate products.

__ CA MICS Analyzer Option for MeasureWare (AUM):

    'prefix.MICS.PARMS(INPUTAUM)'
    'prefix.MICS.PARMS(AUMAURT)'
    'prefix.MICS.PARMS(AUMOPS)'

__ CA MICS Analyzer Option for VAX/VMS (DEX):

    'prefix.MICS.PARMS(INPUTDEX)'
    'prefix.MICS.PARMS(DEXOPS)'

__ CA MICS Analyzer Option for IMS (IMS):

    'prefix.MICS.PARMS(INPUTIMS)'
    'prefix.MICS.PARMS(IMSAURT)'
    'prefix.MICS.PARMS(IMSOPS)'
    'prefix.MICS.PARMS(IMSRLRT)'
    'prefix.MICS.PARMS(IMSSPND)'

__ CA MICS Analyzer Option for VSE/Power (PWR):

    'prefix.MICS.PARMS(INPUTPWR)'
    'prefix.MICS.PARMS(PWRCONFG)'
    'prefix.MICS.PARMS(PWRGROUP)'
    'prefix.MICS.PARMS(PWRGPRTE)'
    'prefix.MICS.PARMS(PWROPS)'
    'prefix.MICS.PARMS(PWRTNRTE)'

__ CA MICS System Reliability Analyzer Option (SRL):

    'prefix.MICS.PARMS(INPUTSRL)'
    'prefix.MICS.PARMS(SRLDEVS)'
    'prefix.MICS.PARMS(SRLOPS)'

__ CA MICS Tandem Option (TDM):

    'prefix.MICS.PARMS(INPUTTDM)'
    'prefix.MICS.PARMS(TDMOPS)'

__ CA MICS Tape Analyzer Option, TLM Component:

    'prefix.MICS.PARMS(INPUTTLM)'
    'prefix.MICS.PARMS(TLMOPS)'
    'prefix.MICS.PARMS(VOLGPRTE)'

__ Field Developed Applications (FDAs) for MSI:

    'prefix.MICS.PARMS(cccOPS)'

__ Field Developed Applications (FDAs) for MTI:

    'prefix.MICS.PARMS(cccOPS)'
    'prefix.MICS.PARMS(INPUTccc)'
    'prefix.MICS.PARMS(cccSYSID)'

__ CA MICS Analyzer Option for VM/CMS (VMC):

    'prefix.MICS.PARMS(INPUTVMC)'
    'prefix.MICS.PARMS(VMCOPS)'

****************************************************
* --------------INPUT SOURCE NOTE------------------*
*                                                  *
* The following components may use SMF data as an  *
* input source:                                    *
*        AST, CIC, DB2, HSM, IDM, MQS, RMF,        *
*        SMF, SNT, VTS, TSO, VCA, WEB, and         *
*        FDAs (if applicable)                      *
*                                                  *
* For the components listed above, you must choose *
* one of the alternative methods shown below.      *
* These specifications identify the data set names *
* containing the raw data that will be processed   *
* by the unit's DAILY update job.                  *
*                                                  *
*   (1) If DAYSMF ON is specified in               *
*       prefix.MICS.PARMS(JCLDEF), then you must   *
*       identify the input data set names in       *
*       prefix.MICS.PARMS(INPUTRDR).               *
*                                                  *
*   (2) If DAYSMF OFF is specified in              *
*       prefix.MICS.PARMS(JCLDEF) you must         *
*       identify the input data set names in       *
*       prefix.MICS.PARMS(INPUTccc) for each       *
*       component in the unit.                     *
*                                                  *
*   (3) If this component is in a unit by itself,  *
*       then you must identify the input data set  *
*       names in prefix.MICS.PARMS(INPUTRDR).      *
*                                                  *
****************************************************

__ CA MICS CA ASTEX Option (AST):

    'prefix.MICS.PARMS(INPUTAST)'
    'prefix.MICS.PARMS(ASTAURT)'
    'prefix.MICS.PARMS(ASTOPS)'

__ CA MICS Analyzer Option for CICS (CIC):

    'prefix.MICS.PARMS(INPUTCIC)'
    'prefix.MICS.PARMS(CICALONE)'
    'prefix.MICS.PARMS(CICAURT)'
    'prefix.MICS.PARMS(CICMSAC)'
    'prefix.MICS.PARMS(CICOPS)'
    'prefix.MICS.PARMS(CICRLRT)'
    'prefix.MICS.PARMS(CICTHRSH)'

__ CA MICS Analyzer Option for DB2 (DB2):

    'prefix.MICS.PARMS(INPUTDB2)'
    'prefix.MICS.PARMS(DB2OPS)'

__ CA MICS Analyzer Option for CA IDMS (IDM):

    'prefix.MICS.PARMS(INPUTIDM)'
    'prefix.MICS.PARMS(IDMAURT)'
    'prefix.MICS.PARMS(IDMOPS)'
    'prefix.MICS.PARMS(IDMRLRT)'
    'prefix.MICS.PARMS(IDMRPTS)'

__ CA MICS Analyzer Option for MQSeries (MQS):

    'prefix.MICS.PARMS(INPUTMQS)'
    'prefix.MICS.PARMS(MQSOPS)'

__ CA MICS Hardware and SCP Analyzer (RMF):

    'prefix.MICS.PARMS(INPUTRMF)'
    'prefix.MICS.PARMS(RMFOPS)'

__ CA MICS Batch and Operations Analyzer (SMF):

    'prefix.MICS.PARMS(INPUTSMF)'
    'prefix.MICS.PARMS(JOBGROUP)'
    'prefix.MICS.PARMS(JOBGPRTE)'
    'prefix.MICS.PARMS(PRINTDEF)'
    'prefix.MICS.PARMS(SMFOPS)'
    'prefix.MICS.PARMS(SMFUOPTS)'
    'prefix.MICS.PARMS(TURNRTE)'

__ CA MICS Tape Analyzer Option, VTS Component:

    'prefix.MICS.PARMS(INPUTVTS)'
    'prefix.MICS.PARMS(VTSOPS)'
    'prefix.MICS.PARMS(VTSCMPID)'

__ CA MICS Network Analyzer Option (SNT):

    'prefix.MICS.PARMS(INPUTSNT)'
    'prefix.MICS.PARMS(NETGROUP)'
    'prefix.MICS.PARMS(NETGPRT)'
    'prefix.MICS.PARMS(NETAPPL)'
    'prefix.MICS.PARMS(SNTCIDRT)'
    'prefix.MICS.PARMS(SNTNIDRT)'
    'prefix.MICS.PARMS(SNTOPS)'
    'prefix.MICS.PARMS(SNTPLURT)'
    'prefix.MICS.PARMS(SNTRPTD)'
    'prefix.MICS.PARMS(SNTRPTM)'

__ CA MICS Analyzer for TSO:

    'prefix.MICS.PARMS(INPUTTSO)'
    'prefix.MICS.PARMS(TSOAURT)'
    'prefix.MICS.PARMS(TSOOPS)'

__ CA MICS Space Analyzer Option (VCA):

    'prefix.MICS.PARMS(INPUTHSM)'
    'prefix.MICS.PARMS(INPUTVCA)'
    'prefix.MICS.PARMS(HSMOPS)'
    'prefix.MICS.PARMS(VCAOPS)'

__ CA MICS Web Analyzer Option (WEB):

    'prefix.MICS.PARMS(INPUTWEB)'
    'prefix.MICS.PARMS(WEBOPS)'
    'prefix.MICS.PARMS(WEBSLIST)'
    'prefix.MICS.PARMS(WLECONF)'
    'prefix.MICS.PARMS(WLESCRPT)'

 __ Field Developed Applications (FDAs) using SMF data
    as input source:
```

```
    'prefix.MICS.PARMS(INPUTccc)'

***IMPORTANT****IMPORTANT***IMPORTANT****IMPORTANT***
*                                                   *
*  FDAs may or may not use SMF data as an input     *
*  source.                                          *
*                                                   *
*  CA MICS does NOT provide an automated interface  *
*  to CA SMF Director from within an FDA.  This     *
*  means that CA MICS will NOT be able to read the  *
*  CA SMF Director duplicate split index and obtain *
*  the necessary input files.  However, any SMF data*
*  sets output from CA SMF Director may still be    *
*  directly input as standard SMF data.             *
*                                                   *
***IMPORTANT****IMPORTANT***IMPORTANT****IMPORTANT***

+---------------------------------------------------+
|  References:  Section 2.3.2, all subsections      |
|               Section 2.3.3.2, all subsections    |
|               Chapter 7, ccc Analyzer Guide(s)    |
|               Section 7.6, System Modification    |
|                 Guide (for MSI)                   |
|               Section 8.7, System Modification    |
|                 Guide (for MTI)                   |
|               Section 1.5.2.3, CA ASTEX Option    |
|                 Guide                             |
|               Section 7.2.2, Application          |
|                 Extension Option for Tandem Guide |
+---------------------------------------------------+
```

__ 11. (OPTIONAL) Update DBMODEL

Update prefix.MICS.PARMS(DBMODEL).

The COPYLIBU job generated the DBMODEL parameter library member. Review and modify the DBMODEL contents.

Change the retention limits on the FILE statements according to your retention needs.

```
+---------------------------------------------------+
| Reference(s): Section 2.3.4                       |
|               Chapter 7, System Modification      |
|                 Guide, for user components        |
+---------------------------------------------------+
```

__ 12. (OPTIONAL) Update COMPRESS

Update prefix.MICS.PARMS(COMPRESS).

The COPYLIBU job generated the COMPRESS parameter library member. Review and modify the COMPRESS contents.

```
+-----------------------------------------------------+
| Reference(s): Section 2.3.2.6                       |
+-----------------------------------------------------+
```

__ 13. Review UNITGEN Options

Review sharedprefix.MICS.PARMS(UNITGEN) for the options in effect.

```
+-----------------------------------------------------+
|  Reference(s):  Section 5.6.1                       |
+-----------------------------------------------------+
```

__ 14. Execute UNITGEN

Submit the following job:

```
  'prefix.MICS.CNTL(UNITGEN)'
```

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

If the UNITGEN NOAUTOSUBMIT, or UNITGEN AUTOSUBMIT and NOOPJCL parameters have been specified, UNITGEN will complete with a normal return code of 4. Otherwise, UNITGEN and UNITGEN1 should complete with a return code of 0.

__ 15. (CONDITIONAL) Execute UNITGEN1

Perform this step only if your installation has specified

UNITGEN NOAUTOSUBMIT

You must manually submit the UNITGEN1 job to perform

the required JCL and parameter generations.

After the UNITGEN job from the previous step completes successfully, submit the following job:

```
  'prefix.MICS.CNTL(UNITGEN1)'
```

__ 16. Verify UNITGEN1 Results

Verify that the UNITGEN1 job executed successfully.

The UNITGEN1 job contains multiple steps that perform COPYLIBU, CYCLEGEN, JCLGEN, cccPGEN, and GDGSGEN processing or part of it.

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

All steps should complete with a return code of zero.

If errors are encountered during UNITGEN1 job execution, correct the errors and rerun UNITGEN and UNITGEN1 jobs respectively.

```
+---------------------------------------------------+
|  Reference(s):  Sections 5.6 and 5.6.1            |
+---------------------------------------------------+
```

__ 17. Execute JCLGENC

Generate the CA MICS database set-up PROCS. Submit the following job:

```
  'prefix.MICS.CNTL(JCLGENC)'
```

```
+----------------------------------------------------+
| Reference(s): Section 3.5.3.3                        |
+----------------------------------------------------+
```

__ 18. Execute PROCJOB3

Move CA MICS database set-up PROCS to the installation PROCLIB by submitting the following job:

```
  'prefix.MICS.CNTL(PROCJOB3)'
```

```
       +---------------------------------------------------+
       | Reference(s): Section 3.5.4                       |
       +---------------------------------------------------+


__ 19. (CONDITIONAL) Execute JCLGEND

       Perform this step only if your installation has
       specified

       UNITGEN NOOPJCL

       If NOOPJCL is specified, UNITGEN1 did not generate the
       CA MICS operational jobs.

       To generate these jobs, submit the following job:

         'prefix.MICS.CNTL(JCLGEND)'

       Ensure that there are no error messages in MICSLOG or
       SYSTSPRT and that the MICSLOG contains the normal
       termination messages, BAS10999I or BAS11999I.

       All steps should complete with a return code of zero.

       +---------------------------------------------------+
       |  Reference(s):  Section 5.6.1                     |
       +---------------------------------------------------+


__ 20. Complete Accounting and Chargeback Processes

       If Accounting and Chargeback is being installed,
       define the parameters and execute the generation
       processes required for this product.  Use the
       checklists in the Accounting and Chargeback User
       Guide.  Once you have completed the checklist sections
       that deal with planning, installation, and activation;
       continue the installation process.

       +---------------------------------------------------+
       |Reference(s): Chapter 2, Accounting and Chargeback |
       |                User Guide                         |
       +---------------------------------------------------+


__ 21. (OPTIONAL) Estimate Database Space
```

Estimate the database space to allocate on disk.

Review the JCL in prefix.MICS.CNTL(DBSIZE). The input data sets should contain data of about one day.

__ Submit the following job:

```
    'prefix.MICS.CNTL(DBSIZE)'
```

```
+-----------------------------------------------------+
| Reference(s): Section 2.3.4.2                       |
+-----------------------------------------------------+
```

__ 22. Allocate Database Files

Allocate the Online Database and associated files.

__ Edit prefix.MICS.CNTL(ALLOCATE) and fill in the SPACE, VOLSER, and UNIT keyword values.

__ Submit the following job:

```
    'prefix.MICS.CNTL(ALLOCATE)'
```

```
+-----------------------------------------------------+
| Reference(s): Section 3.5.6.2                       |
+-----------------------------------------------------+
```

__ 23. Review EXECDEF

Review the default values for the dynamic execution options in prefix.MICS.PARMS(EXECDEF) and update as needed. Carefully review the entries affecting data retention limits.

```
+-----------------------------------------------------+
| References: Section 2.3.5                           |
|             Section A.3                             |
|             Section 7.3.6 Batch and Operations      |
|              Analyzer Guide                         |
+-----------------------------------------------------+
```

__ 24. Initialize Daily Audit Tape for Accounting and Chargeback

If Accounting and Chargeback is being installed in

this database unit, initialize the daily audit tape used for journal consolidation as the first tape in a generation data group series. Submit the following job:

```
  'prefix.MICS.CNTL(ACTINITA)'
```

__ 25. Incremental Update Preparation

If you activated an incremental update for one or more products in this unit database, run the cccIUALC job for each of these products. The cccIUALC job allocates the product's incremental update checkpoint and database files.

__ Edit the member according to your organization's data set allocation parameter standards and DASD space requirements. Submit the following job:

```
  'prefix.MICS.CNTL(cccIUALC)'
```

__ If you specified

```
  INCRDB TAPE
```

for one or more of these products, run the product's cccIUGDG job to create GDG indices for the incremental update tape database files. Submit the following job:

```
  'prefix.MICS.CNTL(cccIUGDG)'
```

```
+----------------------------------------------------+
| Reference(s): Chapter 7, ccc Analyzer Guide(s)     |
+----------------------------------------------------+
```

__ 26. Initialize Database Checkpoint File

Before performing the first database update, initialize the checkpoint file. Submit the following job:

```
  'prefix.MICS.CNTL(CKPTINIT)'
```

```
+----------------------------------------------------+
| Reference(s): Section 3.5.7                        |
+----------------------------------------------------+
```

__ 27. Review Generated Unit

The system is fully generated and ready for checkout. Make a final review of all definitions and JCL, applying any final user modifications.

```
+---------------------------------------------------+
| Reference(s): Section 3.5.7                       |
+---------------------------------------------------+
```

__ 28. Execute DAILY

Run the first daily process. Submit the following job:

```
  'prefix.MICS.CNTL(DAILY)'
```

or

Use the Operational Status and Tracking DAILY command in MWF to submit daily processing.

```
+---------------------------------------------------+
| Reference(s): Section 3.5.7 and Chapter 4         |
+---------------------------------------------------+
```

__ 29. (OPTIONAL) Verify Results in Database

Run a database check job for each product in the unit to verify that key database definitions have been correctly defined and implemented. Submit the following job:

```
  'prefix.MICS.CNTL(cccCHECK)'
```

```
+---------------------------------------------------+
| Reference(s): Section 3.5.7                       |
+---------------------------------------------------+
```

__ 30. (OPTIONAL) Ensure Backup and Restore Are Valid

Run the backup and restore jobs. Submit the following jobs:

```
         'prefix.MICS.CNTL(BACKUP)'
         'prefix.MICS.CNTL(RESTORE)'

       or

       Use the Operational Status and Tracking BACKUP and
       RESTORE commands in MWF to submit the processing.

       +---------------------------------------------------+
       | Reference(s): Section 3.5.7 and Chapter 4         |
       +---------------------------------------------------+


__ 31. Implement Production Operation

       If the database unit being installed is a production
       unit, implement production operation for CA MICS daily
       processing.

       +---------------------------------------------------+
       | Reference(s): Section 3.7.1                       |
       +---------------------------------------------------+

 ************************************************************
 *  Repeat the steps in this section for each new database  *
 *  unit in the complex.                                    *
 ************************************************************
```

3.8.1.5 Finalize the Database Complex

```
************************************************************
*  Complete this checklist only after all of the database  *
*  units in the complex have been installed.               *
************************************************************
```

__ 1. (CONDITIONAL) CAP, PER, or STG Clients Only

If your site is not licensed for CAP, PER, or STG skip this step.

Capacity Planner (CAP) - Use Document Access to print the guide and follow the instructions in Chapter 12.

Performance Manager (PER) - Use Document Access to print the guide and refer to Appendix C.

StorageMate (STG) - Use Document Access to print the guide and refer to Appendix C.

__ 2. Back Up Libraries

Since CA MICS does not provide a standard backup of libraries, back up the libraries below so that recovery can be accomplished.

Complex-level data sets:

```
   sharedprefix.MICS.ASM
   sharedprefix.MICS.BIN
   sharedprefix.MICS.CLIST
   sharedprefix.MICS.CNTL
   sharedprefix.MICS.DIC.TEXT
   sharedprefix.MICS.DOC.TEXT
   sharedprefix.MICS.GENLIB
   sharedprefix.MICS.HOLD.PARMS
   sharedprefix.MICS.HOLD.USER.SOURCE
   sharedprefix.MICS.INC.TEXT
   sharedprefix.MICS.INCLLIB
   sharedprefix.MICS.ISPHLIB
   sharedprefix.MICS.ISPMLIB
   sharedprefix.MICS.ISPPLIB
   sharedprefix.MICS.ISPSLIB
   sharedprefix.MICS.ISPTLIB
   sharedprefix.MICS.LIBLOAD.CNTL
   sharedprefix.MICS.LOAD
   sharedprefix.MICS.LOCALMOD.CNTL
   sharedprefix.MICS.MACAUTOS
   sharedprefix.MICS.MACLIB
   sharedprefix.MICS.MCOLIB
   sharedprefix.MICS.MWFPDS.DATA
   sharedprefix.MICS.MWFSAS.DATA
   sharedprefix.MICS.OBJ
   sharedprefix.MICS.PARMS
   sharedprefix.MICS.PROTOLIB
   sharedprefix.MICS.SASFLS
   sharedprefix.MICS.SCREENS
   sharedprefix.MICS.SOURCE
   sharedprefix.MICS.TABLES
   sharedprefix.MICS.TEST.INCLLIB
   sharedprefix.MICS.TEST.LOAD
   sharedprefix.MICS.TEST.SOURCE
   sharedprefix.MICS.USER.SOURCE
```

Database unit data sets:

```
   prefix.MICS.CNTL
   prefix.MICS.MUOLIB
   prefix.MICS.PARMS
   prefix.MICS.USER.SOURCE

+-----------------------------------------------------+
| Reference(s): Section 3.6.2                         |
+-----------------------------------------------------+
```

__ 3. Verify MICF Full Execution

Check MICF for full execution.

Enter the CA MICS ISPF environment using the method selected by your site (e.g., LIBDEF, CLIST, or logon PROC). Select the MICF option from the CA MICS Workstation Facility. Review any new inquiries and execute MICF features.

```
+-----------------------------------------------------+
|References: Chapter 2, MICF User Guide               |
|            Chapter 3, ccc Analyzer Guide(s)         |
|                        (REPORTS)                    |
+-----------------------------------------------------+
__  4. Print Guides for Products Other than Those Listed in
       Step 1.

       Print a set of CA MICS documentation guides.

       Enter Document Access from the CA MICS Workstation
       Facility.  Choose Document Print and select the guides
       you wish to print.  Once you have completed this
       checklist and generated your products, you will be
       able to print guides tailored to your site's
       definitions.

       +---------------------------------------------------+
       |References: Section 3.6.1                          |
       |            Chapter 6, Document Access Guide       |
       +---------------------------------------------------+


__  5. Verify Tape and DASD Management Systems


       Check that CA MICS has been properly established in
       your installation's tape and DASD management systems.

       +---------------------------------------------------+
       | Reference(s): Section 3.6.2                       |
       +---------------------------------------------------+


__  6. (OPTIONAL) Offsite Storage


       Arrange for offsite storage of CA MICS monthly backup
       tapes.

       +---------------------------------------------------+
       | Reference(s): Section 3.6.2                       |
       +---------------------------------------------------+


__  7. (CONDITIONAL) Read Instructions for CA MICS Data
       Transfer Utility

       If your site uses CA NSM and you will be integrating
       CA NSM data into your own custom MTI component, read
       the installation instructions for the CA MICS Data
       Transfer utility.
```

```
+-----------------------------------------------------+
| Reference(s): Section 4.7                           |
+-----------------------------------------------------+
```

3.8.2 Adding a Product to an Existing Database Unit

This checklist defines the tasks required to add a new product to an existing database unit. The checklist tasks are organized into the following groups:

1 - Define and Generate Product(s)
2 - Reconfigure the Existing Database Unit(s)
3 - Finalize the Reconfigured Database Complex

3.8.2.1 Define and Generate Product(s)

__ 1. Define Products to be Generated

Check off the products that require generation:

Note: CAP, PER, STG, LDE, VCC, and VMT do not require generation.

Standard Products

___ BAS - CA MICS base platform
___ ACT - Accounting and Chargeback Option
___ AST - CA ASTEX Option
___ AUM - Analyzer Option for MeasureWare
___ CIC - Analyzer Option for CICS
___ DB2 - Analyzer Option for DB2
___ DEX - Analyzer Option for VAX/VMS
___ HSM - VCA support for DFHSM
___ IDM - Analyzer Option for CA IDMS
___ IMS - Analyzer Option for IMS
___ MQS - Analyzer Option for MQSeries
___ PWR - Analyzer Option for VSE/Power
___ RMF - Hardware and SCP Analyzer
___ SMF - Batch and Operations Analyzer

```
       ___ SNT - Network Analyzer Option
       ___ SRL - System Reliability Analyzer Option
       ___ TDM - Tandem Option
       ___ TLM - Tape Analyzer Option, TLM Component
       ___ TSO - Analyzer for TSO
       ___ VCA - Space Analyzer Option
       ___ VMC - Analyzer Option for VM/CMS
       ___ VTS - Tape Analyzer Option, VTS Component
       ___ WEB - Web Analyzer Option

       Field Developed Applications (Including MSI and MTI)

       ___ (___) - ____________________________
       ___ (___) - ____________________________
       ___ (___) - ____________________________

       +---------------------------------------------------+
       | Reference(s):  Section 3.4.1                      |
       +---------------------------------------------------+

 ************************************************************
 *  Repeat Steps 2 through 6 in this section for all        *
 *  products that have been checked off for generation.     *
 ************************************************************

__  2. Review Component Generation Options

       Review component generation options in Chapter 7 of
       the product guides for the data integration
       applications (DIAs) you are installing.  Some products
       provide global element tailoring options that you may
       wish to review.  Products that process user SMF data
       sources also provide an option for record selection.

       +---------------------------------------------------+
       |Reference(s):Section 3.4.2                         |
       |             Section 4.4                           |
       |             Section 4.2, System Modification Guide|
       |             Chapter 7, ccc Analyzer Guide(s)      |
       +---------------------------------------------------+
__  3. (CONDITIONAL) Copy Load Module for CICS Analyzer

       Copy the appropriate load modules for the CICS
       Analyzer to the sharedprefix.MICS.LOAD library based
       on your installation's needs.

       INPUT SOURCE                      Load Module
       ------------                      -----------
       The Monitor
           Versions 8.0/8.1, 2.0/2.1       $CRCPRS
           Versions 7.0/7.1                TMV611

       EPILOG 1000                         BGUDCMP

       +-----------------------------------------------------+
       | Reference(s):  Section 6.1, CICS Analyzer Guide    |
       +-----------------------------------------------------+

__  4. (CONDITIONAL) Define Appropriate PARMS Members

       Customize the PARMS members that define account
       codes, cost center codes, and the exit routines with
       which they are associated.

       For the ccc Analyzer and Other Standard Products
       (Including FDAs, MSI and MTI):

           cccACCT  - Account Code Specification
           cccACRT  - Account Code Exit Routine

       For Batch and Operations Analyzer (SMF):

           ACCOUNT  - Batch Accounting Structure
           ACCTRTE  - Batch Accounting Exit Routine
       For Accounting and Chargeback (ACT):

           COSTCTR  - Accounting Cost Center Structure
           COSTRTE  - Accounting Cost Center Exit Routine

       For VAX/VMS Analyzer (DEX):

           DEXACRTD - DEX Disk Usage Accounting Exit Routine
```

```
+-----------------------------------------------------+
| Reference(s):  Section 2.3.1.7, all subsections    |
|                Section 4.1, Accounting and         |
|                             Chargeback User Guide |
|                Chapter 7, ccc Analyzer Guide(s)    |
+-----------------------------------------------------+
```

Customize the PARMS members that define account codes, cost center codes, and the exit routines with which they are associated.

For the ccc Analyzer and Other Standard Products (Including FDAs, MSI and MTI):

```
    cccACCT  - Account Code Specification
    cccACRT  - Account Code Exit Routine
```

For Batch and Operations Analyzer (SMF):

```
    ACCOUNT  - Batch Accounting Structure
    ACCTRTE  - Batch Accounting Exit Routine
```

For Accounting and Chargeback (ACT):

```
    COSTCTR  - Accounting Cost Center Structure
    COSTRTE  - Accounting Cost Center Exit Routine
```

For VAX/VMS Analyzer (DEX):

```
    DEXACRTD - DEX Disk Usage Accounting Exit Routine
```

```
+-----------------------------------------------------+
| Reference(s):  Section 2.3.1.7, all subsections    |
|                Section 4.1, Accounting and         |
|                             Chargeback User Guide |
|                Chapter 7, ccc Analyzer Guide(s)    |
+-----------------------------------------------------+
```

__   5. (CONDITIONAL) Execute JCLGENUC for MTI Components

During the creation of an MTI component, sharedprefix.MICS.PARMS(cccINPUT) is generated. This member contains the data sets that will be used during component creation to determine which CA NSM performance management resources and resource-subtypes will be included in the component.

Ensure that this member contains the input data sets for this component.

If changes were made to cccINPUT, then

__ Edit sharedprefix.MICS.PARMS(JCLGENUC) and copy the
   contents of sharedprefix.MICS.PARMS(JCL$0ccc) into
   this member (JCLGENUC).

__ Submit the following job:

```
     'sharedprefix.MICS.CNTL(JCLGENUC)'
```

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.

```
+----------------------------------------------------+
|Reference(s):Section 8.5, System Modification Guide|
+----------------------------------------------------+
```

During the creation of an MTI component, sharedprefix.MICS.PARMS(cccINPUT) is generated. This member contains the data sets that will be used during component creation to determine which CA NSM performance management resources and resource-subtypes will be included in the component.

Ensure that this member contains the input data sets for this component.

If changes were made to cccINPUT, then

__ Edit sharedprefix.MICS.PARMS(JCLGENUC) and copy the
   contents of sharedprefix.MICS.PARMS(JCL$0ccc) into
   this member (JCLGENUC).

__ Submit the following job:

```
     'sharedprefix.MICS.CNTL(JCLGENUC)'
```

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.

```
+----------------------------------------------------+
|Reference(s):Section 8.5, System Modification Guide|
+----------------------------------------------------+
```

During the creation of an MTI component, sharedprefix.MICS.PARMS(cccINPUT) is generated. This member contains the data sets that will be used during component creation to determine which CA NSM performance management resources and resource-subtypes will be included in the component.

Ensure that this member contains the input data sets for this component.
If changes were made to cccINPUT, then

__ Edit sharedprefix.MICS.PARMS(JCLGENUC) and copy the
   contents of sharedprefix.MICS.PARMS(JCL$0ccc) into
   this member (JCLGENUC).

__ Submit the following job:

```
     'sharedprefix.MICS.CNTL(JCLGENUC)'
```

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.

```
+---------------------------------------------------+
|Reference(s):Section 8.5, System Modification Guide|
+---------------------------------------------------+
__  6.  Execute cccCGEN for Each Product

       Generate the product.

       Run the component generation process for each product,
       where ccc is the component identifier.

       Submit the following job:

'        sharedprefix.MICS.CNTL(cccCGEN)'

       Ensure that there are no error messages in ISPLOG
       and MICSLOG; that MICSLOG contains the normal termination
       message, BAS60199I; and that the job completes with a
       condition code of zero.

       Check off each component generation as it is done,
       running the BASCGEN first:

          __ BASCGEN

          __ ACTCGEN __ ASTCGEN __ AUMCGEN __ CICCGEN
          __ DB2CGEN __ DEXCGEN __ HSMCGEN __ IDMCGEN
          __ IMSCGEN __ MQSCGEN __ PWRCGEN __ RMFCGEN
          __ SMFCGEN __ SNTCGEN __ SRLCGEN __ TLMCGEN
          __ TDMCGEN __ TSOCGEN __ VCACGEN __ VMCCGEN
          __ VTSCGEN __ WEBCGEN

          __ cccCGEN for FDA, MSI, and MTI (if applicable)


       +-----------------------------------------------------+
       | Reference(s):  Section 3.4.3                        |
       +-----------------------------------------------------+

           MTI Components

           ___ (___) - ____________________________
           ___ (___) - ____________________________

 ************************************************************
 *  Repeat Steps 2 through 6 in this section for all        *
 *  products that have been checked off for generation.     *
 ************************************************************
```

3.8.2.2 Reconfigure the Existing Database Unit(s)

```
 *************************************************************
 *  Repeat the steps in this section for each database unit *
 *  in the CA MICS complex that will contain the new        *
 *  product.                                                *
 *************************************************************

__  1. Suspend Operation

       Suspend production operations for the unit database by
       using the Operational Status and Tracking SUSPEND
       command in MWF to deactivate production scheduling.

__  2. (OPTIONAL) Worksheet Preparation

       Complete the set of Installation Preparation
       Worksheets for the database unit being defined.

       +----------------------------------------------------+
       | Reference(s): Section 2.3, all subsections         |
       |               Chapter 7, ccc Analyzer Guide(s)     |
       +----------------------------------------------------+

__  3. Define JCL Parameters


       Define all of the members in the prefix.MICS.PARMS
       library that are required by JCLGEN:

       prefix.MICS.PARMS(JCLDEF)
       prefix.MICS.PARMS(USERJCL)
       prefix.MICS.PARMS(INPUTRDR or INPUTccc) if an SMF
         source component will be installed in this database
         unit
       prefix.MICS.PARMS(JCLNAMES) if the default PROC or
         data set names should not be used for this unit

       +----------------------------------------------------+
       | Reference(s): Section 2.3.3.2.1 (JCLDEF)           |
       |               Section 2.3.3.2.2 (USERJCL)          |
       |               Section 2.3.3.2.3 (INPUTRDR)         |
       |               Section 2.3.3.3.2.3 (JCLNAMES)       |
       |               Chapter 7, ccc Analyzer Guide(s)     |
       +----------------------------------------------------+


__  4. Verify JCL and Clist Attributes
```

Verify the JCL and CLIST attributes for each CA MICS product by reviewing sharedprefix.MICS.PARMS(JCL$Iccc), where ccc is the three character component identifier. If changes are required, make the changes in prefix.MICS.PARMS(JCLINFO).

```
+----------------------------------------------------+
| Reference(s): Section 2.3.3.3.2.2 (JCLINFO)        |
+----------------------------------------------------+
```

__ 5. Generate Unit Installation Jobs

Regenerate the CA MICS unit installation Jobs. Submit the following job:

```
  'prefix.MICS.CNTL(JCLGENB)'
```

```
+----------------------------------------------------+
| Reference(s): Section 3.5.3.3                      |
+----------------------------------------------------+
```

__ 6. Execute PROCJOB2

Move the first group of CA MICS cataloged JCL procedures to the production PROC library by submitting the following job:

```
  'prefix.MICS.CNTL(PROCJOB2)'
```

```
+----------------------------------------------------+
| Reference(s): Section 3.5.4                        |
+----------------------------------------------------+
```

__ 7. Execute COPYLIBU

Move the CA MICS parameter library members that are dependent on the products installed in this database unit. Submit the following job:

```
  'prefix.MICS.CNTL(COPYLIBU)'
```

```
+----------------------------------------------------+
| Reference(s): Section 3.5.2.2                      |
+----------------------------------------------------+
```

__ 8. Define Required Parameters

Define the required PARMS members.

__ CA MICS base platform (BAS):

```
    'prefix.MICS.PARMS(SYSID)'
    'prefix.MICS.PARMS(ZONE)'
    'prefix.MICS.PARMS(SITE)'
```

Update the members in prefix.MICS.PARMS for the appropriate products.

__ CA MICS Analyzer Option for MeasureWare (AUM):

```
    'prefix.MICS.PARMS(INPUTAUM)'
    'prefix.MICS.PARMS(AUMAURT)'
    'prefix.MICS.PARMS(AUMOPS)'
```

__ CA MICS Analyzer Option for VAX/VMS (DEX):

```
    'prefix.MICS.PARMS(INPUTDEX)'
    'prefix.MICS.PARMS(DEXOPS)'
```

__ CA MICS Analyzer Option for IMS (IMS):

```
    'prefix.MICS.PARMS(INPUTIMS)'
    'prefix.MICS.PARMS(IMSAURT)'
    'prefix.MICS.PARMS(IMSOPS)'
    'prefix.MICS.PARMS(IMSRLRT)'
    'prefix.MICS.PARMS(IMSSPND)'
```

__ CA MICS Analyzer Option for VSE/Power (PWR):

```
    'prefix.MICS.PARMS(INPUTPWR)'
    'prefix.MICS.PARMS(PWRCONFG)'
    'prefix.MICS.PARMS(PWRGROUP)'
    'prefix.MICS.PARMS(PWRGPRTE)'
    'prefix.MICS.PARMS(PWROPS)'
    'prefix.MICS.PARMS(PWRTNRTE)'
```

__ CA MICS System Reliability Analyzer Option (SRL):

```
    'prefix.MICS.PARMS(INPUTSRL)'
    'prefix.MICS.PARMS(SRLDEVS)'
    'prefix.MICS.PARMS(SRLOPS)'
```

__ CA MICS Tandem Option (TDM):

```
    'prefix.MICS.PARMS(INPUTTDM)'
    'prefix.MICS.PARMS(TDMOPS)'
```

```
__ CA MICS Tape Analyzer Option, TLM Component:

    'prefix.MICS.PARMS(INPUTTLM)'
    'prefix.MICS.PARMS(TLMOPS)'
    'prefix.MICS.PARMS(VOLGPRTE)'

__ Field Developed Applications (FDAs) for MSI:

    'prefix.MICS.PARMS(cccOPS)'

__ Field Developed Applications (FDAs) for MTI:

    'prefix.MICS.PARMS(cccOPS)'
    'prefix.MICS.PARMS(INPUTccc)'
    'prefix.MICS.PARMS(cccSYSID)'

__ CA MICS Analyzer Option for VM/CMS (VMC):

    'prefix.MICS.PARMS(INPUTVMC)'
    'prefix.MICS.PARMS(VMCOPS)'

****************************************************
* --------------INPUT SOURCE NOTE------------------*
*                                                  *
* The following components may use SMF data as an  *
* input source:                                    *
*        AST, CIC, DB2, HSM, IDM, MQS, RMF,        *
*        SMF, SNT, VTS, TSO, VCA, WEB, and         *
*        FDAs (if applicable)                      *
*                                                  *
* For the components listed above, you must choose *
* one of the alternative methods shown below.      *
* These specifications identify the data set names *
* containing the raw data that will be processed   *
* by the unit's DAILY update job.                  *
*                                                  *
*   (1) If DAYSMF ON is specified in               *
*       prefix.MICS.PARMS(JCLDEF), then you must   *
*       identify the input data set names in       *
*       prefix.MICS.PARMS(INPUTRDR).               *
*                                                  *
*   (2) If DAYSMF OFF is specified in              *
*       prefix.MICS.PARMS(JCLDEF) you must         *
*       identify the input data set names in       *
*       prefix.MICS.PARMS(INPUTccc) for each       *
*       component in the unit.                     *
*                                                  *
*   (3) If this component is in a unit by itself,  *
*       then you must identify the input data set  *
*       names in prefix.MICS.PARMS(INPUTRDR).      *
*                                                  *
****************************************************

__ CA MICS CA ASTEX Option (AST):

    'prefix.MICS.PARMS(INPUTAST)'
    'prefix.MICS.PARMS(ASTAURT)'
    'prefix.MICS.PARMS(ASTOPS)'

__ CA MICS Analyzer Option for CICS (CIC):

    'prefix.MICS.PARMS(INPUTCIC)'
    'prefix.MICS.PARMS(CICALONE)'
    'prefix.MICS.PARMS(CICAURT)'
    'prefix.MICS.PARMS(CICMSAC)'
    'prefix.MICS.PARMS(CICOPS)'
    'prefix.MICS.PARMS(CICRLRT)'
    'prefix.MICS.PARMS(CICTHRSH)'

__ CA MICS Analyzer Option for DB2 (DB2):

    'prefix.MICS.PARMS(INPUTDB2)'
    'prefix.MICS.PARMS(DB2OPS)'

__ CA MICS Analyzer Option for CA IDMS (IDM):

    'prefix.MICS.PARMS(INPUTIDM)'
    'prefix.MICS.PARMS(IDMAURT)'
    'prefix.MICS.PARMS(IDMOPS)'
    'prefix.MICS.PARMS(IDMRLRT)'
    'prefix.MICS.PARMS(IDMRPTS)'

__ CA MICS Analyzer Option for MQSeries (MQS):

    'prefix.MICS.PARMS(INPUTMQS)'
    'prefix.MICS.PARMS(MQSOPS)'

__ CA MICS Hardware and SCP Analyzer (RMF):

    'prefix.MICS.PARMS(INPUTRMF)'
    'prefix.MICS.PARMS(RMFOPS)'

__ CA MICS Batch and Operations Analyzer (SMF):

    'prefix.MICS.PARMS(INPUTSMF)'
    'prefix.MICS.PARMS(JOBGROUP)'
    'prefix.MICS.PARMS(JOBGPRTE)'
    'prefix.MICS.PARMS(PRINTDEF)'
    'prefix.MICS.PARMS(SMFOPS)'
    'prefix.MICS.PARMS(SMFUOPTS)'
    'prefix.MICS.PARMS(TURNRTE)'

__ CA MICS Tape Analyzer Option, VTS Component:

    'prefix.MICS.PARMS(INPUTVTS)'
    'prefix.MICS.PARMS(VTSOPS)'
    'prefix.MICS.PARMS(VTSCMPID)'

__ CA MICS Network Analyzer Option (SNT):

    'prefix.MICS.PARMS(INPUTSNT)'
    'prefix.MICS.PARMS(NETGROUP)'
    'prefix.MICS.PARMS(NETGPRT)'
    'prefix.MICS.PARMS(NETAPPL)'
    'prefix.MICS.PARMS(SNTCIDRT)'
    'prefix.MICS.PARMS(SNTNIDRT)'
    'prefix.MICS.PARMS(SNTOPS)'
    'prefix.MICS.PARMS(SNTPLURT)'
    'prefix.MICS.PARMS(SNTRPTD)'
    'prefix.MICS.PARMS(SNTRPTM)'

__ CA MICS Analyzer for TSO:

    'prefix.MICS.PARMS(INPUTTSO)'
    'prefix.MICS.PARMS(TSOAURT)'
    'prefix.MICS.PARMS(TSOOPS)'

__ CA MICS Space Analyzer Option (VCA):

    'prefix.MICS.PARMS(INPUTHSM)'
    'prefix.MICS.PARMS(INPUTVCA)'
    'prefix.MICS.PARMS(HSMOPS)'
    'prefix.MICS.PARMS(VCAOPS)'

__ CA MICS Web Analyzer Option (WEB):

    'prefix.MICS.PARMS(INPUTWEB)'
    'prefix.MICS.PARMS(WEBOPS)'
    'prefix.MICS.PARMS(WEBSLIST)'
    'prefix.MICS.PARMS(WLECONF)'
    'prefix.MICS.PARMS(WLESCRPT)'

 __ Field Developed Applications (FDAs) using SMF data
    as input source:

     'prefix.MICS.PARMS(INPUTccc)'
```

```
***IMPORTANT****IMPORTANT***IMPORTANT****IMPORTANT***
*                                                   *
*  FDAs may or may not use SMF data as an input     *
*  source.                                          *
*                                                   *
*  CA MICS does NOT provide an automated interface  *
*  to CA SMF Director from within an FDA.  This     *
*  means that CA MICS will NOT be able to read the  *
*  CA SMF Director duplicate split index and obtain *
*  the necessary input files.  However, any SMF data*
*  sets output from CA SMF Director may still be    *
*  directly input as standard SMF data.             *
*                                                   *
***IMPORTANT****IMPORTANT***IMPORTANT****IMPORTANT***

+---------------------------------------------------+
|  References:  Section 2.3.2, all subsections      |
|               Section 2.3.3.2, all subsections    |
|               Chapter 7, ccc Analyzer Guide(s)    |
|               Section 7.6, System Modification    |
|                 Guide (for MSI)                   |
|               Section 8.7, System Modification    |
|                 Guide (for MTI)                   |
|               Section 1.5.2.3, CA ASTEX Option    |
|                 Guide                             |
|               Section 7.2.2, Application          |
|                 Extension Option for Tandem Guide |
+---------------------------------------------------+
```

__ 9. (OPTIONAL) Update DBMODEL

Update prefix.MICS.PARMS(DBMODEL).

The COPYLIBU job generated the DBMODEL parameter library member. Review and modify the DBMODEL contents.

Change the retention limits on the FILE statements according to your retention needs.

```
+---------------------------------------------------+
| Reference(s): Section 2.3.4                       |
|               Chapter 7, System Modification      |
|                 Guide, for user components        |
+---------------------------------------------------+
```

__ 10. (OPTIONAL) Update COMPRESS

Update prefix.MICS.PARMS(COMPRESS).

```
               The COPYLIBU job generated the COMPRESS parameter
               library member.  Review and modify the COMPRESS
               contents.

               +-----------------------------------------------------+
               | Reference(s): Section 2.3.2.6                       |
               +-----------------------------------------------------+

        __ 11. Review EXECDEF

               Review the default values for the dynamic execution
               options in prefix.MICS.PARMS(EXECDEF) and update as
               needed.  Carefully review the entries affecting data
               retention limits.

               +-----------------------------------------------------+
               | References: Section 2.3.5                           |
               |             Section A.3                             |
               |             Section 7.3.6 Batch and Operations      |
               |              Analyzer Guide                         |
               +-----------------------------------------------------+

        __ 12. Review UNITGEN Options

                Review sharedprefix.MICS.PARMS(UNITGEN) for the
                options in effect.

                +-----------------------------------------------------+
                |  Reference(s):  Section 5.6.1                       |
                +-----------------------------------------------------+

        __ 13. Execute UNITGEN

               Submit the following job:

                 'prefix.MICS.CNTL(UNITGEN)'

               Ensure that there are no error messages in MICSLOG or
               SYSTSPRT and that the MICSLOG contains the normal
               termination messages, BAS10999I or BAS11999I.

               If the UNITGEN NOAUTOSUBMIT, or UNITGEN AUTOSUBMIT and
               NOOPJCL parameters have been specified, UNITGEN will
               complete with a normal return code of 4.  Otherwise,
               UNITGEN and UNITGEN1 should complete with a return
               code of 0.

        __ 14. (CONDITIONAL) Execute UNITGEN1
```

Perform this step only if your installation has specified

UNITGEN NOAUTOSUBMIT

You must manually submit the UNITGEN1 job to perform the required JCL and parameter generations.

After the UNITGEN job from the previous step completes successfully, submit the following job:

```
  'prefix.MICS.CNTL(UNITGEN1)'
```

__ 15. Verify UNITGEN1 Results

Verify that the UNITGEN1 job executed successfully.

The UNITGEN1 job contains multiple steps that perform COPYLIBU, CYCLEGEN, JCLGEN, cccPGEN, and GDGSGEN processing or part of it.

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

All steps should complete with a return code of zero.

If errors are encountered during UNITGEN1 job execution, correct the errors and rerun UNITGEN and UNITGEN1 jobs respectively.

```
+--------------------------------------------------+
|  Reference(s):  Sections 5.6 and 5.6.1           |
+--------------------------------------------------+
```

__ 16. Execute PROCJOB3

Move CA MICS database set-up PROCS to the installation PROCLIB by submitting the following job:

```
  'prefix.MICS.CNTL(PROCJOB3)'
```

```
+---------------------------------------------------+
| Reference(s): Section 3.5.4                       |
+---------------------------------------------------+
```

__ 17. (CONDITIONAL) Execute JCLGEND

Perform this step only if your installation has
specified

UNITGEN NOOPJCL

If NOOPJCL is specified, UNITGEN1 did not generate the
CA MICS operational jobs.

To generate these jobs, submit the following job:

```
  'prefix.MICS.CNTL(JCLGEND)'
```

Ensure that there are no error messages in MICSLOG or
SYSTSPRT and that the MICSLOG contains the normal
termination messages, BAS10999I or BAS11999I.

All steps should complete with a return code of zero.

```
+---------------------------------------------------+
|  Reference(s):  Section 5.6.1                     |
+---------------------------------------------------+
```

__ 18. Complete Accounting and Chargeback Processes

If Accounting and Chargeback is being installed,
define the parameters and execute the generation
processes required for this product. Use the
checklists in the Accounting and Chargeback User
Guide. Once you have completed the checklist sections
that deal with planning, installation, and activation;
continue the installation process.

```
+---------------------------------------------------+
|Reference(s): Chapter 2, Accounting and Chargeback |
|                User Guide                         |
+---------------------------------------------------+
```

__ 19. (OPTIONAL) Estimate Database Space

Estimate the database space to allocate on disk.

Review the JCL in prefix.MICS.CNTL(DBSIZE). The input
data sets should contain data of about one day.

__ Submit the following job:

```
    'prefix.MICS.CNTL(DBSIZE)'
```

```
+---------------------------------------------------+
       | Reference(s): Section 2.3.4.2                     |
       +---------------------------------------------------+

__ 20. Back Up Existing Database Unit

       Use the Operational Status and Tracking RESUME,
       BACKUP, and SUSPEND commands in MWF to enable, backup,
       and block CA MICS operational jobs respectively.

       Ensure that the backup job completed with a condition
       code of zero and the data sets were correctly
       cataloged.

       +---------------------------------------------------+
       | Reference(s): Section 4.3.4.4                     |
       +---------------------------------------------------+

__ 21. (CONDITIONAL) Rename Existing Unit Databases

       If you have estimated that your database files will
       require additional space, execute this step and steps
       22, 24, and 25 below.  If your database files should
       not require addition space, these step should be
       omitted.

       Rename or delete the database files.  We recommend
       renaming the files unless you do not have adequate
       disk space.  If renamed, these files may be deleted
       once you are sure that the database unit has been
       correctly allocated and restored.  Delete the data
       sets listed above if you do not have sufficient DASD
       space.

       CURRENT NAME                RENAME
       ------------                ------
       'prefix.MICS.DETAIL'------->'prefix.MICS.XDETAIL'
       'prefix.MICS.DAYS'--------->'prefix.MICS.XDAYS'
       'prefix.MICS.WEEKS'-------->'prefix.MICS.XWEEKS'
       'prefix.MICS.MONTHS'------->'prefix.MICS.XMONTHS'
       'prefix.MICS.YEARS'-------->'prefix.MICS.XYEARS'
       'prefix.MICS.CHECKPT.DATA'
                           ----->'prefix.MICS.XCHECKPT.DATA'

       'prefix.MICS.RESTART.CNTL'
                           ----->'prefix.MICS.XRESTART.CNTL'

       ******************************************************
       * Do NOT rename or delete sharedprefix.MICS.TABLES.  *
       ******************************************************
```

```
__ 22. (CONDITIONAL) Allocate Database Files

       Allocate the Online Database and associated files.

       __ Edit prefix.MICS.CNTL(ALLOCATE) and fill in the
          SPACE, VOLSER, and UNIT keyword values.

       __ Submit the following job:

           'prefix.MICS.CNTL(ALLOCATE)'

       +----------------------------------------------------+
       | Reference(s): Section 3.5.6.2                      |
       +----------------------------------------------------+

__ 23. Incremental Update Preparation

       If you activated an incremental update for one or more
       products in this unit database, run the cccIUALC job
       for each of these products.  The cccIUALC job
       allocates the product's incremental update checkpoint
       and database files.

       __ Edit the member according to your organization's
          data set allocation parameter standards and DASD
          space requirements.  Submit the following job:

            'prefix.MICS.CNTL(cccIUALC)'

       __ If you specified

            INCRDB TAPE

          for one or more of these products, run the
          product's cccIUGDG job to create GDG indices for
          the incremental update tape database files.  Submit
          the following job:

            'prefix.MICS.CNTL(cccIUGDG)'

       +----------------------------------------------------+
       | Reference(s): Chapter 7, ccc Analyzer Guide(s)     |
       +----------------------------------------------------+

__ 24. (CONDITIONAL) Initialize Database Checkpoint File

       Before performing the first database update,
       initialize the checkpoint file.  Submit the following
```

job:

```
  'prefix.MICS.CNTL(CKPTINIT)'
```

```
+-----------------------------------------------------+
| Reference(s): Section 3.5.7                         |
+-----------------------------------------------------+
```

__ 25. (CONDITIONAL) Restore the Database

Use the Operational Status and Tracking RESUME, RESTORE, and SUSPEND commands in MWF to enable, restore, and block CA MICS operational jobs respectively.

```
+-----------------------------------------------------+
| Reference(s): Section 4.3.4.4                       |
+-----------------------------------------------------+
```

__ 26. Initialize Daily Audit Tape for Accounting and Chargeback

If Accounting and Chargeback is being installed in this database unit, initialize the daily audit tape used for journal consolidation as the first tape in a generation data group series. Submit the following job:

```
  'prefix.MICS.CNTL(ACTINITA)'
```

__ 27. Review Generated Unit

The system is fully generated and ready for checkout. Make a final review of all definitions and JCL, applying any final user modifications.

```
+-----------------------------------------------------+
| Reference(s): Section 3.5.7                         |
+-----------------------------------------------------+
```

__ 28. Resume Operations

Enable CA MICS operational jobs for this unit database. Use the Operational Status and Tracking RESUME command in MWF to enable CA MICS operational jobs.

__ 29. Resume Daily Processing

```
       Use the Operational Status and Tracking DAILY command
       in MWF to resume daily processing.

       +---------------------------------------------------+
       | Reference(s): Section 3.5.7                       |
       +---------------------------------------------------+

__ 30. (OPTIONAL) Verify Results in Database

       Run a database check job for each product in the unit
       to verify that key database definitions have been
       correctly defined and implemented.  Submit the
       following job:

         'prefix.MICS.CNTL(cccCHECK)'

       +---------------------------------------------------+
       | Reference(s): Section 3.5.7                       |
       +---------------------------------------------------+

__ 31. Move to Production

       Reactivate production scheduling of CA MICS processing
       for this unit database.


 ************************************************************
 *  Repeat the steps in this section for each database unit *
 *  in the CA MICS complex that will contain the new        *
 *  product.                                                *
 ************************************************************
```

3.8.2.3 Finalize the Reconfigured Database Complex

```
 ************************************************************
 *  The checklist in this section should only be completed  *
 *  after all database units have been reconfigured in      *
 *  the CA MICS complex.                                    *
 ************************************************************

__  1. Back Up Libraries

       Since CA MICS does not provide a standard backup of
       libraries, back up the libraries below so that
       recovery can be accomplished.

       Complex-level data sets:

          sharedprefix.MICS.ASM
          sharedprefix.MICS.BIN
          sharedprefix.MICS.CLIST
          sharedprefix.MICS.CNTL
          sharedprefix.MICS.DIC.TEXT
          sharedprefix.MICS.DOC.TEXT
          sharedprefix.MICS.GENLIB
          sharedprefix.MICS.HOLD.PARMS
          sharedprefix.MICS.HOLD.USER.SOURCE
          sharedprefix.MICS.INC.TEXT
          sharedprefix.MICS.INCLLIB
          sharedprefix.MICS.ISPHLIB
          sharedprefix.MICS.ISPMLIB
          sharedprefix.MICS.ISPPLIB
          sharedprefix.MICS.ISPSLIB
          sharedprefix.MICS.ISPTLIB
          sharedprefix.MICS.LIBLOAD.CNTL
          sharedprefix.MICS.LOAD
          sharedprefix.MICS.LOCALMOD.CNTL
          sharedprefix.MICS.MACAUTOS
          sharedprefix.MICS.MACLIB
          sharedprefix.MICS.MCOLIB
          sharedprefix.MICS.MWFPDS.DATA
          sharedprefix.MICS.MWFSAS.DATA
          sharedprefix.MICS.OBJ
          sharedprefix.MICS.PARMS
          sharedprefix.MICS.PROTOLIB
          sharedprefix.MICS.SASFLS
          sharedprefix.MICS.SCREENS
          sharedprefix.MICS.SOURCE
          sharedprefix.MICS.TABLES
          sharedprefix.MICS.TEST.INCLLIB
          sharedprefix.MICS.TEST.LOAD
```

```
   sharedprefix.MICS.TEST.SOURCE
   sharedprefix.MICS.USER.SOURCE
```

Database unit data sets:

```
   prefix.MICS.CNTL
   prefix.MICS.MUOLIB
   prefix.MICS.PARMS
   prefix.MICS.USER.SOURCE
```

```
+---------------------------------------------------+
| Reference(s): Section 3.6.2                       |
+---------------------------------------------------+
```

__ 2. Verify MICF Full Execution

Check MICF for full execution.

Enter the CA MICS ISPF environment using the method selected by your site (e.g., LIBDEF, CLIST, or logon PROC). Select the MICF option from the CA MICS Workstation Facility. Review any new inquiries and execute MICF features.

```
+---------------------------------------------------+
|References: Chapter 2, MICF User Guide             |
|            Chapter 3, ccc Analyzer Guide(s)       |
|                        (REPORTS)                  |
+---------------------------------------------------+
```

__ 3. Print Guides

Print a set of CA MICS documentation guides.

Enter Document Access from the CA MICS Workstation Facility. Choose Document Print and select the guides you wish to print. Once you have completed this checklist and generated your products, you will be able to print guides tailored to your site's definitions.

```
+---------------------------------------------------+
|References: Section 3.6.1                          |
|            Chapter 6, Document Access Guide       |
+---------------------------------------------------+
```

3.8.3 Adding a New Product to a New Database Unit

This checklist defines the tasks required to add a new product to a new database unit. The checklist tasks are organized into the following groups:

1 - Define and Generate New Product(s)
2 - Generate the New Database Unit(s)
3 - Finalize the Reconfigured Database Complex

3.8.3.1 Define and Generate New Product(s)

__ 1. Define Products to be Generated

Check off the products that require generation:

Note: CAP, PER, STG, LDE, VCC, and VMT do not require generation.

Standard Products

___ BAS - CA MICS base platform

___ ACT - Accounting and Chargeback Option

___ AST - CA ASTEX Option

___ AUM - Analyzer Option for MeasureWare

___ CIC - Analyzer Option for CICS

___ DB2 - Analyzer Option for DB2

___ DEX - Analyzer Option for VAX/VMS

___ HSM - VCA support for DFHSM

___ IDM - Analyzer Option for CA IDMS

___ IMS - Analyzer Option for IMS

___ MQS - Analyzer Option for MQSeries

___ PWR - Analyzer Option for VSE/Power

___ RMF - Hardware and SCP Analyzer

___ SMF - Batch and Operations Analyzer

```
       ___ SNT - Network Analyzer Option
       ___ SRL - System Reliability Analyzer Option
       ___ TDM - Tandem Option
       ___ TLM - Tape Analyzer Option, TLM Component
       ___ TSO - Analyzer for TSO
       ___ VCA - Space Analyzer Option
       ___ VMC - Analyzer Option for VM/CMS
       ___ VTS - Tape Analyzer Option, VTS Component
       ___ WEB - Web Analyzer Option

       Field Developed Applications (Including MSI and MTI)

       ___ (___) - ____________________________
       ___ (___) - ____________________________
       ___ (___) - ____________________________

       +---------------------------------------------------+
       | Reference(s):  Section 3.4.1                      |
       +---------------------------------------------------+


 *************************************************************
 *  Repeat Steps 2 through 6 in this section for all         *
 *  products that have been checked off for generation.      *
 *************************************************************

__  2. Review Component Generation Options

       Review component generation options in Chapter 7 of
       the product guides for the data integration
       applications (DIAs) you are installing.  Some products
       provide global element tailoring options that you may
       wish to review.  Products that process user SMF data
       sources also provide an option for record selection.

       +---------------------------------------------------+
       |Reference(s):Section 3.4.2                         |
       |             Section 4.4                           |
       |             Section 4.2, System Modification Guide|
       |             Chapter 7, ccc Analyzer Guide(s)      |
       +---------------------------------------------------+


__  3. (CONDITIONAL) Copy Load Module for the CICS Analyzer


       Copy the appropriate load modules for the CICS
       Analyzer to the sharedprefix.MICS.LOAD library based
       on your installation's needs.
```

```
INPUT SOURCE                      Load Module
------------                      -----------
The Monitor
    Versions 8.0/8.1, 2.0/2.1       $CRCPRS
    Versions 7.0/7.1                TMV611

EPILOG 1000                         BGUDCMP

+----------------------------------------------------+
| Reference(s):  Section 6.1, CICS Analyzer Guide    |
+----------------------------------------------------+
```

__ 4. (CONDITIONAL) Define Appropriate PARMS Members

Customize the PARMS members that define account codes, cost center codes, and the exit routines with which they are associated.

For the ccc Analyzer and Other Standard Products (Including FDAs, MSI and MTI):

```
    cccACCT  - Account Code Specification
    cccACRT  - Account Code Exit Routine
```

For Batch and Operations Analyzer (SMF):

```
    ACCOUNT  - Batch Accounting Structure
    ACCTRTE  - Batch Accounting Exit Routine
```

For Accounting and Chargeback (ACT):

```
    COSTCTR  - Accounting Cost Center Structure
    COSTRTE  - Accounting Cost Center Exit Routine
```

For VAX/VMS Analyzer (DEX):

```
    DEXACRTD - DEX Disk Usage Accounting Exit Routine
```

```
+--------------------------------------------------+
| Reference(s):  Section 2.3.1.7, all subsections   |
|                Section 4.1, Accounting and        |
|                             Chargeback User Guide |
|                Chapter 7, ccc Analyzer Guide(s)   |
+--------------------------------------------------+
```

For MTI Components supporting account codes:

```
    cccACCT  - Account Code Specification
    cccACRT  - Account Code Exit Routine
```

__ 5. (CONDITIONAL) Execute JCLGENUC for MTI Components

During the creation of an MTI component, sharedprefix.MICS.PARMS(cccINPUT) is generated. This member contains the data sets that will be used during component creation to determine which CA NSM performance management resources and resource-subtypes will be included in the component.

Ensure that this member contains the input data sets for this component.
If changes were made to cccINPUT, then

__ Edit sharedprefix.MICS.PARMS(JCLGENUC) and copy the contents of sharedprefix.MICS.PARMS(JCL$0ccc) into this member (JCLGENUC).

__ Submit the following job:

```
     'sharedprefix.MICS.CNTL(JCLGENUC)'
```

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.

```
+-----------------------------------------------------+
|Reference(s):Section 8.5, System Modification Guide|
+-----------------------------------------------------+
```

__ 6. Execute cccCGEN for Each Product

Generate the product.

Run the component generation process for each product, where ccc is the component identifier.

Submit the following job:

```
'         sharedprefix.MICS.CNTL(cccCGEN)'
```

Ensure that there are no error messages in ISPLOG and MICSLOG; that MICSLOG contains the normal termination message, BAS60199I; and that the job completes with a condition code of zero.

Check off each component generation as it is done, running the BASCGEN first:

```
   __ BASCGEN

   __ ACTCGEN __ ASTCGEN __ AUMCGEN __ CICCGEN
   __ DB2CGEN __ DEXCGEN __ HSMCGEN __ IDMCGEN
   __ IMSCGEN __ MQSCGEN __ PWRCGEN __ RMFCGEN
   __ SMFCGEN __ SNTCGEN __ SRLCGEN __ TLMCGEN
   __ TDMCGEN __ TSOCGEN __ VCACGEN __ VMCCGEN
   __ VTSCGEN __ WEBCGEN

   __ cccCGEN for FDA, MSI, and MTI (if applicable)
```

```
+-----------------------------------------------------+
| Reference(s):  Section 3.4.3                       |
+-----------------------------------------------------+
```

MTI Components

```
   ___ (___) - ____________________________
   ___ (___) - ____________________________
```

```
************************************************************
*  Repeat Steps 2 through 6 in this section for all        *
*  products that have been checked off for generation.     *
************************************************************
```

3.8.3.2 Generate the New Database Unit(s)

```
 ************************************************************
 *  Repeat the steps in this section for each new database  *
 *  unit in the CA MICS complex.                            *
 ************************************************************

__  1. Define Catalog Index Prefixes/Tapeprefixes

       Define the catalog index for the prefix and the
       tapeprefix.

       No action is required if the high-level node name
       has already been defined as an alias in the master
       catalog.

       +---------------------------------------------------+
       | Reference(s): Section 3.5.1                       |
       +---------------------------------------------------+


__  2. (OPTIONAL) Worksheet Preparation

       Complete the set of Installation Preparation
       Worksheets for the database unit being defined.

       +---------------------------------------------------+
       | Reference(s): Section 2.3, all subsections        |
       |               Chapter 7, ccc Analyzer Guide(s)    |
       +---------------------------------------------------+


__  3. Execute COPYLIBS

       Build the database unit libraries.

       __ Edit sharedprefix.MICS.CNTL(COPYLIBS) and fill in
          the parameters on the EXEC statement.  DO NOT enter
          parameter values on the PROC statement.

       __ Submit the following job:

           'sharedprefix.MICS.CNTL(COPYLIBS)'

       +---------------------------------------------------+
       | Reference(s): Section 3.5.2                       |
       +---------------------------------------------------+
```

__ 4. Define JCL Parameters

Define all of the members in the prefix.MICS.PARMS library that are required by JCLGEN:

prefix.MICS.PARMS(JCLDEF)
prefix.MICS.PARMS(USERJCL)
prefix.MICS.PARMS(INPUTRDR or INPUTccc) if an SMF source component will be installed in this database unit
prefix.MICS.PARMS(JCLNAMES) if the default PROC or data set names should not be used for this unit

```
+-----------------------------------------------------+
| Reference(s): Section 2.3.3.2.1 (JCLDEF)            |
|               Section 2.3.3.2.2 (USERJCL)           |
|               Section 2.3.3.2.3 (INPUTRDR)          |
|               Section 2.3.3.3.2.3 (JCLNAMES)        |
|               Chapter 7, ccc Analyzer Guide(s)      |
+-----------------------------------------------------+
```

__ 5. Verify JCL and CLIST Attributes

Verify the JCL and CLIST attributes for each CA MICS product by reviewing sharedprefix.MICS.PARMS(JCL$Iccc), where ccc is the three character component identifier. If changes are required, make the changes in prefix.MICS.PARMS(JCLINFO).

```
+-----------------------------------------------------+
| Reference(s): Section 2.3.3.3.2.2 (JCLINFO)         |
+-----------------------------------------------------+
```

__ 6. Generate Unit Installation Jobs

Generate the CA MICS Unit Installation Jobs.

__ Edit prefix.MICS.CNTL(JCLGENA). Enter the parameters on the EXEC statement and the data set name (DSN=) parameter on the //ULOCKIT DD statement.

__ Submit the following job:

```
'prefix.MICS.CNTL(JCLGENA)'
```

```
       +---------------------------------------------------+
       | Reference(s): Section 3.5.3.2                     |
       +---------------------------------------------------+


__  7. Review JCL in PROCJOB2

       Check the JCL generated by JCLGENA by examining
       the following members in prefix.MICS.CNTL:

         PROCJOB2   MICSNDB   MICSDU

       If the JCL and JES parameters do not meet your
       installation's standards, update the
       prefix.MICS.PARMS members JCLDEF or USERJCL and
       resubmit JCLGENA.

       +---------------------------------------------------+
       | Reference(s): Section 3.3.3                       |
       +---------------------------------------------------+


__  8. Execute PROCJOB2

       Move the first group of CA MICS cataloged JCL
       procedures to the production PROC library by
       submitting the following job:

         'prefix.MICS.CNTL(PROCJOB2)'

       +---------------------------------------------------+
       | Reference(s): Section 3.5.4                       |
       +---------------------------------------------------+


__  9. Execute COPYLIBU

       Move the CA MICS parameter library members that are
       dependent on the products installed in this database
       unit.  Submit the following job:

         'prefix.MICS.CNTL(COPYLIBU)'

       +---------------------------------------------------+
       | Reference(s): Section 3.5.2.2                     |
       +---------------------------------------------------+


__ 10. Define Required Parameters
```

Define the required PARMS members.

__ CA MICS base platform (BAS):

```
    'prefix.MICS.PARMS(SYSID)'
    'prefix.MICS.PARMS(ZONE)'
    'prefix.MICS.PARMS(SITE)'
```

Update the members in prefix.MICS.PARMS for the appropriate products.

__ CA MICS Analyzer Option for MeasureWare (AUM):

```
    'prefix.MICS.PARMS(INPUTAUM)'
    'prefix.MICS.PARMS(AUMAURT)'
    'prefix.MICS.PARMS(AUMOPS)'
```

__ CA MICS Analyzer Option for VAX/VMS (DEX):

```
    'prefix.MICS.PARMS(INPUTDEX)'
    'prefix.MICS.PARMS(DEXOPS)'
```

__ CA MICS Analyzer Option for IMS (IMS):

```
    'prefix.MICS.PARMS(INPUTIMS)'
    'prefix.MICS.PARMS(IMSAURT)'
    'prefix.MICS.PARMS(IMSOPS)'
    'prefix.MICS.PARMS(IMSRLRT)'
    'prefix.MICS.PARMS(IMSSPND)'
```

__ CA MICS Analyzer Option for VSE/Power (PWR):

```
    'prefix.MICS.PARMS(INPUTPWR)'
    'prefix.MICS.PARMS(PWRCONFG)'
    'prefix.MICS.PARMS(PWRGROUP)'
    'prefix.MICS.PARMS(PWRGPRTE)'
    'prefix.MICS.PARMS(PWROPS)'
    'prefix.MICS.PARMS(PWRTNRTE)'
```

__ CA MICS System Reliability Analyzer Option (SRL):

```
    'prefix.MICS.PARMS(INPUTSRL)'
    'prefix.MICS.PARMS(SRLDEVS)'
    'prefix.MICS.PARMS(SRLOPS)'
```

__ CA MICS Tandem Option (TDM):

```
    'prefix.MICS.PARMS(INPUTTDM)'
    'prefix.MICS.PARMS(TDMOPS)'

__ CA MICS Tape Analyzer Option, TLM Component:

    'prefix.MICS.PARMS(INPUTTLM)'
    'prefix.MICS.PARMS(TLMOPS)'
    'prefix.MICS.PARMS(VOLGPRTE)'

__ Field Developed Applications (FDAs) for MSI:

    'prefix.MICS.PARMS(cccOPS)'

__ Field Developed Applications (FDAs) for MTI:

    'prefix.MICS.PARMS(cccOPS)'
    'prefix.MICS.PARMS(INPUTccc)'
    'prefix.MICS.PARMS(cccSYSID)'

__ CA MICS Analyzer Option for VM/CMS (VMC):

    'prefix.MICS.PARMS(INPUTVMC)'
    'prefix.MICS.PARMS(VMCOPS)'

****************************************************
* --------------INPUT SOURCE NOTE------------------*
*                                                  *
* The following components may use SMF data as an  *
* input source:                                    *
*        AST, CIC, DB2, HSM, IDM, MQS, RMF,        *
*        SMF, SNT, VTS, TSO, VCA, WEB, and         *
*        FDAs (if applicable)                      *
*                                                  *
* For the components listed above, you must choose *
* one of the alternative methods shown below.      *
* These specifications identify the data set names *
* containing the raw data that will be processed   *
* by the unit's DAILY update job.                  *
*                                                  *
*   (1) If DAYSMF ON is specified in               *
*       prefix.MICS.PARMS(JCLDEF), then you must   *
*       identify the input data set names in       *
*       prefix.MICS.PARMS(INPUTRDR).               *
*                                                  *
*   (2) If DAYSMF OFF is specified in              *
*       prefix.MICS.PARMS(JCLDEF) you must         *
*       identify the input data set names in       *
*       prefix.MICS.PARMS(INPUTccc) for each       *
*       component in the unit.                     *
*                                                  *
*   (3) If this component is in a unit by itself,  *
*       then you must identify the input data set  *
*       names in prefix.MICS.PARMS(INPUTRDR).      *
*                                                  *
****************************************************

__ CA MICS CA ASTEX Option (AST):

    'prefix.MICS.PARMS(INPUTAST)'
    'prefix.MICS.PARMS(ASTAURT)'
    'prefix.MICS.PARMS(ASTOPS)'

__ CA MICS Analyzer Option for CICS (CIC):

    'prefix.MICS.PARMS(INPUTCIC)'
    'prefix.MICS.PARMS(CICALONE)'
    'prefix.MICS.PARMS(CICAURT)'
    'prefix.MICS.PARMS(CICMSAC)'
    'prefix.MICS.PARMS(CICOPS)'
    'prefix.MICS.PARMS(CICRLRT)'
    'prefix.MICS.PARMS(CICTHRSH)'

__ CA MICS Analyzer Option for DB2 (DB2):

    'prefix.MICS.PARMS(INPUTDB2)'
    'prefix.MICS.PARMS(DB2OPS)'

__ CA MICS Analyzer Option for CA IDMS (IDM):

    'prefix.MICS.PARMS(INPUTIDM)'
    'prefix.MICS.PARMS(IDMAURT)'
    'prefix.MICS.PARMS(IDMOPS)'
    'prefix.MICS.PARMS(IDMRLRT)'
    'prefix.MICS.PARMS(IDMRPTS)'

__ CA MICS Analyzer Option for MQSeries (MQS):

    'prefix.MICS.PARMS(INPUTMQS)'
    'prefix.MICS.PARMS(MQSOPS)'

__ CA MICS Hardware and SCP Analyzer (RMF):

    'prefix.MICS.PARMS(INPUTRMF)'
    'prefix.MICS.PARMS(RMFOPS)'

__ CA MICS Batch and Operations Analyzer (SMF):

    'prefix.MICS.PARMS(INPUTSMF)'
    'prefix.MICS.PARMS(JOBGROUP)'
    'prefix.MICS.PARMS(JOBGPRTE)'
    'prefix.MICS.PARMS(PRINTDEF)'
    'prefix.MICS.PARMS(SMFOPS)'
    'prefix.MICS.PARMS(SMFUOPTS)'
    'prefix.MICS.PARMS(TURNRTE)'

__ CA MICS Tape Analyzer Option, VTS Component:

    'prefix.MICS.PARMS(INPUTVTS)'
    'prefix.MICS.PARMS(VTSOPS)'
    'prefix.MICS.PARMS(VTSCMPID)'

__ CA MICS Network Analyzer Option (SNT):

    'prefix.MICS.PARMS(INPUTSNT)'
    'prefix.MICS.PARMS(NETGROUP)'
    'prefix.MICS.PARMS(NETGPRT)'
    'prefix.MICS.PARMS(NETAPPL)'
    'prefix.MICS.PARMS(SNTCIDRT)'
    'prefix.MICS.PARMS(SNTNIDRT)'
    'prefix.MICS.PARMS(SNTOPS)'
    'prefix.MICS.PARMS(SNTPLURT)'
    'prefix.MICS.PARMS(SNTRPTD)'
    'prefix.MICS.PARMS(SNTRPTM)'

__ CA MICS Analyzer for TSO:

    'prefix.MICS.PARMS(INPUTTSO)'
    'prefix.MICS.PARMS(TSOAURT)'
    'prefix.MICS.PARMS(TSOOPS)'

__ CA MICS Space Analyzer Option (VCA):

    'prefix.MICS.PARMS(INPUTHSM)'
    'prefix.MICS.PARMS(INPUTVCA)'
    'prefix.MICS.PARMS(HSMOPS)'
    'prefix.MICS.PARMS(VCAOPS)'

__ CA MICS Web Analyzer Option (WEB):

    'prefix.MICS.PARMS(INPUTWEB)'
    'prefix.MICS.PARMS(WEBOPS)'
    'prefix.MICS.PARMS(WEBSLIST)'
    'prefix.MICS.PARMS(WLECONF)'
    'prefix.MICS.PARMS(WLESCRPT)'

 __ Field Developed Applications (FDAs) using SMF data
    as input source:
```

```
                  'prefix.MICS.PARMS(INPUTccc)'

              ***IMPORTANT****IMPORTANT***IMPORTANT****IMPORTANT***
              *                                                   *
              *  FDAs may or may not use SMF data as an input     *
              *  source.                                          *
              *                                                   *
              *  CA MICS does NOT provide an automated interface  *
              *  to CA SMF Director from within an FDA.  This     *
              *  means that CA MICS will NOT be able to read the  *
              *  CA SMF Director duplicate split index and obtain *
              *  the necessary input files.  However, any SMF data*
              *  sets output from CA SMF Director may still be    *
              *  directly input as standard SMF data.             *
              *                                                   *
              ***IMPORTANT****IMPORTANT***IMPORTANT****IMPORTANT***

              +---------------------------------------------------+
              |  References:  Section 2.3.2, all subsections      |
              |               Section 2.3.3.2, all subsections    |
              |               Chapter 7, ccc Analyzer Guide(s)    |
              |               Section 7.6, System Modification    |
              |                 Guide (for MSI)                   |
              |               Section 8.7, System Modification    |
              |                 Guide (for MTI)                   |
              |               Section 1.5.2.3, CA ASTEX Option    |
              |                 Guide                             |
              |               Section 7.2.2, Application          |
              |                 Extension Option for Tandem Guide |
              +---------------------------------------------------+


      __ 11. (OPTIONAL) Update DBMODEL

             Update prefix.MICS.PARMS(DBMODEL).

             The COPYLIBU job generated the DBMODEL parameter
             library member.  Review and modify the DBMODEL
             contents.

             Change the retention limits on the FILE statements
             according to your retention needs.

             +---------------------------------------------------+
             | Reference(s): Section 2.3.4                       |
             |               Chapter 7, System Modification      |
             |                 Guide, for user components        |
             +---------------------------------------------------+
```

__ 12. (OPTIONAL) Update COMPRESS

Update prefix.MICS.PARMS(COMPRESS).

The COPYLIBU job generated the COMPRESS parameter library member. Review and modify the COMPRESS contents.

```
+-----------------------------------------------------+
| Reference(s): Section 2.3.2.6                       |
+-----------------------------------------------------+
```

__ 13. Review UNITGEN Options

Review sharedprefix.MICS.PARMS(UNITGEN) for the options in effect.

```
+-----------------------------------------------------+
|  Reference(s):  Section 5.6.1                       |
+-----------------------------------------------------+
```

__ 14. Execute UNITGEN

Submit the following job:

```
  'prefix.MICS.CNTL(UNITGEN)'
```

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

If the UNITGEN NOAUTOSUBMIT, or UNITGEN AUTOSUBMIT and NOOPJCL parameters have been specified, UNITGEN will complete with a normal return code of 4. Otherwise, UNITGEN and UNITGEN1 should complete with a return code of 0.

__ 15. (CONDITIONAL) Execute UNITGEN1

Perform this step only if your installation has specified

UNITGEN NOAUTOSUBMIT

You must manually submit the UNITGEN1 job to perform

the required JCL and parameter generations.

After the UNITGEN job from the previous step completes successfully, submit the following job:

```
  'prefix.MICS.CNTL(UNITGEN1)'
```

__ 16. Verify UNITGEN1 Results

Verify that the UNITGEN1 job executed successfully.

The UNITGEN1 job contains multiple steps that perform COPYLIBU, CYCLEGEN, JCLGEN, cccPGEN, and GDGSGEN processing or part of it.

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

All steps should complete with a return code of zero.

If errors are encountered during UNITGEN1 job execution, correct the errors and rerun UNITGEN and UNITGEN1 jobs respectively.

```
+---------------------------------------------------+
|  Reference(s):  Sections 5.6 and 5.6.1            |
+---------------------------------------------------+
```

__ 17. Execute JCLGENC

Generate the CA MICS database set-up PROCS. Submit the following job:

```
  'prefix.MICS.CNTL(JCLGENC)'
```

```
+-----------------------------------------------------+
| Reference(s): Section 3.5.3.3                       |
+-----------------------------------------------------+
```

__ 18. Execute PROCJOB3

Move CA MICS database set-up PROCS to the installation PROCLIB by submitting the following job:

```
  'prefix.MICS.CNTL(PROCJOB3)'
```

```
       +---------------------------------------------------+
       | Reference(s): Section 3.5.4                       |
       +---------------------------------------------------+


__ 19. (CONDITIONAL) Execute JCLGEND

       Perform this step only if your installation has
       specified

       UNITGEN NOOPJCL

       If NOOPJCL is specified, UNITGEN1 did not generate the
       CA MICS operational jobs.

       To generate these jobs, submit the following job:

         'prefix.MICS.CNTL(JCLGEND)'

       Ensure that there are no error messages in MICSLOG or
       SYSTSPRT and that the MICSLOG contains the normal
       termination messages, BAS10999I or BAS11999I.

       All steps should complete with a return code of zero.

       +---------------------------------------------------+
       |  Reference(s):  Section 5.6.1                     |
       +---------------------------------------------------+


__ 20. Complete Accounting and Chargeback Processes

       If Accounting and Chargeback is being installed,
       define the parameters and execute the generation
       processes required for this product.  Use the
       checklists in the Accounting and Chargeback User
       Guide.  Once you have completed the checklist sections
       that deal with planning, installation, and activation;
       continue the installation process.

       +---------------------------------------------------+
       |Reference(s): Chapter 2, Accounting and Chargeback |
       |                User Guide                         |
       +---------------------------------------------------+


__ 21. (OPTIONAL) Estimate Database Space
```

```
       Estimate the database space to allocate on disk.

       Review the JCL in prefix.MICS.CNTL(DBSIZE). The input
       data sets should contain data of about one day.

       __ Submit the following job:

           'prefix.MICS.CNTL(DBSIZE)'

       +----------------------------------------------------+
       | Reference(s): Section 2.3.4.2                      |
       +----------------------------------------------------+


__ 22. Allocate Database Files

       Allocate the Online Database and associated files.

       __ Edit prefix.MICS.CNTL(ALLOCATE) and fill in the
          SPACE, VOLSER, and UNIT keyword values.

       __ Submit the following job:

           'prefix.MICS.CNTL(ALLOCATE)'

       +----------------------------------------------------+
       | Reference(s): Section 3.5.6.2                      |
       +----------------------------------------------------+


__ 23. Review EXECDEF

       Review the default values for the dynamic execution
       options in prefix.MICS.PARMS(EXECDEF) and update as
       needed.  Carefully review the entries affecting data
       retention limits.

       +----------------------------------------------------+
       | References: Section 2.3.5                          |
       |             Section A.3                            |
       |             Section 7.3.6 Batch and Operations     |
       |              Analyzer Guide                        |
       +----------------------------------------------------+


__ 24. Initialize Daily Audit Tape for Accounting and
       Chargeback

       If Accounting and Chargeback is being installed in
```

this database unit, initialize the daily audit tape used for journal consolidation as the first tape in a generation data group series. Submit the following job:

```
  'prefix.MICS.CNTL(ACTINITA)'
```

__ 25. Incremental Update Preparation

If you activated an incremental update for one or more products in this unit database, run the cccIUALC job for each of these products. The cccIUALC job allocates the product's incremental update checkpoint and database files.

__ Edit the member according to your organization's data set allocation parameter standards and DASD space requirements. Submit the following job:

```
  'prefix.MICS.CNTL(cccIUALC)'
```

__ If you specified

```
  INCRDB TAPE
```

for one or more of these products, run the product's cccIUGDG job to create GDG indices for the incremental update tape database files. Submit the following job:

```
  'prefix.MICS.CNTL(cccIUGDG)'
```

```
+-----------------------------------------------------+
| Reference(s): Chapter 7, ccc Analyzer Guide(s)     |
+-----------------------------------------------------+
```

__ 26. Initialize Database Checkpoint File

Before performing the first database update, initialize the checkpoint file. Submit the following job:

```
  'prefix.MICS.CNTL(CKPTINIT)'
```

```
+-----------------------------------------------------+
| Reference(s): Section 3.5.7                        |
+-----------------------------------------------------+
```

__ 27. Review Generated Unit

The system is fully generated and ready for checkout. Make a final review of all definitions and JCL, applying any final user modifications.

```
+---------------------------------------------------+
| Reference(s): Section 3.5.7                       |
+---------------------------------------------------+
```

__ 28. Execute DAILY

Run the first daily process. Submit the following job:

```
  'prefix.MICS.CNTL(DAILY)'
```

or

Use the Operational Status and Tracking DAILY command in MWF to submit daily processing.

```
+---------------------------------------------------+
| Reference(s): Section 3.5.7 and Chapter 4         |
+---------------------------------------------------+
```

__ 29. (OPTIONAL) Verify Results in Database

Run a database check job for each product in the unit to verify that key database definitions have been correctly defined and implemented. Submit the following job:

```
  'prefix.MICS.CNTL(cccCHECK)'
```

```
+---------------------------------------------------+
| Reference(s): Section 3.5.7                       |
+---------------------------------------------------+
```

__ 30. (OPTIONAL) Ensure Backup and Restore Are Valid

Run the backup and restore jobs. Submit the following jobs:

```
         'prefix.MICS.CNTL(BACKUP)'
         'prefix.MICS.CNTL(RESTORE)'

       or

       Use the Operational Status and Tracking BACKUP and
       RESTORE commands in MWF to submit the processing.

       +---------------------------------------------------+
       | Reference(s): Section 3.5.7 and Chapter 4         |
       +---------------------------------------------------+


__ 31. Implement Production Operation

       If the database unit being installed is a production
       unit, implement production operation for CA MICS daily
       processing.

       +---------------------------------------------------+
       | Reference(s): Section 3.7.1                       |
       +---------------------------------------------------+

 *************************************************************
 *  Repeat the steps in this section for each new database  *
 *  unit in the complex.                                     *
 *************************************************************
```

3.8.3.3 Finalize the Reconfigured Database Complex

```
 ************************************************************
 *  Complete this checklist only after all of the database  *
 *  units in the complex have been installed.               *
 ************************************************************

__  1. Back Up Libraries

       Since CA MICS does not provide a standard backup of
       libraries, back up the libraries below so that
       recovery can be accomplished.

       Complex-level data sets:

          sharedprefix.MICS.ASM
          sharedprefix.MICS.BIN
          sharedprefix.MICS.CLIST
          sharedprefix.MICS.CNTL
          sharedprefix.MICS.DIC.TEXT
          sharedprefix.MICS.DOC.TEXT
          sharedprefix.MICS.GENLIB
          sharedprefix.MICS.HOLD.PARMS
          sharedprefix.MICS.HOLD.USER.SOURCE
          sharedprefix.MICS.INC.TEXT
          sharedprefix.MICS.INCLLIB
          sharedprefix.MICS.ISPHLIB
          sharedprefix.MICS.ISPMLIB
          sharedprefix.MICS.ISPPLIB
          sharedprefix.MICS.ISPSLIB
          sharedprefix.MICS.ISPTLIB
          sharedprefix.MICS.LIBLOAD.CNTL
          sharedprefix.MICS.LOAD
          sharedprefix.MICS.LOCALMOD.CNTL
          sharedprefix.MICS.MACAUTOS
          sharedprefix.MICS.MACLIB
          sharedprefix.MICS.MCOLIB
          sharedprefix.MICS.MWFPDS.DATA
          sharedprefix.MICS.MWFSAS.DATA
          sharedprefix.MICS.OBJ
          sharedprefix.MICS.PARMS
          sharedprefix.MICS.PROTOLIB
          sharedprefix.MICS.SASFLS
          sharedprefix.MICS.SCREENS
          sharedprefix.MICS.SOURCE
          sharedprefix.MICS.TABLES
          sharedprefix.MICS.TEST.INCLLIB
          sharedprefix.MICS.TEST.LOAD
          sharedprefix.MICS.TEST.SOURCE
          sharedprefix.MICS.USER.SOURCE


       Database unit data sets:

          prefix.MICS.CNTL
          prefix.MICS.MUOLIB
          prefix.MICS.PARMS
          prefix.MICS.USER.SOURCE

       +---------------------------------------------------+
       | Reference(s): Section 3.6.2                       |
       +---------------------------------------------------+


__  2. Verify MICF Full Execution


       Check MICF for full execution.

       Enter the CA MICS ISPF environment using the method
       selected by your site (e.g., LIBDEF, CLIST, or logon
       PROC).  Select the MICF option from the CA MICS
       Workstation Facility.  Review any new inquiries and
       execute MICF features.

       +---------------------------------------------------+
       |References: Chapter 2, MICF User Guide             |
       |            Chapter 3, ccc Analyzer Guide(s)       |
       |                        (REPORTS)                  |
       +---------------------------------------------------+


__  3. Print Guides

       Print a set of CA MICS documentation guides.

       Enter Document Access from the CA MICS Workstation
       Facility.  Choose Document Print and select the guides
       you wish to print.  Once you have completed this
       checklist and generated your products, you will be
       able to print guides tailored to your site's
       definitions.

       +---------------------------------------------------+
       |References: Section 3.6.1                          |
       |            Chapter 6, Document Access Guide       |
       +---------------------------------------------------+
__   4. Verify Tape and DASD Management Systems


       Check that CA MICS has been properly established in
       your installation's tape and DASD management systems.

       +-----------------------------------------------------+
       | Reference(s): Section 3.6.2                         |
       +-----------------------------------------------------+

__   5. (OPTIONAL) Offsite Storage

       Arrange for offsite storage of CA MICS monthly backup
       tapes.

       +-----------------------------------------------------+
       | Reference(s): Section 3.6.2                         |
       +-----------------------------------------------------+
```

3.8.4 Adding a Database Unit to a Database Complex

This checklist defines the tasks required to create a new database unit in an existing database complex. The checklist tasks are organized into the following groups:

1 - Generate the New Database Unit(s)
2 - Finalize the Reconfigured Database Complex

3.8.4.1 Generate the New Database Unit(s)

```
 ************************************************************
 *  Repeat the steps in this section for each new database  *
 *  unit in the CA MICS complex.                            *
 ************************************************************

__  1. Define Catalog Index Prefixes/Tapeprefixes

       Define the catalog index for the prefix and the
       tapeprefix.

       No action is required if the high-level node name
       has already been defined as an alias in the master
       catalog.

       +----------------------------------------------------+
       | Reference(s): Section 3.5.1                        |
       +----------------------------------------------------+


__  2. (OPTIONAL) Worksheet Preparation

       Complete the set of Installation Preparation
       Worksheets for the database unit being defined.

       +----------------------------------------------------+
       | Reference(s): Section 2.3, all subsections         |
       |               Chapter 7, ccc Analyzer Guide(s)     |
       +----------------------------------------------------+


__  3. Execute COPYLIBS

       Build the database unit libraries.

       __ Edit sharedprefix.MICS.CNTL(COPYLIBS) and fill in
          the parameters on the EXEC statement.  DO NOT enter
          parameter values on the PROC statement.

       __ Submit the following job:

           'sharedprefix.MICS.CNTL(COPYLIBS)'

       +----------------------------------------------------+
       | Reference(s): Section 3.5.2                        |
       +----------------------------------------------------+
```

__ 4. Define JCL Parameters

Define all of the members in the prefix.MICS.PARMS library that are required by JCLGEN:

prefix.MICS.PARMS(JCLDEF)
prefix.MICS.PARMS(USERJCL)
prefix.MICS.PARMS(INPUTRDR or INPUTccc) if an SMF
  source component will be installed in this database
  unit
prefix.MICS.PARMS(JCLNAMES) if the default PROC or
  data set names should not be used for this unit

```
+---------------------------------------------------+
| Reference(s): Section 2.3.3.2.1 (JCLDEF)          |
|               Section 2.3.3.2.2 (USERJCL)         |
|               Section 2.3.3.2.3 (INPUTRDR)        |
|               Section 2.3.3.3.2.3 (JCLNAMES)      |
|               Chapter 7, ccc Analyzer Guide(s)    |
+---------------------------------------------------+
```

__ 5. Verify JCL and CLIST Attributes

Verify the JCL and CLIST attributes for each CA MICS product by reviewing sharedprefix.MICS.PARMS(JCL$Iccc), where ccc is the three character component identifier. If changes are required, make the changes in prefix.MICS.PARMS(JCLINFO).

```
+---------------------------------------------------+
| Reference(s): Section 2.3.3.3.2.2 (JCLINFO)       |
+---------------------------------------------------+
```

__ 6. Generate Unit Installation Jobs

Generate the CA MICS Unit Installation Jobs.

__ Edit prefix.MICS.CNTL(JCLGENA). Enter the parameters on the EXEC statement and the data set name (DSN=) parameter on the //ULOCKIT DD statement.

__ Submit the following job:

```
'prefix.MICS.CNTL(JCLGENA)'
```

```
       +---------------------------------------------------+
       | Reference(s): Section 3.5.3.2                     |
       +---------------------------------------------------+


__  7. Review JCL in PROCJOB2

       Check the JCL generated by JCLGENA by examining
       the following members in prefix.MICS.CNTL:

         PROCJOB2   MICSNDB   MICSDU

       If the JCL and JES parameters do not meet your
       installation's standards, update the
       prefix.MICS.PARMS members JCLDEF or USERJCL and
       resubmit JCLGENA.

       +---------------------------------------------------+
       | Reference(s): Section 3.3.3                       |
       +---------------------------------------------------+


__  8. Execute PROCJOB2

       Move the first group of CA MICS cataloged JCL
       procedures to the production PROC library by
       submitting the following job:

         'prefix.MICS.CNTL(PROCJOB2)'

       +---------------------------------------------------+
       | Reference(s): Section 3.5.4                       |
       +---------------------------------------------------+


__  9. Execute COPYLIBU

       Move the CA MICS parameter library members that are
       dependent on the products installed in this database
       unit.  Submit the following job:

         'prefix.MICS.CNTL(COPYLIBU)'

       +---------------------------------------------------+
       | Reference(s): Section 3.5.2.2                     |
       +---------------------------------------------------+


__ 10. Define Required Parameters
```

```
Define the required PARMS members.

__ CA MICS base platform (BAS):

    'prefix.MICS.PARMS(SYSID)'
    'prefix.MICS.PARMS(ZONE)'
    'prefix.MICS.PARMS(SITE)'

Update the members in prefix.MICS.PARMS for the
appropriate products.

__ CA MICS Analyzer Option for MeasureWare (AUM):

    'prefix.MICS.PARMS(INPUTAUM)'
    'prefix.MICS.PARMS(AUMAURT)'
    'prefix.MICS.PARMS(AUMOPS)'

__ CA MICS Analyzer Option for VAX/VMS (DEX):

    'prefix.MICS.PARMS(INPUTDEX)'
    'prefix.MICS.PARMS(DEXOPS)'

__ CA MICS Analyzer Option for IMS (IMS):

    'prefix.MICS.PARMS(INPUTIMS)'
    'prefix.MICS.PARMS(IMSAURT)'
    'prefix.MICS.PARMS(IMSOPS)'
    'prefix.MICS.PARMS(IMSRLRT)'
    'prefix.MICS.PARMS(IMSSPND)'

__ CA MICS Analyzer Option for VSE/Power (PWR):

    'prefix.MICS.PARMS(INPUTPWR)'
    'prefix.MICS.PARMS(PWRCONFG)'
    'prefix.MICS.PARMS(PWRGROUP)'
    'prefix.MICS.PARMS(PWRGPRTE)'
    'prefix.MICS.PARMS(PWROPS)'
    'prefix.MICS.PARMS(PWRTNRTE)'

__ CA MICS System Reliability Analyzer Option (SRL):

    'prefix.MICS.PARMS(INPUTSRL)'
    'prefix.MICS.PARMS(SRLDEVS)'
    'prefix.MICS.PARMS(SRLOPS)'

__ CA MICS Tandem Option (TDM):

    'prefix.MICS.PARMS(INPUTTDM)'
    'prefix.MICS.PARMS(TDMOPS)'

__ CA MICS Tape Analyzer Option, TLM Component:

    'prefix.MICS.PARMS(INPUTTLM)'
    'prefix.MICS.PARMS(TLMOPS)'
    'prefix.MICS.PARMS(VOLGPRTE)'

__ Field Developed Applications (FDAs) for MSI:

    'prefix.MICS.PARMS(cccOPS)'

__ Field Developed Applications (FDAs) for MTI:

    'prefix.MICS.PARMS(cccOPS)'
    'prefix.MICS.PARMS(INPUTccc)'
    'prefix.MICS.PARMS(cccSYSID)'

__ CA MICS Analyzer Option for VM/CMS (VMC):

    'prefix.MICS.PARMS(INPUTVMC)'
    'prefix.MICS.PARMS(VMCOPS)'

****************************************************
* --------------INPUT SOURCE NOTE------------------*
*                                                  *
* The following components may use SMF data as an  *
* input source:                                    *
*       AST, CIC, DB2, HSM, IDM, MQS, RMF,         *
*       SMF, SNT, VTS, TSO, VCA, WEB, and          *
*       FDAs (if applicable)                       *
*                                                  *
* For the components listed above, you must choose *
* one of the alternative methods shown below.      *
* These specifications identify the data set names *
* containing the raw data that will be processed   *
* by the unit's DAILY update job.                  *
*                                                  *
*   (1) If DAYSMF ON is specified in               *
*       prefix.MICS.PARMS(JCLDEF), then you must   *
*       identify the input data set names in       *
*       prefix.MICS.PARMS(INPUTRDR).               *
*                                                  *
*   (2) If DAYSMF OFF is specified in              *
*       prefix.MICS.PARMS(JCLDEF) you must         *
*       identify the input data set names in       *
*       prefix.MICS.PARMS(INPUTccc) for each       *
*       component in the unit.                     *
*                                                  *
*   (3) If this component is in a unit by itself,  *
*       then you must identify the input data set  *
*       names in prefix.MICS.PARMS(INPUTRDR).      *
*                                                  *
****************************************************

__ CA MICS CA ASTEX Option (AST):

    'prefix.MICS.PARMS(INPUTAST)'
    'prefix.MICS.PARMS(ASTAURT)'
    'prefix.MICS.PARMS(ASTOPS)'

__ CA MICS Analyzer Option for CICS (CIC):

    'prefix.MICS.PARMS(INPUTCIC)'
    'prefix.MICS.PARMS(CICALONE)'
    'prefix.MICS.PARMS(CICAURT)'
    'prefix.MICS.PARMS(CICMSAC)'
    'prefix.MICS.PARMS(CICOPS)'
    'prefix.MICS.PARMS(CICRLRT)'
    'prefix.MICS.PARMS(CICTHRSH)'

__ CA MICS Analyzer Option for DB2 (DB2):

    'prefix.MICS.PARMS(INPUTDB2)'
    'prefix.MICS.PARMS(DB2OPS)'

__ CA MICS Analyzer Option for CA IDMS (IDM):

    'prefix.MICS.PARMS(INPUTIDM)'
    'prefix.MICS.PARMS(IDMAURT)'
    'prefix.MICS.PARMS(IDMOPS)'
    'prefix.MICS.PARMS(IDMRLRT)'
    'prefix.MICS.PARMS(IDMRPTS)'

__ CA MICS Analyzer Option for MQSeries (MQS):

    'prefix.MICS.PARMS(INPUTMQS)'
    'prefix.MICS.PARMS(MQSOPS)'

__ CA MICS Hardware and SCP Analyzer (RMF):

    'prefix.MICS.PARMS(INPUTRMF)'
    'prefix.MICS.PARMS(RMFOPS)'

__ CA MICS Batch and Operations Analyzer (SMF):

    'prefix.MICS.PARMS(INPUTSMF)'
    'prefix.MICS.PARMS(JOBGROUP)'
    'prefix.MICS.PARMS(JOBGPRTE)'
    'prefix.MICS.PARMS(PRINTDEF)'
    'prefix.MICS.PARMS(SMFOPS)'
    'prefix.MICS.PARMS(SMFUOPTS)'
    'prefix.MICS.PARMS(TURNRTE)'

__ CA MICS Tape Analyzer Option, VTS Component:

    'prefix.MICS.PARMS(INPUTVTS)'
    'prefix.MICS.PARMS(VTSOPS)'
    'prefix.MICS.PARMS(VTSCMPID)'

__ CA MICS Network Analyzer Option (SNT):

    'prefix.MICS.PARMS(INPUTSNT)'
    'prefix.MICS.PARMS(NETGROUP)'
    'prefix.MICS.PARMS(NETGPRT)'
    'prefix.MICS.PARMS(NETAPPL)'
    'prefix.MICS.PARMS(SNTCIDRT)'
    'prefix.MICS.PARMS(SNTNIDRT)'
    'prefix.MICS.PARMS(SNTOPS)'
    'prefix.MICS.PARMS(SNTPLURT)'
    'prefix.MICS.PARMS(SNTRPTD)'
    'prefix.MICS.PARMS(SNTRPTM)'

__ CA MICS Analyzer for TSO:

    'prefix.MICS.PARMS(INPUTTSO)'
    'prefix.MICS.PARMS(TSOAURT)'
    'prefix.MICS.PARMS(TSOOPS)'

__ CA MICS Space Analyzer Option (VCA):

    'prefix.MICS.PARMS(INPUTHSM)'
    'prefix.MICS.PARMS(INPUTVCA)'
    'prefix.MICS.PARMS(HSMOPS)'
    'prefix.MICS.PARMS(VCAOPS)'

__ CA MICS Web Analyzer Option (WEB):

    'prefix.MICS.PARMS(INPUTWEB)'
    'prefix.MICS.PARMS(WEBOPS)'
    'prefix.MICS.PARMS(WEBSLIST)'
    'prefix.MICS.PARMS(WLECONF)'
    'prefix.MICS.PARMS(WLESCRPT)'

 __ Field Developed Applications (FDAs) using SMF data
    as input source:
            'prefix.MICS.PARMS(INPUTccc)'

        ***IMPORTANT****IMPORTANT***IMPORTANT****IMPORTANT***
        *                                                   *
        *  FDAs may or may not use SMF data as an input     *
        *  source.                                          *
        *                                                   *
        *  CA MICS does NOT provide an automated interface  *
        *  to CA SMF Director from within an FDA.  This     *
        *  means that CA MICS will NOT be able to read the  *
        *  CA SMF Director duplicate split index and obtain *
        *  the necessary input files.  However, any SMF data*
        *  sets output from CA SMF Director may still be    *
        *  directly input as standard SMF data.             *
        *                                                   *
        ***IMPORTANT****IMPORTANT***IMPORTANT****IMPORTANT***

        +---------------------------------------------------+
        |  References:  Section 2.3.2, all subsections      |
        |               Section 2.3.3.2, all subsections    |
        |               Chapter 7, ccc Analyzer Guide(s)    |
        |               Section 7.6, System Modification    |
        |                 Guide (for MSI)                   |
        |               Section 8.7, System Modification    |
        |                 Guide (for MTI)                   |
        |               Section 1.5.2.3, CA ASTEX Option    |
        |                 Guide                             |
        |               Section 7.2.2, Application          |
        |                 Extension Option for Tandem Guide |
        +---------------------------------------------------+


__ 11. (OPTIONAL) Update DBMODEL

       Update prefix.MICS.PARMS(DBMODEL).

       The COPYLIBU job generated the DBMODEL parameter
       library member.  Review and modify the DBMODEL
       contents.

       Change the retention limits on the FILE statements
       according to your retention needs.

       +---------------------------------------------------+
       | Reference(s):  Section 2.3.4                      |
       |                Chapter 7, System Modification     |
       |                  Guide, for user components       |
       +---------------------------------------------------+
```

__ 12. (OPTIONAL) Update COMPRESS

Update prefix.MICS.PARMS(COMPRESS).

The COPYLIBU job generated the COMPRESS parameter library member. Review and modify the COMPRESS contents.

```
+-----------------------------------------------------+
| Reference(s): Section 2.3.2.6                       |
+-----------------------------------------------------+
```

__ 13. Review UNITGEN Options

Review sharedprefix.MICS.PARMS(UNITGEN) for the options in effect.

```
+-----------------------------------------------------+
|  Reference(s):  Section 5.6.1                       |
+-----------------------------------------------------+
```

__ 14. Execute UNITGEN

Submit the following job:

```
  'prefix.MICS.CNTL(UNITGEN)'
```

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

If the UNITGEN NOAUTOSUBMIT, or UNITGEN AUTOSUBMIT and NOOPJCL parameters have been specified, UNITGEN will complete with a normal return code of 4. Otherwise, UNITGEN and UNITGEN1 should complete with a return code of 0.

__ 15. (CONDITIONAL) Execute UNITGEN1

Perform this step only if your installation has specified

UNITGEN NOAUTOSUBMIT

You must manually submit the UNITGEN1 job to perform

the required JCL and parameter generations.

After the UNITGEN job from the previous step completes successfully, submit the following job:

```
  'prefix.MICS.CNTL(UNITGEN1)'
```

__ 16. Verify UNITGEN1 Results

Verify that the UNITGEN1 job executed successfully.

The UNITGEN1 job contains multiple steps that perform COPYLIBU, CYCLEGEN, JCLGEN, cccPGEN, and GDGSGEN processing or part of it.

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

All steps should complete with a return code of zero.

If errors are encountered during UNITGEN1 job execution, correct the errors and rerun UNITGEN and UNITGEN1 jobs respectively.

```
+---------------------------------------------------+
|  Reference(s):  Sections 5.6 and 5.6.1            |
+---------------------------------------------------+
```

__ 17. Execute JCLGENC

Generate the CA MICS database set-up PROCS. Submit the following job:

```
  'prefix.MICS.CNTL(JCLGENC)'
```

```
+----------------------------------------------------+
| Reference(s): Section 3.5.3.3                      |
+----------------------------------------------------+
```

__ 18. Execute PROCJOB3

Move CA MICS database set-up PROCS to the installation PROCLIB by submitting the following job:

```
  'prefix.MICS.CNTL(PROCJOB3)'
```

```
+----------------------------------------------------+
| Reference(s): Section 3.5.4                        |
+----------------------------------------------------+
```

__ 19. (CONDITIONAL) Execute JCLGEND

Perform this step only if your installation has specified

UNITGEN NOOPJCL

If NOOPJCL is specified, UNITGEN1 did not generate the CA MICS operational jobs.

To generate these jobs, submit the following job:

```
  'prefix.MICS.CNTL(JCLGEND)'
```

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

All steps should complete with a return code of zero.

```
+----------------------------------------------------+
|  Reference(s):  Section 5.6.1                      |
+----------------------------------------------------+
```

__ 20. Complete Accounting and Chargeback Processes

If Accounting and Chargeback is being installed, define the parameters and execute the generation processes required for this product. Use the checklists in the Accounting and Chargeback User Guide. Once you have completed the checklist sections that deal with planning, installation, and activation; continue the installation process.

```
+----------------------------------------------------+
|Reference(s): Chapter 2, Accounting and Chargeback |
|                 User Guide                         |
+----------------------------------------------------+
```

__ 21. (OPTIONAL) Estimate Database Space

```
       Estimate the database space to allocate on disk.

       Review the JCL in prefix.MICS.CNTL(DBSIZE). The input
       data sets should contain data of about one day.

       __ Submit the following job:

           'prefix.MICS.CNTL(DBSIZE)'

       +---------------------------------------------------+
       | Reference(s): Section 2.3.4.2                     |
       +---------------------------------------------------+


__ 22. Allocate Database Files

       Allocate the Online Database and associated files.

       __ Edit prefix.MICS.CNTL(ALLOCATE) and fill in the
          SPACE, VOLSER, and UNIT keyword values.

       __ Submit the following job:

           'prefix.MICS.CNTL(ALLOCATE)'

       +---------------------------------------------------+
       | Reference(s): Section 3.5.6.2                     |
       +---------------------------------------------------+


__ 23. Review EXECDEF

       Review the default values for the dynamic execution
       options in prefix.MICS.PARMS(EXECDEF) and update as
       needed.  Carefully review the entries affecting data
       retention limits.

       +---------------------------------------------------+
       | References: Section 2.3.5                         |
       |             Section A.3                           |
       |             Section 7.3.6 Batch and Operations    |
       |              Analyzer Guide                       |
       +---------------------------------------------------+


__ 24. Initialize Daily Audit Tape for Accounting and
       Chargeback

       If Accounting and Chargeback is being installed in
```

this database unit, initialize the daily audit tape used for journal consolidation as the first tape in a generation data group series. Submit the following job:

```
  'prefix.MICS.CNTL(ACTINITA)'
```

__ 25. Incremental Update Preparation

If you activated an incremental update for one or more products in this unit database, run the cccIUALC job for each of these products. The cccIUALC job allocates the product's incremental update checkpoint and database files.

__ Edit the member according to your organization's data set allocation parameter standards and DASD space requirements. Submit the following job:

```
  'prefix.MICS.CNTL(cccIUALC)'
```

__ If you specified

```
  INCRDB TAPE
```

for one or more of these products, run the product's cccIUGDG job to create GDG indices for the incremental update tape database files. Submit the following job:

```
  'prefix.MICS.CNTL(cccIUGDG)'
```

```
+-----------------------------------------------------+
| Reference(s): Chapter 7, ccc Analyzer Guide(s)      |
+-----------------------------------------------------+
```

__ 26. Initialize Database Checkpoint File

Before performing the first database update, initialize the checkpoint file. Submit the following job:

```
  'prefix.MICS.CNTL(CKPTINIT)'
```

```
+-----------------------------------------------------+
| Reference(s): Section 3.5.7                         |
+-----------------------------------------------------+
__ 27. Review Generated Unit

       The system is fully generated and ready for checkout.
       Make a final review of all definitions and JCL,
       applying any final user modifications.

       +---------------------------------------------------+
       | Reference(s): Section 3.5.7                       |
       +---------------------------------------------------+


__ 28. Execute DAILY

       Run the first daily process.  Submit the following
       job:

         'prefix.MICS.CNTL(DAILY)'

       or

       Use the Operational Status and Tracking DAILY command
       in MWF to submit daily processing.

       +---------------------------------------------------+
       | Reference(s): Section 3.5.7 and Chapter 4         |
       +---------------------------------------------------+


__ 29. (OPTIONAL) Verify Results in Database

       Run a database check job for each product in the unit
       to verify that key database definitions have been
       correctly defined and implemented.  Submit the
       following job:

         'prefix.MICS.CNTL(cccCHECK)'

       +---------------------------------------------------+
       | Reference(s): Section 3.5.7                       |
       +---------------------------------------------------+


__ 30. (OPTIONAL) Ensure Backup and Restore Are Valid

       Run the backup and restore jobs.  Submit the following
       jobs:
```

```
                 'prefix.MICS.CNTL(BACKUP)'
                 'prefix.MICS.CNTL(RESTORE)'

               or

               Use the Operational Status and Tracking BACKUP and
               RESTORE commands in MWF to submit the processing.

               +-----------------------------------------------------+
               | Reference(s): Section 3.5.7 and Chapter 4           |
               +-----------------------------------------------------+


        __ 31. Implement Production Operation

               If the database unit being installed is a production
               unit, implement production operation for CA MICS daily
               processing.

               +-----------------------------------------------------+
               | Reference(s): Section 3.7.1                         |
               +-----------------------------------------------------+

         ************************************************************
         *  Repeat the steps in this section for each new database  *
         *  unit in the complex.                                    *
         ************************************************************
```

3.8.4.2 Finalize the Reconfigured Database

```
 ************************************************************
 *  Complete this checklist only after all of the database  *
 *  units in the complex have been installed.               *
 ************************************************************

__  1. Back Up Libraries


       Since CA MICS does not provide a standard backup of
       libraries, back up the libraries below so that
       recovery can be accomplished.

       Complex-level data sets:

          sharedprefix.MICS.ASM
          sharedprefix.MICS.BIN
          sharedprefix.MICS.CLIST
          sharedprefix.MICS.CNTL
          sharedprefix.MICS.DIC.TEXT
          sharedprefix.MICS.DOC.TEXT
          sharedprefix.MICS.GENLIB
          sharedprefix.MICS.HOLD.PARMS
          sharedprefix.MICS.HOLD.USER.SOURCE
          sharedprefix.MICS.INC.TEXT
          sharedprefix.MICS.INCLLIB
          sharedprefix.MICS.ISPHLIB
          sharedprefix.MICS.ISPMLIB
          sharedprefix.MICS.ISPPLIB
          sharedprefix.MICS.ISPSLIB
          sharedprefix.MICS.ISPTLIB
          sharedprefix.MICS.LIBLOAD.CNTL
          sharedprefix.MICS.LOAD
          sharedprefix.MICS.LOCALMOD.CNTL
          sharedprefix.MICS.MACAUTOS
          sharedprefix.MICS.MACLIB
          sharedprefix.MICS.MCOLIB
          sharedprefix.MICS.MWFPDS.DATA
          sharedprefix.MICS.MWFSAS.DATA
          sharedprefix.MICS.OBJ
          sharedprefix.MICS.PARMS
          sharedprefix.MICS.PROTOLIB
          sharedprefix.MICS.SASFLS
          sharedprefix.MICS.SCREENS
          sharedprefix.MICS.SOURCE
          sharedprefix.MICS.TABLES
          sharedprefix.MICS.TEST.INCLLIB
          sharedprefix.MICS.TEST.LOAD
          sharedprefix.MICS.TEST.SOURCE
          sharedprefix.MICS.USER.SOURCE


       Database unit data sets:

          prefix.MICS.CNTL
          prefix.MICS.MUOLIB
          prefix.MICS.PARMS
          prefix.MICS.USER.SOURCE

       +----------------------------------------------------+
       | Reference(s): Section 3.6.2                        |
       +----------------------------------------------------+


__  2. Verify MICF Full Execution


       Check MICF for full execution.

       Enter the CA MICS ISPF environment using the method
       selected by your site (e.g., LIBDEF, CLIST, or logon
       PROC).  Select the MICF option from the CA MICS
       Workstation Facility.  Review any new inquiries and
       execute MICF features.

       +----------------------------------------------------+
       |References: Chapter 2, MICF User Guide              |
       |            Chapter 3, ccc Analyzer Guide(s)        |
       |                        (REPORTS)                   |
       +----------------------------------------------------+


__  3. Verify Tape and DASD Management Systems


       Check that CA MICS has been properly established in
       your installation's tape and DASD management systems.

       +----------------------------------------------------+
       | Reference(s): Section 3.6.2                        |
       +----------------------------------------------------+


__  4. (OPTIONAL) Offsite Storage

       Arrange for offsite storage of CA MICS monthly backup
       tapes.
```

```
+----------------------------------------------------+
| Reference(s): Section 3.6.2                        |
+----------------------------------------------------+
```

3.8.5 Split an Existing Database Unit

This checklist defines the tasks required to implement a database split for an existing database unit. The checklist tasks are organized into the following two sections:

```
    1 - Existing Database Split Checklist
    2 - Finalize the Reconfigured Database Complex
```

3.8.5.1 Existing Database Split Checklist

__ 1. Suspend Operation

Suspend production operations for the unit database by using the Operational Status and Tracking SUSPEND command in MWF to deactivate production scheduling.

__ 2. Back Up Existing Database Unit

Use the Operational Status and Tracking RESUME, BACKUP, and SUSPEND commands in MWF to enable, backup, and block CA MICS operational jobs respectively.

Ensure that the backup job completed with a condition code of zero and the data sets were correctly cataloged.

```
+---------------------------------------------------+
| Reference(s): Section 4.3.4.4                     |
+---------------------------------------------------+
```

__ 3. Save Existing RESTORE Job

Save a copy of the existing RESTORE job.

This step is necessary to preserve the old version of the RESTORE job in case a restore is needed from the old database unit structure. For example, you can use the ISPF copy utility or ISPF edit to copy 'prefix.MICS.CNTL(RESTORE)' to 'prefix.MICS.CNTL(RSTROLD)'. The RESTORE job in prefix.MICS.CNTL will be regenerated by JCLGENE.

__ 4. Edit DBSPLIT

Edit prefix.MICS.PARMS(DBSPLIT) to add information areas and DDnames.

```
+---------------------------------------------------+
| Reference(s): Section 2.3.3.3.2.4                 |
+---------------------------------------------------+
```

__ 5. Execute JCLGENE

Generate the jobs affected by DBSPLIT. Submit the following job:

'prefix.MICS.CNTL(JCLGENE)'

```
+---------------------------------------------------+
| Reference(s): Section 2.3.3.1.1 and 3.5.3.3       |
+---------------------------------------------------+
```

__ 6. Execute GDGSGEN

Update/create GDG entries for installed products in the OS CVOL or VSAM catalog. Submit the following job:

'prefix.MICS.CNTL(GDGSGEN)'

```
+---------------------------------------------------+
| Reference(s): Section 3.5.6.3                     |
+---------------------------------------------------+
```

__ 7. Execute PROCJOB3

Move CA MICS database set-up PROCS to the installation PROCLIB by submitting the following job:

'prefix.MICS.CNTL(PROCJOB3)'

```
+---------------------------------------------------+
| Reference(s): Section 3.5.4                       |
+---------------------------------------------------+
```

__ 8. Allocate Additional SAS Data Libraries

Allocate the additional SAS data libraries needed for a split database unit.

The ALLOCATE job contains JCL statements for allocating the new database files in addition to the existing database files. You can allocate your new database files by making a copy of the ALLOCATE job, deleting all DISP=(NEW,CATLG) JCL statements that refer to existing data sets, leaving just the DD statements to allocate and initialize your new database files. Then fill in the UNIT, VOL=SER=, SPACE=, and/or other data set allocation parameters as required for your installation and database space requirements. Then submit the tailored job to allocate and initialize the data sets.

```
+---------------------------------------------------+
       | Reference(s): Section 3.5.6.2                      |
       +----------------------------------------------------+

__  9. Move Information Area Files

       Move information area files to new data libraries.

       Use PROC COPY to copy all of the files that belong to
       the information areas in these new libraries to the
       new data libraries (see DBSPLIT definition).  This
       copy operation can be done using a " SELECT iii:  ; "
       statement with PROC COPY, where iii is the information
       area ID.

__ 10. Delete Information Area Files

       Delete information area files in old data libraries.

       When you have verified that all the necessary files
       have been moved successfully to the new SAS data
       libraries, delete the original versions from the old
       data libraries using PROC DATASETS.

__ 11. Back Up New Database Unit Structure

       Use the Operational Status and Tracking RESUME,
       BACKUP, and SUSPEND commands in MWF to enable, backup,
       and block CA MICS operational jobs respectively.

       Ensure that the backup job completed with a condition
       code of zero and the data sets were correctly
       cataloged.

       +----------------------------------------------------+
       | Reference(s): Section 4.3.4.4                      |
       +----------------------------------------------------+

__ 12. Verify User SAS Code

       Change any user SAS code that refers to CA MICS files
       by using explicit DDnames to use &diiit macro
       variables instead.

       +----------------------------------------------------+
       | Reference(s): SAS Documentation                    |
       +----------------------------------------------------+
__ 13. Resume Operations


       Enable CA MICS operational jobs for this unit
       database.  Use the Operational Status and Tracking
       RESUME command in MWF to enable CA MICS operational
       jobs.

__ 14. Resume Daily Processing

       Use the Operational Status and Tracking DAILY command
       in MWF to resume daily processing.

       +---------------------------------------------------+
       | Reference(s): Section 3.5.7                       |
       +---------------------------------------------------+
```

3.8.5.2 Finalize the Reconfigured Database Complex

```
 ************************************************************
 *  The checklist in this section should only be completed  *
 *  after all database units have been reconfigured in      *
 *  the CA MICS complex.                                    *
 ************************************************************

__  1. Back Up Libraries

       Since CA MICS does not provide a standard backup of
       libraries, back up the libraries below so that
       recovery can be accomplished.

       Complex-level data sets:

          sharedprefix.MICS.ASM
          sharedprefix.MICS.BIN
          sharedprefix.MICS.CLIST
          sharedprefix.MICS.CNTL
          sharedprefix.MICS.DIC.TEXT
          sharedprefix.MICS.DOC.TEXT
          sharedprefix.MICS.GENLIB
          sharedprefix.MICS.HOLD.PARMS
          sharedprefix.MICS.HOLD.USER.SOURCE
          sharedprefix.MICS.INC.TEXT
          sharedprefix.MICS.INCLLIB
          sharedprefix.MICS.ISPHLIB
          sharedprefix.MICS.ISPMLIB
          sharedprefix.MICS.ISPPLIB
          sharedprefix.MICS.ISPSLIB
          sharedprefix.MICS.ISPTLIB
          sharedprefix.MICS.LIBLOAD.CNTL
          sharedprefix.MICS.LOAD
          sharedprefix.MICS.LOCALMOD.CNTL
          sharedprefix.MICS.MACAUTOS
          sharedprefix.MICS.MACLIB
          sharedprefix.MICS.MCOLIB
          sharedprefix.MICS.MWFPDS.DATA
          sharedprefix.MICS.MWFSAS.DATA
          sharedprefix.MICS.OBJ
          sharedprefix.MICS.PARMS
          sharedprefix.MICS.PROTOLIB
          sharedprefix.MICS.SASFLS
          sharedprefix.MICS.SCREENS
          sharedprefix.MICS.SOURCE
          sharedprefix.MICS.TABLES
          sharedprefix.MICS.TEST.INCLLIB
          sharedprefix.MICS.TEST.LOAD
          sharedprefix.MICS.TEST.SOURCE
          sharedprefix.MICS.USER.SOURCE


       Database unit data sets:

          prefix.MICS.CNTL
          prefix.MICS.MUOLIB
          prefix.MICS.PARMS
          prefix.MICS.USER.SOURCE

       +----------------------------------------------------+
       | Reference(s): Section 3.6.2                        |
       +----------------------------------------------------+

__  2. Verify MICF Full Execution

       Check MICF for full execution.

       Enter the CA MICS ISPF environment using the method
       selected by your site (e.g., LIBDEF, CLIST, or logon
       PROC).  Select the MICF option from the CA MICS
       Workstation Facility.  Review any new inquiries and
       execute MICF features.

       +----------------------------------------------------+
       |References: Chapter 2, MICF User Guide              |
       |            Chapter 3, ccc Analyzer Guide(s)        |
       |                        (REPORTS)                   |
       +----------------------------------------------------+

__  3. Verify Tape and DASD Management Systems

       Check that CA MICS has been properly established in
       your installation's tape and DASD management systems.

       +----------------------------------------------------+
       | Reference(s): Section 3.6.2                        |
       +----------------------------------------------------+

__  4. (OPTIONAL) Arrange Offsite Storage

       Arrange for offsite storage of CA MICS monthly backup
       tapes.

       +----------------------------------------------------+
       | Reference(s): Section 3.6.2                        |
       +----------------------------------------------------+
```

3.8.6 Loading New Products

```
************************ BACKGROUND *************************
*                                                           *
* The CA MICS tape or ESD PAX file includes code and        *
* documentation for all CA MICS products.  It also          *
* delivers maintenance and enhancements for your licensed   *
* CA MICS products as part of the CA MICS Product Support   *
* Program (PSP).                                            *
*                                                           *
* Use these instructions to install newly licensed CA MICS  *
* products into an existing CA MICS complex.  Since product *
* installation may need the most current CA MICS            *
* maintenance, the installation process will check to       *
* ensure that the most recent PSP maintenance has been      *
* loaded from the distribution medium.                      *
*                                                           *
* CA LMP keys are required for each product you install.    *
* You must ensure that the keys for new products have been  *
* installed on each CPU on which the new products will be   *
* installed.                                                *
*                                                           *
*************************************************************

To load the new product, complete the following steps:

__  1. Ensure Libraries are Cataloged

       Make sure that the following libraries are cataloged.
       If they are not, contact Technical Support at
       http://ca.com/support.

       The libraries are:

       sharedprefix.MICS.PSP.CNTL
       sharedprefix.MICS.PSP.PC.TEXT
       sharedprefix.MICS.PSP.ISPTLIB
       sharedprefix.MICS.TAPELOAD.CNTL

__  2. Ensure Libraries are Not Migrated

       Ensure your sharedprefix.MICS and
       sharedprefix.MICS.PSP libraries are not migrated.
       Their data set names begin with the following
       qualifiers:

       sharedprefix.MICS            (complex libraries)
       sharedprefix.MICS.PSP        (PSP libraries)
```

__ 3. Compress Libraries

The code and text for the products on your tape are copied directly to your sharedprefix.MICS and sharedprefix.MICS.PSP libraries. To avoid potential space problems, compress your sharedprefix.MICS and PSP libraries before loading the new product.

__ 4. Back Up Complex and PSP Libraries

The code and text for the products on the PSP distribution medium are copied directly to your sharedprefix.MICS and sharedprefix.MICS.PSP libraries. To avoid potential space problems, compress your sharedprefix.MICS and PSP libraries before loading the new product.

__ 5. Ensure PSP Libraries have been Refreshed

Ensure that your Product Support Program (PSP) libraries have been previously refreshed from the most recently available PSP distribution. If not, it would be best to refresh your PSP libraries before continuing.

__ 6. Ensure MWF is Initialized

Refer to the "How to Use the PSP" guide for detailed instructions on initializing the CA MICS Workstation Facility (MWF) and using PSP Online to apply product changes.

__ 7. Review the Cover Letter

Review the cover letter and associated product maintenance letters (PML) for the latest important technical information.

__ 8. Install LMP Keys for New Products

You are required to obtain and install a CA LMP key for each CA MICS product. One key is required for each CPU that the product will be loaded to. Refer to Section 3.2.2 for more information. If there are questions about CA LMP keys, contact Technical Support at http://ca.com/support.

__ 9. Load Optional Products

```
             Optional products can be loaded from either tape or
             ESD PAX.  There are two ways to load the products,
             either via the panel-driven load process or via
             pre-existing JCL.  Select one of the following
             substeps to load the product:

             Step 9a. - Panel-driven process that supports tape
                        and ESD PAX.

             Step 9b. - ESDPLD1 - JCL supporting ESD PAX

             Step 9c. - PRODLD1 - JCL supporting tape

       __ 9a. Load Optional Products Using MWF


             This step uses a panel-driven process to generate the
             JCL to load your optional products to DASD.  It
             combines parameters you provide on MWF panels with
             parameters that were specified during the
             complex-level JCLGEN, and generates the JCL to
             load from the PSP distribution data sets.

             __ Select the MWF panel called Select New Products and
                Submit the Product Load Job (MWF;5;2;3;4).

             __ Type the following parameters on the panel:

                SR      is the new product being loaded.  Type the
                        three-letter code of the new product.

                        If you are loading more than one new
                        product, each code must be separated by a
                        blank.

                        Note:  Only specify your NEW products!  If
                        you specify products that are already
                        installed, they will be overlayed.  This
                        can critically impact your CA MICS
                        database.

                        Use the MWF CONFIG command to help you
                        determine which CA MICS products are
                        installed.

                From    When the incoming PSP distribution files
                Tape    are on a tape, specify YES.  Otherwise,
                        specify NO or leave this blank.  Note that
```

```
        this option is mutually exclusive with the
        "From ESD-DASD" option below.

Tape    is the volume serial number of the PSP
VOLSER  distribution tape that you want to load.

Tape    is the expiration date for a foreign tape
EXPDT   if your tape management system requires
        one.

Tape    is the name of the tape device on which the
UNIT    CA MICS distribution tape is to be mounted.

From    When the CA MICS distribution files come
ESD-    from an ESD PAX file, specify YES.
DASD    Otherwise, specify NO or leave this blank.
        Note that this option is mutually exclusive
        with the "From Tape" option above.

High    is the high level data set qualifier(s)
Level   that are common to all the incoming PSP
Qualif- distribution files on DASD. It corresponds
ier     to "yourHLQ" used in the ESD PAX UNZIPJCL
        job.

PSP     is the volume serial number of the
VOLSER  direct access storage device to be used to
        store the new PSP libraries.  This will be
        used as the default VOLSER for allocation
        of new PSP data sets.  It can be overridden
        for individual data set names by modifying
        entries in PSPOVER as described in Appendix
        A.1.  If the "SMS JCL Tailoring" option
        (below) is "YES", this parameter is
        ignored.

PSP     is the unit name of the direct access
UNIT    storage device to be used to store the new
        PSP data sets.  This will be used as the
        default unit for allocation of new PSP data
        sets.  It can be overridden for individual
        data set names by modifying entries in
        PSPOVER as described in Appendix A.1.  If
        the "Omit UNIT & VOL=SER from JCL" option
        (below) is YES, this parameter is ignored.

PSP     is the SMS STORCLAS to be used to store the
STOR-   new PSP data sets.  This will be used as
CLAS    the default STORCLAS for allocation of new
```

PSP data sets. It can be overridden for individual data set names by modifying entries in PSPOVER as described in Appendix A.1.

PSP DATACLAS is the SMS DATACLAS to be used to store the new PSP data sets. This will be used as the default DATACLAS for allocation of new PSP data sets. It can be overridden for individual data set names by modifying entries in PSPOVER as described in Appendix A.1.

PSP MGMTCLAS is the SMS MGMTCLAS to be used to store the new PSP data sets. This will be used as the default MGMTCLAS for allocation of new PSP data sets. It can be overridden for individual data set names by modifying entries in PSPOVER as described in Appendix A.1.

PSP DSNTYPE can be either blank or LIBRARY. Specify LIBRARY to allocate new PSP data sets as PDSE data sets (partition data sets extended). It can be overridden for individual data set names except for the PSP LOAD library, by modifying entries in PSPOVER as described in Appendix A.1.

PSP DISP is the disposition of all PSP libraries except sharedprefix.MICS.PSP.CNTL, PSP.PC.TEXT, and PSP.ISPTLIB.

Note: If the libraries listed below exist, specify "OLD,KEEP". If the libraries do not exist, specify "NEW,CATLG,DELETE".

sharedprefix.MICS.PSP.ASM
sharedprefix.MICS.PSP.BIN
sharedprefix.MICS.PSP.CLIST
sharedprefix.MICS.PSP.DIC.TEXT
sharedprefix.MICS.PSP.DOC.TEXT
sharedprefix.MICS.PSP.GENLIB
sharedprefix.MICS.PSP.INC.TEXT
sharedprefix.MICS.PSP.INCLLIB
sharedprefix.MICS.PSP.ISPHLIB
sharedprefix.MICS.PSP.ISPMLIB
sharedprefix.MICS.PSP.ISPPLIB
sharedprefix.MICS.PSP.ISPSLIB

```
sharedprefix.MICS.PSP.LOAD
sharedprefix.MICS.PSP.MACAUTOS
sharedprefix.MICS.PSP.MACLIB
sharedprefix.MICS.PSP.OBJ
sharedprefix.MICS.PSP.PARMS
sharedprefix.MICS.PSP.PROTOLIB
sharedprefix.MICS.PSP.SASFLS
sharedprefix.MICS.PSP.SOURCE
sharedprefix.MICS.PSP.USOURCE
```

If your site does not have its PSP libraries, new PSP libraries will be allocated as part of the INSTALL steps. These PSP libraries will contain all product changes required to install your new products. For more information about the PSP libraries, see "How To Use the PSP".

PSP BLOCKSIZE is the block size that will be used to allocate your new PSP data sets.

Note: Do not override the default of 6160 unless required by your site.

Loader is the program name of the IBM Loader at your site. This is typically IEWLDRGO, HEWLDRGO, or LOADER. If in doubt, contact your system programmer.

Region size is the region for each step. The default of 4096 KB should be adequate.

Omit UNIT & VOL=SER From JCL will cause the JCL to be generated without "VOL=SER=" or "UNIT=" parameters in DD statements for temporary disk data sets or for allocating new PSP data sets. If the System Managed Storage (DFSMS) implementation at your site requires that UNIT= and VOL=SER= parameters be omitted, then specify YES, otherwise, specify NO.

Edit generated JCL is either YES or NO. If YES, then you will be given an opportunity to edit the load JCL before it is submitted.

__ When your parameters are complete, do one of the following:

__ Press END to continue

OR

__ Type CANCEL and press ENTER to stop the process.

__ If you pressed END and if Edit generated JCL was YES, then you will see a temporary data set containing the generated tape load job.

__ Review the JCL and verify that it conforms to your site's standards. Make a copy of the JCL in a permanent data set for future reference.

__ Press END to submit the job.

Note: Modules are written to disk by executing the program named LPEBLDD. It runs as a "non-authorized" program. Some mainframe security systems, such as CA Top Secret, have a feature that restricts non-authorized programs from writing authorized load modules. (An authorized load module is link-edited with an attribute of AC(1).) If your site has this security feature activated and is licensed for the CA MICS Space Collector (VCC), then the load job will fail with an S913 ABEND while trying to write an authorized VCC load module. To prevent this ABEND, request your security administrator to permit LPEBLDD to write authorized load modules.

__ Exit PSP Online. Do this by pressing END until the CA MICS Administrator Facility panel is displayed.

__ After the job finishes, check the return codes. All steps must have a return code of zero except the following:

COMPABND must not execute. It only executes if the PSP refresh requirement is not met. It will force the job to ABEND.

PSPCHK2 may be 0 or 4.

#ASCOPY2 and PSPUPDTE must be 0 when PSPCHK2 is 4.

#ASCOPY2 and PSPUPDTE will not execute when PSPCHK2 is 0. This is normal.

```
          Any other return codes are errors.  If there is an
          error, you must correct it and resubmit the job.
          If the error persists, contact Technical Support at
          http://ca.com/support.

       __ Proceed to Step 10.

__ 9b. Execute ESDPLD1

       This step supports loading your optional products to
       disk from an ESD PAX file.

  ***********************************************************
  *  CAUTION:  Do NOT EXECUTE STEP 9b if you have already   *
  *            executed Step 9a or Step 9c.                 *
  ***********************************************************

       __ Edit sharedprefix.MICS.TAPELOAD.CNTL(ESDPLD1).

          Provide a JOB card and complete the following JCL
          statements on the LOADPROD EXEC statement.

//***********************************************************
//LOADPROD EXEC LOADPROC,                                   *
//*                                   PARM                  *
//*                              -------------------        *
//    RSIZE='4096K',        <--- REGION FOR ALL STEPS       *
//    YOURHLQ='_____',      <--- MICS ESD PREFIX            *
//    SPREFIX='_____',      <--- MICS SHARED PREFIX         *
//    SMICS='MICS.',        <--- SHARED MICSLEVEL?          *
//    SASPFX='______',      <--- SAS SYSTEM LIBRARY PREFIX  *
//    SASNAME='entry',      <--- SAS     for SAS V9         *
//*                              SASXAL  for SAS V8         *
//    SASCFG='config',      <--<< BATCH IF SAS V8           *
//*                               BATW0 IF SAS V9           *
//    LANG1='',             <--<< SAS V9 LANGUAGE           *
//*                               EN = ENGLISH              *
//*                               BLANK IF V8               *
//    LANG2='',             <--<< SAS V9 ENCODING VALUE     *
//*                               W0. = EBCDIC (1047)       *
//*                                MUST HAVE TRAILING .     *
//*                                BLANK IF V8              *
//    PSPVOL=,              <--- VOLUME FOR NEW PSP LIBS    *
//    PSPDSP='_____',       <--- PSP LIB DISPOSITION        *
//    PSPUNIT=,             <--- UNIT FOR NEW PSP LIBS      *
//    PSPSTORC=,            <--  OPTIONAL PSP STORCLAS=     *
//    PSPDATAC=,            <--  OPTIONAL PSP DATACLAS=     *
//    PSPMGMTC=,            <--  OPTIONAL PSP MGMTCLAS=     *
//    PSPDSNT=,             <--  OPT., PUT LIBRARY FOR PDSE *
//    BLOCK=6160,           <--- PSP LIBRARY BLOCKSIZE OK?  *
//    WKUNIT=SYSDA ,        <--- IS WORK DASD UNIT OK?      *
//    SR='___',             <--- NEW PRODUCTS. SEPARATE WITH*
//*                              A BLANK. ENCLOSE WITHIN    *
//*                              APOSTROPHES.               *
//    LOADER=IEWLDRGO       <--- YOUR SITE'S LOADER NAME    *
//***********************************************************

         You must complete the following parameters on the
         LOADPROD EXEC statement, not those on the PROC
         statement:

         RSIZE   is the region for each step. The default of
                 4 MB should be adequate.

         YOURHLQ specifies the high level qualifier of the
                 ESD PAX data sets. It must be replaced with
                 the "yourHLQ" used in the ESD PAX UNZIPJCL
                 job.

         SPREFIX is the standard CA MICS sharedprefix. Do
                 not end the sharedprefix with a period (.).

         SMICS   is the MICS level for the sharedprefix.MICS
                 and sharedprefix.MICS.PSP libraries.
                 Specify this parameter with a trailing
                 period.  If you have specified NOMICSLEVEL
                 on the sharedprefix statement in the JCLDEFC
                 member of sharedprefix.MICS.PARMS, you must
                 change SMICS to null (SMICS=,).

         SASPFX  specifies the common data set name prefix
                 for your SAS libraries. Note that this name
                 is specified with no trailing period.  This
                 prefix is used in the following DDs to form
                 the names of your site's SAS libraries used
                 for CA MICS:

                 STEPLIB - SAS load library
                 CONFIG  - SAS CNTL library that holds a
                           SAS system configuration file
                 SASHELP - SAS help library
                 SASMSG  - SAS message library

                 Review the data set names specified in the
                 DDs.  If they are incorrect, correct the
                 names and comment out SASPFX.
```

```
SASNAME specifies the program name of the SAS
        Program Product in the load library
        specified in the STEPLIB DD.

SASCFG  is the member name of your SAS configuration
        file.  Normally this is a member of the SAS
        CNTL data set.  For SAS version 8, the
        member name is usually BATCH.  For version 9
        and above it is usually BATW0.

LANG1   is the language code required with SAS
        Versions 9 and higher, for example, EN
        stands for English. The value is blank for
        SAS Version 8.

LANG2   is the encoding value required with SAS
        Version 9 and higher.  The value for EBCDIC
        is W0 and must have a trailing period (.).
        The value is blank for SAS Version 8.

PSPUNIT specify the unit name and volume serial
and     number of the direct access storage device
PSPVOL  to be used to store the new PSP libraries.
        These parameters will be used as default
        parameters for allocation of new PSP data
        sets. They can be overridden for individual
        data set names by modifying entries in
        PSPOVER as described in Appendix A.1.  If
        you do not need to specify a parameter for
        PSPUNIT, code it as null (that is,
        'PSPUNIT=,').  Likewise, if you do not need
        to specify a parameter for PSPVOL, code it
        as null (that is, 'PSPVOL=,').

PSP     is the SMS STORCLAS to be used to store the
STORC   new PSP libraries.  This will be used as
        the default STORCLAS for allocation of new
        PSP data sets. It can be overridden for
        individual data set names by modifying
        entries in PSPOVER as described in Appendix
        A.1.  If you do not need to specify a
        parameter for PSPSTORC, code it as null
        (that is, 'PSPSTORC=,').

PSP     is the SMS DATACLAS to be used to store the
DATAC   new PSP libraries.  This will be used as
        the default DATACLAS for allocation of new
        PSP data sets. It can be overridden for
        individual data set names by modifying
        entries in PSPOVER as described in Appendix
        A.1.  If you do not need to specify a
        parameter for PSPDATAC, code it as null
        (that is, 'PSPDATAC=,').

PSP     is the SMS MGMTCLAS to be used to store the
MGMTC   new PSP libraries.  This will be used as
        the default MGMTCLAS for allocation of new
        PSP data sets. It can be overridden for
        individual data set names by modifying
        entries in PSPOVER as described in Appendix
        A.1.  If you do not need to specify a
        parameter for PSPMGMTC, code it as null
        (that is, 'PSPMGMTC=,').

PSP     can be either blank or LIBRARY.  Specify
DSNT    LIBRARY to allocate new PSP data sets as
        partitioned data set extended (PDSE) data
        sets.  It can be overridden for individual
        data set names except for the PSP LOAD
        library, by modifying entries in PSPOVER
        as described in Appendix A.1.

PSPDSP  is the disposition of all PSP libraries
        except sharedprefix.MICS.PSP.CNTL,
        PSP.PC.TEXT, and PSP.ISPTLIB.  If the
        following libraries exist, use the value
        'OLD,KEEP'.  If the following libraries do
        not exist, use the value 'NEW,CATLG,DELETE'.
        Verify that you have the
        sharedprefix.MICS.PSP.CNTL, PSP.PC.TEXT, and
        PSP.ISPTLIB libraries already allocated.

        sharedprefix.MICS.PSP.ASM
        sharedprefix.MICS.PSP.BIN
        sharedprefix.MICS.PSP.CLIST
        sharedprefix.MICS.PSP.DIC.TEXT
        sharedprefix.MICS.PSP.DOC.TEXT
        sharedprefix.MICS.PSP.GENLIB
        sharedprefix.MICS.PSP.INC.TEXT
        sharedprefix.MICS.PSP.INCLLIB
        sharedprefix.MICS.PSP.ISPHLIB
        sharedprefix.MICS.PSP.ISPMLIB
        sharedprefix.MICS.PSP.ISPPLIB
        sharedprefix.MICS.PSP.ISPSLIB
        sharedprefix.MICS.PSP.LOAD
        sharedprefix.MICS.PSP.MACAUTOS
        sharedprefix.MICS.PSP.MACLIB
        sharedprefix.MICS.PSP.OBJ
```

sharedprefix.MICS.PSP.PARMS
sharedprefix.MICS.PSP.PROTOLIB
sharedprefix.MICS.PSP.SASFLS
sharedprefix.MICS.PSP.SOURCE
sharedprefix.MICS.PSP.USOURCE

WKUNIT specifies the unit name for a direct access work device.

BLOCK is the block size value used for the new PSP libraries. Do not override the default of 6160 unless required by your site.

SR is the three-letter code for the new product being loaded.

If you are loading more than one new product, each code must be separated by a blank.

Note: Only specify your NEW products! If you specify products that are already installed, they will be overlayed. This can critically impact your CA MICS database.

Use the MWF CONFIG command to help you determine which CA MICS products are installed.

LOADER specifies the program name of the IBM Loader at your site. This is typically IEWLDRGO, HEWLDRGO, or LOADER. If in doubt, contact your system programmer.

__ Submit 'sharedprefix.MICS.TAPELOAD.CNTL(ESDPLD1)'.

Note: Modules are written to disk by executing the program named LPEBLDD. It runs as a "non-authorized" program. Some mainframe security systems, such as CA Top Secret, have a feature that restricts non-authorized programs from writing authorized load modules. (An authorized load module is link-edited with an attribute of AC(1)). If your site has this security feature activated and is licensed for the CA MICS Space Collector (VCC), then the load job will fail with an S913 ABEND while trying to write an authorized VCC load module. To prevent this ABEND, request your security administrator to permit LPEBLDD to write

```
          authorized load modules.

       __ After the job finishes, check the return codes.
          All steps must have return codes of zero except the
          following:

          COMPABND must not execute.  If it does execute, the
          PSP refresh level requirement has not been met and
          it will force the job to ABEND.

          PSPCHK2 may be 0 or 4.

          #ASCOPY2 and PSPUPDTE must be 0 if PSPCHK2 is 4.

          #ASCOPY2 and PSPUPDTE will not execute if PSPCHK2
          is 0.  This is normal.

          Any other return codes are errors.  If there is an
          error, you must correct it and resubmit the job.
          If the error persists, contact Technical Support at
          http://ca.com/support.

       __ Proceed to Step 10.

__ 9c. Execute PRODLD1

       This step supports loading your optional products to
       disk from a magnetic tape.

  ***********************************************************
  *  CAUTION:  Do NOT EXECUTE STEP 9c if you have already   *
  *            completed step 9a or Step 9b.                *
  ***********************************************************

       __ Edit sharedprefix.MICS.TAPELOAD.CNTL(PRODLD1).

          Provide a JOB card and complete the following JCL
          statements on the LOADPROD EXEC statement.

//***********************************************************
//LOADPROD EXEC LOADPROC,                                   *
//*                                       PARM              *
//*                                  --------------         *
//  RSIZE='4096K',            <--- REGION FOR ALL STEPS     *
//  SPREFIX='______________', <--- MICS SHARED PREFIX       *
//  SMICS='MICS.',            <--- SHARED MICSLEVEL?        *
//  TUNIT='____',             <--- IS TAPE UNIT OK?         *
//  TEXPDT=98000,             <--- IS TAPE EXPDT OK?        *
//  TVOL=______,              <--- TAPE VOLSER              *
//  SASPFX='_______________', <--- SAS SYSTEM LIBRARY PREFIX*
//  SASNAME=SASXAL,           <--- SAS PROGRAM NAME         *
//  SASCFG='config',          <--<< BATCH IF SAS V8         *
//*                                 BATW0 IF SAS V9         *
//  LANG1='',                 <--<< SAS V9 LANGUAGE         *
//*                                 EN = ENGLISH            *
//*                                 BLANK IF V8             *
//  LANG2='',                 <--<< SAS V9 ENCODING VALUE   *
//*                                 W0. = EBCDIC (1047)     *
//*                                 MUST HAVE TRAILING .    *
//*                                 BLANK IF V8             *
//  PSPVOL=,                  <--- VOLUME FOR NEW PSP LIBS  *
//  PSPDSP='________',        <--- PSP LIB DISPOSITION      *
//  PSPUNIT=,                 <--- UNIT FOR NEW PSP LIBS    *
//  PSPSTORC=,                <--  OPTIONAL PSP STORCLAS=   *
//  PSPDATAC=,                <--  OPTIONAL PSP DATACLAS=   *
//  PSPMGMTC=,                <--  OPTIONAL PSP MGMTCLAS=   *
//  PSPDSNT=,                 <--  OPT, PUT LIBRARY for PDSE*
//  BLOCK=6160,               <--- PSP LIBRARY BLOCKSIZE OK?*
//  WKUNIT=SYSDA,             <--- IS WORK DASD UNIT OK?    *
//  SR='___',                 <--- NEW PRODUCTS.SEPARATE    *
//*                                WITH A BLANK. ENCLOSE    *
//*                                WITHIN APOSTROPHES.      *
//  LOADER=IEWLDRGO           <--- YOUR SITE'S LOADER NAME  *
//***********************************************************
```

You must complete the following parameters on the LOADPROD EXEC statement, not those on the PROC statement:

RSIZE   is the amount of region storage for steps in this job.

SPREFIX   is the sharedprefix for the shared CA MICS and PSP libraries. Note that this is specified with no trailing period (for example, DB1.SP rather than DB1.SP.).

SMICS   is the MICS level for the sharedprefix.MICS and sharedprefix.MICS.PSP libraries. Specify this parameter with a trailing period. If you have specified NOMICSLEVEL on the sharedprefix statement in the JCLDEFC member of sharedprefix.MICS.PARMS, you must change SMICS to null (SMICS=,).

TUNIT   is the unit name of the tape device on which the CA MICS distribution tape is to be mounted.

TEXPDT   is the tape expiration date, if your site uses a tape management system.

TVOL   is the volume serial number of the CA MICS distribution tape as it appears on the "CA MICS Encryption Parameters."

SASPFX   is the common data set name prefix for your SAS libraries. Note that this name is specified with no trailing period. This prefix is used in the following DDs to form the names of your site's SAS libraries for CA MICS:

```
STEPLIB - SAS load library
CONFIG  - SAS CNTL library that holds a
          SAS system configuration file
SASHELP - SAS help library
SASMSG  - SAS message library
```

Review the data set names specified in the DDs. If they are incorrect, correct the names and comment out SASPFX.

SASNAME   is the program name of the SAS Program Product in the load library specified in the STEPLIB DD.

SASCFG   is the member name of your SAS configuration file. Normally this is a member of the SAS CNTL data set. For SAS Versions 7 and 8, the member name is usually BATCH and BATW0 for Versions 9 and above.

LANG1   is the language code required with SAS Versions 9 and higher, for example, EN stands for English. The value is blank for SAS Version 8.

LANG2   is the encoding value required with SAS Version 9 and higher. The value for EBCDIC is W0 and must have a trailing period (.). The value is blank for Sas Version 8.

PSPVOL   is the name of the DASD volume where the new PSP libraries will be located. If they already exist, code this parameter as null (that is, PSPVOL=,).

PSPDSP is the disposition of all PSP libraries except sharedprefix.MICS.PSP.CNTL, PSP.PC.TEXT, and PSP.ISPTLIB. If the following libraries exist, use the value 'OLD,KEEP'. If the following libraries do not exist, use the value 'NEW,CATLG,DELETE'. Verify that you have the sharedprefix.MICS.PSP.CNTL, PSP.PC.TEXT, and PSP.ISPTLIB libraries already allocated.

sharedprefix.MICS.PSP.ASM
sharedprefix.MICS.PSP.BIN
sharedprefix.MICS.PSP.CLIST
sharedprefix.MICS.PSP.DIC.TEXT
sharedprefix.MICS.PSP.DOC.TEXT
sharedprefix.MICS.PSP.GENLIB
sharedprefix.MICS.PSP.INC.TEXT
sharedprefix.MICS.PSP.INCLLIB
sharedprefix.MICS.PSP.ISPHLIB
sharedprefix.MICS.PSP.ISPMLIB
sharedprefix.MICS.PSP.ISPPLIB
sharedprefix.MICS.PSP.ISPSLIB
sharedprefix.MICS.PSP.LOAD
sharedprefix.MICS.PSP.MACAUTOS
sharedprefix.MICS.PSP.MACLIB
sharedprefix.MICS.PSP.OBJ
sharedprefix.MICS.PSP.PARMS
sharedprefix.MICS.PSP.PROTOLIB
sharedprefix.MICS.PSP.SASFLS
sharedprefix.MICS.PSP.SOURCE
sharedprefix.MICS.PSP.USOURCE

PSPUNIT is the unit name of the DASD volumes where the new PSP libraries will be located. If PSP libraries already exist, code this parameter as null (that is, PSPUNIT=,).

PSP STORC is the SMS STORCLAS to be used to store the new PSP libraries. This will be used as the default STORCLAS for allocation of new PSP data sets. It can be overridden for individual data set names by modifying entries in PSPOVER as described in Appendix A.1. If you do not need to specify a parameter for PSPSTORC, code it as null (that is, 'PSPSTORC=,').

PSP DATAC

is the SMS DATACLAS to be used to store the new PSP libraries. This will be used as the default DATACLAS for allocation of new PSP data sets. It can be overridden for individual data set names by modifying entries in PSPOVER as described in Appendix A.1. If you do not need to specify a parameter for PSPDATAC, code it as null (that is, 'PSPDATAC=,').

PSP MGMTC

is the SMS MGMTCLAS to be used to store the new PSP libraries. This will be used as the default MGMTCLAS for allocation of new PSP data sets. It can be overridden for individual data set names by modifying entries in PSPOVER as described in Appendix A.1. If you do not need to specify a parameter for PSPMGMTC, code it as null (that is, 'PSPMGMTC=,').

PSPDSNT

can be either blank or LIBRARY. Specify LIBRARY to allocate new PSP data sets as partitioned data set extended (PDSE) data sets. It can be overridden for individual data set names except for the PSP LOAD library, by modifying entries in PSPOVER as described in Appendix A.1.

BLOCK

is the block size value used for the new PSP libraries.

WKUNIT

is the unit name for a direct access work device.

SR

is the three-letter code for the new product being loaded.

If you are loading more than one new product, each code must be separated by a blank.

Note: Only specify your NEW products! If you specify products that are already installed, they will be overlayed! This can critically impact your CA MICS database.

Use the MWF CONFIG command to help you determine installed CA MICS products.

LOADER   is the program name of the IBM Loader at your site. This is typically IEWLDRGO, HEWLDRGO, or LOADER. If in doubt, contact your system programmer.

__ Submit the following job:

```
'sharedprefix.MICS.TAPELOAD.CNTL(PRODLD1)'
```

Note: Modules are written to disk by executing the program named LPEBLDD. It runs as a "non-authorized" program. Some mainframe security systems, such as CA Top Secret, have a feature that restricts non-authorized programs from writing authorized load modules. (An authorized load module is link-edited with an attribute of AC(1)). If your site has this security feature activated and is licensed for the CA MICS Space Collector (VCC), then the load job will fail with an S913 ABEND while trying to write an authorized VCC load module. To prevent this ABEND, request your security administrator to permit LPEBLDD to write authorized load modules.

__ After the job finishes, check the return codes. All steps must have return codes of zero except the following:

COMPABND must not execute. If it does execute, the PSP refresh level requirement has not been met and it will force the job to ABEND.

PSPCHK2 may be 0 or 4.

#ASCOPY2 and PSPUPDTE must be 0 if PSPCHK2 is 4.

#ASCOPY2 and PSPUPDTE will not execute if PSPCHK2 is 0. This is normal.

Any other return codes are errors. If there is an error, you must correct it and resubmit the job. If the error persists, contact Technical Support at http://ca.com/support.

__ Proceed to the next step.

```
__ 10. Receive New Tables

       NOTE: This step is ESSENTIAL for product installation!

       Use PSP Online to receive new product tables.  Select
       the Product Support Online Services panel (MWF;5;2).

       The Receive New PSP Tables dialog will be displayed.
       Respond to the prompts to ensure that the tables are
       received and saved.

__ 11. Complete Optional Product Installation

       Note: This step is ESSENTIAL for product installation.

       Use PSP Online to complete product installation.

       The ccc0000 job, which you are about to execute, helps
       ensure that required product maintenance is applied.
       It also copies product parameters, account routines,
       and MICF inquiries to their required libraries.  This
       must be done for the product to execute correctly.

       Use PSP Online to select and print product change
       ccc0000 (where "ccc" is the three-character identifier
       listed in your new product shipment letter).

       Note: When the ccc0000 product change is selected, all
       unapplied prerequisite and corequisite product changes
       will be selected as well.  You must ALSO print the
       checklists for these product changes.

       If you are reinstalling an optional product and
       ccc0000 does not appear on the Select Product Changes
       for Apply panel in PSP Online, perform the following
       steps to select the product change:

       __ Type "X ccc0000" on the command line and press
          ENTER.

       __ Type "R" on the line next to ccc0000 and press
          ENTER.

       __ Type "S" on the line next to ccc0000 and press
          ENTER.

       __ If you experience difficulty with this process,
          contact Technical Support.
```

Note: Review all PC text documents for requisite product changes and for possible local modification conflicts before applying any product changes.

Apply the product change. You must ensure that the ccc0000 installation job ends with a condition code of zero.

For assistance, contact Technical Support at http://ca.com/support.

__ 12. Complete Checklists for Each Optional Product

You must ensure that ALL checklists, including those used for the optional product's installation and its prerequisite and corequisite product changes, are completed in the following order:

__ Each prerequisite product change

__ The ccc0000 product change

__ Each corequisite product change

__ 13. Back Up System

Create a backup copy of all sharedprefix and prefix libraries.

__ 14. Retain Distribution Tape

Retain your CA MICS distribution files. They may be used to install other CA MICS optional products.

Chapter 4: Operation

The Operation chapter explains the activities involved in operating CA MICS and how to perform them. The chapter is made up of the following sections:

1 - Overview: Introduces you to the logic behind CA MICS
               operations (why things must be done).

2 - Guidelines: Provides task lists for operational
                 processes (what things must be done).

3 - Reference: Provides details that help you perform each
                task (how to do the things that must be
                done).

4 - Administration: Discusses administrative tasks the
                     System Administrator needs to perform.

5 - CA MICS Data Integration: Describes the integration of
                 CA MICS data with NSM.

6 - NSM Data Integration: Describes the integration of NSM
                 data with CA MICS.

7 - CA MICS Data Transfer: Provides instructions for
                 installing the CA MICS Data Transfer
                 Utility.

8 - CA MICS Documentation Conversion: Provides instructions
                 for converting the CA MICS documentation to
                 a Web viewable format.

You should read the Overview section before continuing with CA MICS installation. The Overview section introduces you to operational facilities and concepts.

The Operational Guidelines section is a high level guide or road map to CA MICS production operation. You should browse the Operational Guidelines right after reading the Overview. Then refer to the Operational Guidelines whenever you perform CA MICS operational processing or recovery.

Use the Reference section when you need more information. The Operational Guidelines will instruct you to perform a task. The Reference will tell you exactly what must be done.

The Administration section is both an overview and a reference for CA MICS administration activities. You should browse the Administration section and then use it as a reference when you perform administrative activities.

If your site uses NSM for network management, Sections 5, 6, and 7 will help you to exploit the tools provided with CA MICS for integrating NSM data and CA MICS data. Section 5 describes how you can add CA MICS data to your NSM processing. Section 6 explains how NSM data can be integrated into CA MICS processing. Section 7, as stated above, provides instructions for installing the CA MICS Data Transfer utility.

If your site uses USS and HFS facilities, section 8 will help you make CA MICS documentation available via Web Browser (e.g., Internet Explorer or Netscape).

This section contains the following topics:

4.1 Overview

To help you operate CA MICS, this overview describes the significance of operational procedures (why things must be done). You should read this entire section before proceeding to Operational Guidelines, Reference, or Administration.

A CA MICS complex is one installed copy of CA MICS. It consists of the following:

- One or more CA MICS unit databases

- sharedprefix and prefix libraries - programs, JCL, CLISTs, parameter definitions, and documentation

- Application Databases - SAS data files not subject to time sequence data, summarization, or aging rules. The TABLES data area (or file) is one example. The Change Tracking File provided in the Batch and Operations Analyzer's Operations Information Area is stored in the TABLES area.

The CA MICS Database is updated periodically (daily, weekly, monthly, and yearly). The update processes must be closely monitored to ensure that they run serially: each update must complete successfully before the next update is started.

The following topics explain this requirement and describe facilities to ensure that sequential updates take place:

1 - CA MICS Database Organization
2 - Operational Processes, Jobs, and Steps
3 - Processing Schedule
4 - Restart Facility
5 - Select and Force Facilities
6 - Checkpoint File
7 - Database Update Audit Trail
8 - Operational Status and Tracking Facility
9 - Administrative Activities

4.1.1 CA MICS Database Organization

The Database contains one PRIMARY unit database and multiple unit databases (units) to meet an installation's needs. Optionally, the Database contains one or more TEST and/or SPECIAL units. We recommend installing a TEST unit database in the complex for testing updates to products prior to placing the updates into production. A SPECIAL unit database can be installed in a complex if the need arises to perform special studies. TEST and SPECIAL units are not included in scheduled update processes.

The unit database is the smallest operational element in a CA MICS complex. Each unit database functions and is updated independently, so:

o updates can be run in parallel, ensuring database availability by shortening the elapsed update time.

o updates can be scheduled around the availability of input data.

You can organize and protect unit databases by areas of responsibility such as Network, IMS, or CICS.

Data volume is another consideration in configuring your unit databases. Since database update processing for multiple unit databases can execute concurrently, you can process more input data in less time by spreading this processing over multiple unit databases. For example, you might define separate unit databases for different CICS regions (or applications), thereby spreading your large volume CICS processing over multiple unit databases that can be updated concurrently.

You can also use the CA MICS incremental update facility to spread your daily database update processing over multiple, smaller incremental updates executed throughout the day. However, incremental update introduces additional complexity in operations and scheduling, and may result in processing large batch jobs during your installation's "prime shift." In general, you should address data volume issues by spreading the workload over multiple unit databases before using incremental update to address data volume issues. You may also be able to reduce the impact of large input data volumes through careful review of your CA MICS parameters, especially account code structures, to ensure that you are collecting and keeping only the data you need and want.

Standard CA MICS reporting facilities are designed to increase database availability by giving you the option to automatically offload report processing from the database update operational jobs to stand-alone reporting jobstreams that access the database with DISP=SHR. For example, the DAILYRPT, WEEKRPT, and MONTHRPT operational jobs can execute concurrently with end-user online database inquiry and any other "read-only" processing.

Further database availability is provided by the option to offload weekly and monthly archive tape processing from the WEEKLY and MONTHLY operational jobs. When requested, CA MICS will automatically submit stand-alone AUDIT, HISTW, and/or HISTM jobs to perform Archive Audit, Archive Weekly History, and/or Archive Monthly History processing. These jobs access the database with DISP=SHR and can run concurrently with CA MICS report processing jobs, end-user online database inquiry, and any other "read-only" processing.

4.1.2 Operational Processes, Jobs, and Steps

An operational process is a series of batch jobs that make up a logical unit of work. Operational processes update and maintain the CA MICS Database.

There are four operational processes in CA MICS:

- DAILY process - the DAILY job followed by the BACKUP job

  or optionally

  one or more SPLITSMF (optional) and INCRccc jobs followed by the DAILY job and the BACKUP job.

- WEEKLY process - the DAILY job followed by the WEEKLY job followed by the BACKUP job

- MONTHLY process - the DAILY job followed by the MONTHLY job followed by the BACKUP job

- YEARLY process - the DAILY job followed by the YEARLY job followed by the BACKUP job

CA MICS provides standard operational jobs for updating, reporting, maintaining, and recovering its database:

- DAILY - run each day to update the DETAIL and DAYS timespan files plus week-to-date and month-to-date files in the WEEKS and MONTHS timespans

- INCRccc - (optional) run one or more times a day for each product for which you have activated the CA MICS incremental update facility. The INCRccc jobs update the product's incremental update DETAIL and DAYS level files, which are then "rolled up" to the DETAIL and DAYS timespan files by the DAILY job.

- SPLITSMF - (optional) run one or more times a day to split the SMF input data into multiple files for processing in the INCRccc jobs. SPLITSMF is a stand-alone version of the DAILY job's DAYSMF step and applies only to those products which take their input from the SMF files, and which are marked as INCRUPDATE YES and INCRSPLIT USE in prefix.MICS.PARMS(cccOPS).

- o WEEKLY - run once each week after the DAILY job for WEEKS timespan cycle close-out, weekly archive audit, and weekly archive history processing

- o MONTHLY - run once each month after the DAILY job for MONTHS timespan cycle close-out and monthly archive history processing and for updating year-to-date files

- o YEARLY - run once each year after the MONTHLY job for YEARS timespan cycle close-out

- o BACKUP - run daily, bi-daily, or weekly (per user-specified backup frequency) to generate a tape backup of the entire database

- o SCHEDULE - run each day to submit scheduled processing

- o RESTORE - run whenever the database is damaged or must be recovered

- o AUDIT - (optional) run after WEEKLY to perform optional Archive Audit processing. The AUDIT job is used when you specify

  ```
  ARCHIVE AUDIT YES JOB
  ```

  in prefix.MICS.PARMS(JCLDEF).

  NOTE: You can execute AUDIT more frequently (e.g., twice a week or daily) if DASD space is inadequate for retaining sufficient DETAIL/DAYS cycles for weekly audit tape creation.

  In this case, it is necessary to add the following to prefix.MICS.PARMS(EXECDEF)

  ```
  USERDEF AUDITCWK YES
  ```

  This parameter overrides the default audit archive processing so that data for the current week is retained and copied to the new audit archive tape cycle.

- HISTW - (optional) run after WEEKLY to perform optional Archive Weekly History processing. The HISTW job is used when you specify

  ```
  ARCHIVE HISTW YES JOB
  ```

  in prefix.MICS.PARMS(JCLDEF).

- HISTM - (optional) run after MONTHLY to perform optional Archive Monthly History processing. The HISTM job is used when you specify

  ```
  ARCHIVE HISTM YES JOB
  ```

  in prefix.MICS.PARMS(JCLDEF).

- IUDBINIT - (optional) run to re-initialize in-progress incremental update processing after restoring the unit database files (i.e., after running the RESTORE job). Messages in the RESTORE job MICSLOG will prompt you to execute IUDBINIT when needed.

- DAILYRPT - run after DAILY to produce daily production reports

- WEEKRPT - run after WEEKLY to produce weekly production reports

- MONTHRPT - run after MONTHLY to produce monthly production reports

- RSTRTBLS - run to restore the TABLES and SCREENS data sets

- RSTRTLIB - run to restore ISPF-based information

- DAYSMFR - run as necessary to recreate DAILY job work files normally populated by the DAYSMF step.

- ACTDAY1R - run as necessary to restore the CA MICS Accounting and Chargeback DAY1 audit file

- RSTATUS - run as necessary to update Operational Status and Tracking control tables or to replace a lost Run Status Report

These jobs are generated as part of the installation process because each job is tailored to the installation's environment. Individual CA MICS products may also generate additional operational jobs for unique, product-specific processing.

The CA MICS operational jobs in each unit database are unique. Job steps are defined based on the products installed in the unit. For example, the DAILY job for a unit that contains the Hardware and SCP Analyzer (step 20, component identifier RMF) and the Batch and Operations Analyzer (step 30, component identifier SMF) would contain the following steps:

```
o  DAYALL - allocates work file space
o  DAYSMF - selects and splits SMF input data to work files
            for processing by Database update steps
o  DAY020 - updates the RMF product Database files
o  DAY030 - updates the SMF product Database files
o  DAY200 - updates exceptions (EXC02nnn for RMF and
            EXC03nnn for SMF)
o  DAY400 - submits daily report processing
o  DAY500 - processes non-standard user reporting, if any
o  DAY900 - terminates the daily job and frees work file
            space
o  DAYRSR - produces the Run Status Report
```

CA MICS operational jobs are built by unit and made up of independent steps for the following reasons:

o Reduced run time. Separate jobs for each unit enable CA MICS Database updates to run concurrently. For example, for a complex containing two units (unit A having the SMF, RMF, and SNT analyzers and unit B having the IMS and DB2 analyzers), unit A's update can be processed while unit B is being updated, because the units are independent.

If the units were dependent, the time needed to update the database would be greater, because unit A would need to complete its processing before unit B's could begin.

NOTE: With the CA MICS incremental update facility, individual product incremental updates can execute concurrently with incremental updates for other products in the unit database. However, end-of-day processing for the unit database is still a serial process, and if the units were dependent, unit A would need to complete its end-of-day processing before unit B's could begin.

- Input data independence. Separate jobs for each unit enable Database update scheduling based on input data availability. In the prior example, unit A's update can be processed before the IMS and DB2 input data used to update unit B are available.

  When all products are installed in a single unit database, no updates can occur until ALL input data is available.

  NOTE: With the CA MICS incremental update facility, individual product incremental updates can occur when the product's data becomes available. Individual product incremental updates are independent of any other product's processing in the unit database. However, the end-of-day, DAILY job, can not execute until ALL products are ready for end-of-day processing.

- Flexibility. The ability to restart after a failure at the job step that failed saves:

  - Time, because the work that had been completed prior to the failed step is retained

  - Resources, because the system does not reprocess work it had successfully completed

  For example, if an operational job fails part way through processing due to an I/O error with the input tape, the work completed in prior steps is retained and the job can be restarted at the step that failed when the error is corrected.

  If CA MICS operational jobs were not step-restartable, the entire run would be lost. In a unit database that contains multiple products, this loss could be very significant.

4.1.3 Processing Schedule

The CA MICS Database must be updated on a timely and progressive basis to ensure complete and accurate data. A processing schedule is required to ensure that the appropriate operational processes take place. The schedule establishes that the jobs that make up these processes occur on a daily, weekly, monthly, and yearly basis.

o Execute the DAILY process each day (except when the WEEKLY, MONTHLY, or YEARLY process is scheduled--WEEKLY, MONTHLY, and YEARLY include DAILY).

  You may also choose to enable incremental update for one or more products, in which case you might execute the SPLITSMF and/or INCRccc jobs (by product) at scheduled points during the day, and then run the DAILY process at night for end-of-day close-out and reporting.

  NOTE: You can optionally offload DAILY reporting to the stand-alone DAILYRPT job, thereby reducing DAILY job elapsed time and increasing database availability. You can use the CA MICS AUTOSUBMIT option to automatically submit the DAILYRPT job, or you can schedule daily reporting externally.

o Execute the WEEKLY process at the beginning of each new week (the WEEKLY process is composed of the DAILY, WEEKLY, and BACKUP operational jobs).

  NOTE: You can optionally offload WEEKLY reporting to the stand-alone WEEKRPT job, Archive Audit tape processing to the stand-alone AUDIT job, and/or Archive Weekly History processing to the stand-alone HISTW job, thereby reducing WEEKLY job elapsed time and increasing database availability. You can use CA MICS AUTOSUBMIT options to automatically submit the WEEKRPT, AUDIT, and/or HISTW jobs, or you can schedule weekly reporting and/or weekly archive processing externally.

o Execute the MONTHLY process at the beginning of each new month (the MONTHLY process is composed of the DAILY, MONTHLY, and BACKUP operational jobs).

  NOTE: You can optionally offload MONTHLY reporting to the stand-alone MONTHRPT job, and/or Archive Monthly History processing to the stand-alone HISTM job, thereby reducing MONTHLY job elapsed time and increasing database availability. You can use CA MICS AUTOSUBMIT options to automatically submit the MONTHRPT and/or HISTM jobs, or you can schedule monthly reporting and/or weekly archive processing externally.

o Execute the YEARLY process at the beginning of each new year (the YEARLY process is composed of the DAILY, YEARLY, and BACKUP operational jobs).

You can define the CA MICS processing schedule to your installation's production scheduling facility or you can use standard CA MICS facilities to schedule production work.

CA MICS provides an online facility, Operational Status and Tracking, that determines which operational process (DAILY, WEEKLY, MONTHLY or YEARLY) needs to be performed. You can use the facility to generate and submit a tailored job stream to accomplish the required operational process or you can use it to monitor performance of your installation's production scheduling facility.

If your installation does not have a production scheduling facility AND you do not want to use online Operational Status and Tracking for production process scheduling, then CA MICS provides a job (SCHEDULE) that also determines which operational process needs to be performed and builds a tailored job stream to accomplish the process. You can submit the SCHEDULE job manually or run SCHEDULE under the control of a production job scheduler.

4.1.4 Restart Facility

The need to recover from problems such as system crashes, DASD hardware errors, tape handling errors, or tape I/O errors leads to the requirement to be able to restart CA MICS operational jobs. CA MICS is step restartable, meaning that an operational job can be restarted at any step within it. In addition, many CA MICS Data Integration Applications support optional internal step restart.

If an error occurs during the processing of an operational CA MICS job, CA MICS forces a user abend to abort the execution. After the problem has been corrected, the CA MICS restart facility allows the job to be restarted at the step that failed. When activated, optional internal step restart facilities automatically resume step execution near the point of failure.

Step Restart

Standard step restart addresses the problem of losing completed work when a subsequent task fails. Each CA MICS operational job step is self-contained such that it is functionally independent of all other job steps. The only requirement is that job steps be executed in a pre-defined order, which is controlled through the CA MICS Checkpoint File (see Section 4.1.6). Once a job step successfully completes execution, the work performed in that step is protected from loss due to failures in a subsequent step.

Internal Step Restart

Internal step restart addresses the problem of losing work completed in a job step prior to the point of failure. Database update jobs steps are a series of linear processing tasks or "phases,"

- o Read raw input data, convert to SAS format, and output to intermediate work files.

- o Sort, analyze, summarize, and enhance intermediate work file contents to create a new DETAIL cycle.

- o Summarize DETAIL data to create new DAYS cycles and to update current week-to-date and month-to-date cycles.

- o Production cutover of new database cycles and existing cycle "aging."

With standard step restart, previously completed processing must be repeated as the step always restarts at the beginning. With internal step restart, reprocessing completed "phases" can be avoided.

CA MICS internal step restart utilizes a checkpoint/restart technique. When internal step restart is active, the database update job step "checkpoints" (or saves) intermediate results (work file contents) and the operational environmental at the end of each processing phase. Then, if required, the database update step can resume execution at the beginning of the processing phase in which the failure occurred. 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.

Internal step restart is implemented by product, with the number and location of restart points dependent upon product architecture and data volumes. In addition, internal step restart is activated at the product level according to your unique installation requirements. While internal step restart can dramatically reduce problem recovery cost, both in time and system resources, there is overhead associated with taking checkpoints and managing saved materials. Thus, you may find that, for some products and unit database environments, costs outweigh potential savings. For example, internal step restart may be of little value for a CICS unit supporting test regions, while internal step restart may prove very valuable in unit database updates for your production CICS regions.

Internal step restart uses OS/390 dynamic allocation services to create new data sets and to access existing data sets. Data set allocation parameters are specified by product in prefix.MICS.PARMS(cccOPS) and permanent changes to data set allocation parameters (e.g., to increase the space allocation for the WORK data set) require both changing the cccOPS parameter and executing the corresponding cccPGEN job. However, in restart situations, such as when recovering from a production job abend, you may temporarily override data set allocation parameters for one or more dynamically allocated data sets by using the //PARMOVRD facility (see Section 2.3.6).

See the individual product guides for instructions and considerations for activating internal step restart in a specific product, and see CA MICS Planning, Installation, Operation, and Maintenance Guide Section 2.3.6 for more information on execution-time override of dynamic data set allocation parameters.

4.1.5 Select and Force Facilities

To preserve the integrity of data in the CA MICS Database, a number of rigid input checks occur prior to data being written to the Database. At times, however, special processing requirements dictate the need to safely bypass these protection features.

The Select facility provides a means to select only a portion of the data on a given input file. A common use for the Select facility occurs when processing cumulative input files such as the SMF history files. For example, if a daily tape is damaged, the Select facility allows you to use a cumulative tape to access the day's data.

The Force facility overrides standard Database time-range checks, enabling the System Administrator to process data for date/time ranges that are already in the Database. For example, if a portion of data was unavailable for the standard daily update, perhaps because a remote site's data transmission link to the host site was down, the Force facility enables the System Administrator to include the remote site's "old" data in the Database.

4.1.6 Checkpoint File

The CA MICS Checkpoint File is a permanent data set containing information about jobs and data for an entire unit database. The Checkpoint File is frequently referred to as the unit checkpoint. It is concerned with ensuring database integrity across database update job steps and update cycles.

- The unit checkpoint should not be confused with database update restart checkpoints which are temporary, product specific files used to save an image of the operational environment for restart within a database update job step.

- The unit checkpoint is also separate and distinct from the optional incremental update checkpoint files which are permanent, product specific data sets used to ensure the integrity of incremental update processing for a single CA MICS data integration product.

The CA MICS Checkpoint file includes the termination status of each operational job and date ranges of data that has been used to update the CA MICS Database. Each unit database maintains its own Checkpoint File because operational jobs are specific to each unit.

The CA MICS Checkpoint File:

- enables each of the operational jobs to avoid out-of-order execution if a prior job or job step did not complete successfully

- contains information needed to conduct duplicate data protection checks

- contains control fields used to implement the Select and Force capabilities introduced earlier

4.1.7 Database Update Audit Trail

An update audit trail is automatically built for each unit database as part of normal processing output. It is written to the Database Input History Log File (ADMIHL) and is used to track operational activity.

The file contains records documenting input data that has been used to update the database for a given system (SYSID) and component (such as RMF) for a given update cycle. It includes the time range the data encompassed and the number of records that were processed.

The Input History Log File helps the System Administrator answer questions about what data is in the Database. Also consult it when considering the use of the Select and Force facilities.

You can browse the Input History Log via the online Operational Status and Tracking facility. You can also access and report the ADMIHL file with a MICF inquiry. See the distributed BASLA1, BASLA2, BASLA3, and BASLA4 inquiries for examples of accessing the ADMIHL file.

4.1.8 Operational Status and Tracking Facility

CA MICS provides online user access to the database, documentation, and special administrator facilities through the CA MICS Workstation Facility (MWF), which runs as an ISPF application. Part of MWF is the CA MICS Administrator Facility (MAF). The CA MICS system administrator uses MAF to support and control the CA MICS complex.

Operational Status and Tracking is a MAF application that simplifies operational job control for the CA MICS database. It gives the CA MICS administrator a single point of control for all unit databases in a given complex. The administrator can do the following:

- Schedule production database updates
- Monitor operational job processes
- Check database update status
- Review database update history
- Perform error condition recovery/restart
- Initiate database backup and restore
- Review the CA MICS database update audit trail
- Review the CA MICS checkpoint data set
- Update the CA MICS checkpoint force and select time ranges
- Control database updates during Product Support Program (PSP) maintenance application

4.1.9 Administrative Activities

CA MICS provides online user access to the database, documentation, and special administrator facilities through the CA MICS Workstation Facility (MWF), which runs as an ISPF application.

Part of MWF is the CA MICS Administrator Facility (MAF). The CA MICS system administrator uses MAF to support and control the CA MICS complex:

- Through MAF parameters, the system administrator controls the operation and appearance of MWF and the interactive CA MICS applications available under it. MAF parameters are specified once and apply to all MWF users. CA MICS users can override selected parameters as required. Once overridden, a user's parameter value is not affected by MAF options changes.

- Through MAF's Operational Status and Tracking dialog, the system administrator can monitor and control CA MICS production processing.

- Through MAF's Product Support Program (PSP) Online Services dialog, the system administrator can review available product changes, apply product changes, and review the status of CA MICS system maintenance.

- Through the Authorization Assignment dialog, the system administrator can review and update specifications that control access to CA MICS' ISPF applications. Authorization assignment allows the system administrator to limit access to each of the MWF applications and to limit the use of certain facilities within those applications. For example, access to MAF is controlled by the Authorization Assignment dialog.

- Through the CA MICS Administrator Utilities dialog, the system administrator can run reports that identify compression candidates and recommend CA MICS database compression specifications.

4.2 Operational Guidelines

Written for the System Administrator and production control personnel, this section lists the things that need to be done to operate CA MICS. It is organized like a cookbook, meaning that it lists the tasks that need to be performed without describing why they need to be done or how to do them. (If you need to know why something needs to be done, refer to the Overview section. If you need to know how to do something, refer to the Reference section.)

The topics presented in this section are:

1 - Getting Started
2 - Production Operations
3 - Monitoring CA MICS Operations
4 - Diagnosing Operational Problems
5 - System Restart and Recovery
6 - Entering Data Into CA MICS

4.2.1 Getting Started

When you first install CA MICS, you should execute the system manually for a period of time before implementing full production operation. This "break-in" period may vary from a few days to a month or two depending on your installation's unique requirements and production batch facilities.

- You should use the "break-in" period to familiarize yourself with CA MICS production operation and especially problem diagnosis and restart/recovery facilities.

- You should also defer exploiting incremental update facilities until you gain experience with standard CA MICS database update processing. Incremental update processing introduces additional operational complexity and scheduling considerations that are better avoided when first integrating CA MICS into your daily production operations.

The steps described in this section provide a good introduction for the new CA MICS administrator. Remember to limit your "Getting Started" activities to a TEST unit database. These procedures are NOT appropriate for a production database containing valid information.

Getting started guidelines assume CA MICS is installed and ready to run. (See Chapters 2 and 3 for installation parameters and instructions.) Though there are many operational functions, these guidelines help you become familiar with the system before moving it into production mode.

INITIAL CA MICS OPERATIONAL PROCESSING

Use the CA MICS Operational Status and Tracking facility of MWF to schedule, submit, and monitor CA MICS operational processing.

- Each day, when input data is available, use the SCHEDULE command to initiate CA MICS processing.

- Review the Operational Status and Tracking display to monitor the status of today's processing.

- Resolve any problem conditions and restart if necessary.

- After processing completes successfully, review Database status (STATUS command), Database update audit trail

(HISTORY command), CA MICS checkpoint contents (CHECKPT command), and printed outputs (JCL log, MICSLOG, and SAS log).

CA MICS DATABASE UPDATE TEST

WARNING: The following should be performed ONLY on a TEST unit database. DO NOT PERFORM THIS PROCEDURE ON A PRODUCTION CA MICS UNIT DATABASE.

Use CA MICS Operational Status and Tracking to perform DAILY, WEEKLY, and MONTHLY processing.

o DAILY Process

  - When input data is available, use the DAILY command to initiate CA MICS processing.

  - Review the Operational Status and Tracking display to monitor the status of today's processing.

  - Resolve any problem conditions and restart if necessary.

  - After processing completes successfully, review Database status (STATUS command), Database update audit trail (HISTORY command), CA MICS checkpoint contents, and printed outputs (JCL log, MICSLOG, and SAS log).

o WEEKLY Process

  - When a new day's input data is available, use the WEEKLY command to initiate CA MICS processing.

  - Review the Operational Status and Tracking display to monitor the status of today's processing.

  - Resolve any problem conditions and restart if necessary.

  - If you activated the optional Archive Audit and/or Archive Weekly History facilities and chose the option to offload this processing to the stand-alone AUDIT and/or HISTW jobs, you will need to ensure that these jobs are submitted and execute successfully. If you enabled the AUTOSUBMIT option for the selected archive processing, then the WEEKLY job WEEK300 step will have automatically submitted the AUDIT and/or HISTW job for execution.

  - After processing completes successfully, review Database status (STATUS command), Database update audit trail (HISTORY command), CA MICS checkpoint contents, and printed outputs (JCL log, MICSLOG, and SAS log).

- MONTHLY Process

  - When a new day's input data is available, use the MONTHLY command to initiate CA MICS processing.

  - Review the Operational Status and Tracking display to monitor the status of today's processing.

  - Resolve any problem conditions and restart if necessary.

  - If you activated the optional Archive Monthly History facility and chose the option to offload this processing to the stand-alone HISTM job, you will need to ensure that this job is submitted and executes successfully. If you enabled the ARCHIVE HISTM JOB AUTOSUBMIT option, then the MONTHLY job MONTH300 step will have automatically submitted the HISTM job for execution.

  - After processing completes successfully, review Database status (STATUS command), Database update audit trail (HISTORY command), CA MICS checkpoint contents, and printed outputs (JCL log, MICSLOG, and SAS log).

- RESTORE Job

  - Restore the Database from the backup tape created by the WEEKLY process.

  - After processing completes successfully, review Database status (STATUS command), Database update audit trail (HISTORY command), CA MICS checkpoint contents, and printed outputs (JCL log, MICSLOG, and SAS log).

  - Database status and contents should be the same as it was at the completion of the WEEKLY processing.

  - Restore the Database from the monthly backup (MBACK) tape created by the MONTHLY process.

- After processing completes successfully, review Database status (STATUS command), Database update audit trail (HISTORY command), CA MICS checkpoint contents, and printed outputs (JCL log, MICSLOG, and SAS log).

- Database status and contents should be the same as it was at the completion of the MONTHLY processing.

If the unit database contains the Batch and Operations Analyzer, you may optionally update the Operations Information Area IPL Outage Times and Reason Codes File. See the Data Entry section (4.2.6) for more information on manual update of Operations Information Area files.

CA MICS PROCESSING ABEND, RESTART, and FORCE/SELECT

WARNING: The following should be performed ONLY on a TEST unit database. DO NOT PERFORM THIS PROCEDURE ON A PRODUCTION CA MICS UNIT DATABASE.

This procedure uses two files of input data.

o TAPE 1: Input data for one day (or part of a day).

o TAPE 2: Input data for a day (or part of a day) after the TAPE 1 data. Date/times on TAPE 2 must be greater than date/times on TAPE 1.

Use the CA MICS Operational Status and Tracking to perform DAILY processing for the TAPE 2 input data.

o Use the DAILY EDIT command to submit the CA MICS DAILY job with TAPE 2 as input.

o After processing completes successfully, review Database status (STATUS command), Database update audit trail (HISTORY command), CA MICS checkpoint contents, and printed outputs (JCL log, MICSLOG, and SAS log).

Next, use CA MICS Operational Status and Tracking to perform DAILY processing for the TAPE 1 input data.

o Use the DAILY EDIT command to submit the CA MICS DAILY job with TAPE 1 as input.

o CA MICS processing should fail with a U300 abend because the data from TAPE 1 is older than (date/time is less than) data already input to the Database.

- Use the FORCE command to select and force the TAPE 1 data into the Database.

- Use the RESTART command to restart processing for TAPE 1.

- After processing completes successfully, review Database status (STATUS command), Database update audit trail (HISTORY command), CA MICS checkpoint contents, and printed outputs (JCL log, MICSLOG, and SAS log).

4.2.2 Production Operations

There are a number of ways to operate CA MICS in a production mode. If use of a production scheduler, either proprietary or internally developed, is widely accepted at your installation, you should place the CA MICS operational jobs under its control.

If a production scheduler is not available, you can use either Operational Status and Tracking or you can operate CA MICS in a manual mode.

Production processing techniques are discussed in the following sections:

1 - Production Job Scheduler
2 - Operational Status and Tracking Facility
3 - Manual Operation
4 - Parallel Database Update Processing

4.2.2.1 Production Job Scheduler

Using a production job scheduler (a vendor package or timed job submission) with CA MICS enables you to automatically execute the Database update processing according to a pre-established schedule. The advantage to using a production job scheduler is that it adheres to your installation standards and is familiar to your staff.

If you have activated incremental update for one or more products in the unit database, you will probably use a production job scheduler to schedule the SPLITSMF job (if used) and the INCRccc job for each product using incremental update. SPLITSMF and INCRccc job scheduling is dependent upon the availability of the input data and on day-time batch processing considerations unique to your installation.

For the CA MICS end-of-day processing (i.e., DAILY, WEEKLY, MONTHLY, and BACKUP operational jobs), you can use a production job scheduler to either schedule each operational job individually, or you can use the CA MICS SCHEDULE job. The approach you choose depends on your installation's production processing standards and procedures.

o If you use the CA MICS SCHEDULE job, then only one item is defined to the production scheduler for the CA MICS end-of-day processing. The SCHEDULE job determines what needs to be done, builds a job stream to do it, and submits the job stream to the internal reader.

o If you use the CA MICS SCHEDULE job, then restart and recovery processes within CA MICS would most likely be controlled outside of the production scheduler.

o Some organizations restrict the use of the internal reader or may have standards requiring that the production scheduling facility control ALL production processing. In this situation, you must implement the SCHEDULE algorithm in the production scheduler definitions.

In either case, you can still monitor CA MICS processing with the Operational Status and Tracking facility.

Consult your installation's operations or production control group to find out what they will require to manage CA MICS production processing.

The remainder of this section provides more information on implementing CA MICS end-of-day production operation with your installation's production batch scheduling facilities.

USING THE CA MICS SCHEDULE JOB

Define the CA MICS SCHEDULE job to your production batch scheduling system.

o Submit the SCHEDULE job each day as soon as practical after CA MICS input data is available.

   NOTE: If you activated Incremental Update for one or more products in this unit database, then the SCHEDULE job should NOT be submitted until after completing all INCRccc processing for the day.

o SCHEDULE will automatically submit DAILY, WEEKLY, MONTHLY, YEARLY, and BACKUP processing as required.

If you specified AUTOSUBMIT YES in prefix.MICS.PARMS(JCLDEF), CA MICS will automatically submit the DAILYRPT, WEEKRPT, and MONTHRPT jobs using the JES internal reader. (These RPT jobs execute MICF inquiry production reporting and produce MBO and exception reports for the products in the unit.)

If you specified AUTOSUBMIT NO in prefix.MICS.PARMS(JCLDEF), you should reconsider your decision to use the CA MICS SCHEDULE job because CA MICS will NOT automatically submit DAILYRPT, WEEKRPT, and MONTHRPT. Therefore, MICF inquiry production processing will not execute. (The MBO and exception reports are produced by the DAILY, WEEKLY, and MONTHLY jobs, so those reports will be available.)

If you specified the YES, JOB, and AUTOSUBMIT options for the ARCHIVE AUDIT, HISTW, and/or HISTM parameters in prefix.MICS.PARMS(JCLDEF), CA MICS will automatically submit the AUDIT, HISTW, and/or HISTM jobs using the JES internal reader. (These jobs offload archive tape processing from the WEEKLY and/or MONTHLY operational jobs to increase database availability. This processing can alternatively be executed within the WEEKLY and/or MONTHLY operational jobs.)

If you specified the YES and JOB options for the ARCHIVE AUDIT, HISTW, and/or HISTM parameters in prefix.MICS.PARMS(JCLDEF) BUT did NOT include the AUTOSUBMIT option on these parameter statements, you should reconsider your decision to use the CA MICS SCHEDULE job because CA MICS will NOT automatically submit AUDIT, HISTW, or HISTM.

DETAIL CA MICS JOB SCHEDULING

Define the CA MICS processing schedule to your production batch scheduling system for each unit database. Primary considerations/tasks are noted below. The CA MICS processing schedule is documented in the Reference section.

If you specified AUTOSUBMIT NO in prefix.MICS.PARMS(JCLDEF):

- Schedule the DAILY job each day as soon as practical after CA MICS input data is available.

  NOTE: If you activated Incremental Update for one or more products in this unit database, then the SCHEDULE job should NOT be submitted until after completing all INCRccc processing for the day.

- Schedule DAILYRPT after DAILY completes successfully.

- Schedule WEEKLY after DAILY completes successfully at the beginning of a new week.

- If you specified

  ```
  ARCHIVE AUDIT YES JOB
  ```

  in prefix.MICS.PARMS(JCLDEF) without including the AUTOSUBMIT option, then schedule AUDIT after WEEKLY completes successfully.

  If you included the AUTOSUBMIT option, the WEEK300 step automatically submits AUDIT using the JES internal reader.

  If you did not specify the JOB option, then Archive Audit processing is performed in the WEEK300 step within the WEEKLY operational job.

- If you specified

  ```
  ARCHIVE HISTW YES JOB
  ```

  in prefix.MICS.PARMS(JCLDEF) without including the AUTOSUBMIT option, then schedule HISTW after WEEKLY completes successfully.

  If you included the AUTOSUBMIT option, the WEEK300 step automatically submits HISTW using the JES internal reader.

  If you did not specify the JOB option, then Archive Weekly

History processing is performed in the WEEK300 step within the WEEKLY operational job.

- Schedule WEEKRPT after WEEKLY completes successfully.

- Schedule MONTHLY after DAILY completes successfully at the beginning of a new month.

- If you specified

  ```
  ARCHIVE HISTM YES JOB
  ```

  in prefix.MICS.PARMS(JCLDEF) without including the AUTOSUBMIT option, then schedule HISTM after MONTHLY completes successfully.

  If you included the AUTOSUBMIT option, the MONTH300 step automatically submits HISTM using the JES internal reader.

  If you did not specify the JOB option, then Archive Monthly History processing is performed in the MONTH300 step within the MONTHLY operational job.

- Schedule MONTHRPT after MONTHLY completes successfully.

- Schedule YEARLY after MONTHLY completes successfully at the beginning of a new year.

- Schedule BACKUP processing as required. Backups should occur after DAILY, WEEKLY, MONTHLY, and YEARLY processing completes for the day.

If you specified AUTOSUBMIT YES in prefix.MICS.PARMS(JCLDEF), CA MICS will automatically submit the DAILYRPT, WEEKRPT, and MONTHRPT jobs using the JES internal reader. You must still define DAILY, WEEKLY, MONTHLY, YEARLY, and BACKUP to your production scheduler.

- Schedule the DAILY job each day as soon as practical after CA MICS input data is available.

  NOTE: If you activated Incremental Update for one or more products in this unit database, then the SCHEDULE job should NOT be submitted until after completing all INCRccc processing for the day.

- Schedule WEEKLY after DAILY completes successfully at the beginning of a new week.

```
   NOTE:  Archive Audit and Archive Weekly History processing
          considerations (the AUDIT and HISTW jobs) are the
          same as described above (under AUTOSUBMIT NO).

o  Schedule MONTHLY after DAILY completes successfully at
   the beginning of a new month.

   NOTE:  Archive Monthly History processing considerations
          (the HISTM job) is the same as described above
          (under AUTOSUBMIT NO).

o  Schedule YEARLY after MONTHLY completes successfully at
   the beginning of a new year.

o  Schedule BACKUP processing as required.  Backups should
   occur after DAILY, WEEKLY, MONTHLY, and YEARLY processing
   completes for the day.
```

4.2.2.2 Operational Status and Tracking Facility

Operational Status and Tracking is an online tool for initiating and monitoring CA MICS database operational processing. It is available to authorized users from the CA MICS Administrator Facility (MAF) option of the CA MICS Workstation Facility (MWF).

If you have activated incremental update for one or more products in the unit database, you will probably use a production job scheduler or a manual procedure to schedule the optional SPLITSMF job and the INCRccc jobs (one for each product using incremental update). SPLITSMF and INCRccc job scheduling is dependent upon the availability of the input data and on day-time batch processing considerations unique to your data center. Operational Status and Tracking does not currently provide facilities for scheduling either the SPLITSMF job or the INCRccc jobs, however they can still be manually submitted and tracked using Operational Status and Tracking.

For the CA MICS end-of-day processing, you can use the SCHEDULE command to automatically submit DAILY, WEEKLY, MONTHLY, or YEARLY processing according to the CA MICS processing schedule. Alternatively, you can use Operational Status and Tracking to "manually" submit DAILY, WEEKLY, MONTHLY, or YEARLY processing. Operational Status and Tracking will still prompt you when WEEKLY, MONTHLY, or YEARLY processing is due.

The remainder of this section provides more information on implementing CA MICS end-of-day production operation with Operational Status and Tracking. See the reference section and the online tutorials for detailed documentation on Operational Status and Tracking facilities, panel data entry instructions, and commands.

USING THE SCHEDULE COMMAND

Specify the correct overdue time-of-day for each CA MICS unit database on the Operational Status and Tracking display.

Authorize production control personnel to access CA MICS Operational Status and Tracking.

Authorize production control personnel to update the prefix.MICS.RESTART.CNTL data set.

Train production control personnel to use Operational Status and Tracking to control and monitor production CA MICS processing. The information in section 4.2.1 is a good starting point for training people to use the facility.

Each Day:

o Enter the SCHEDULE command for each CA MICS unit database as soon as practical after CA MICS input data is available.

   Note: If you activated Incremental Update for one or more products in this unit database, do NOT enter the SCHEDULE command until after completing all INCRccc processing for the day.

o Use the Operational Status and Tracking display to monitor CA MICS production processing. Initiate problem determination, correction, and restart if necessary.

If you specified AUTOSUBMIT YES in prefix.MICS.PARMS(JCLDEF), CA MICS will automatically submit the DAILYRPT, WEEKRPT, and MONTHRPT jobs using the JES internal reader.

If you specified AUTOSUBMIT NO in prefix.MICS.PARMS(JCLDEF), you should reconsider your decision to use Operational Status and Tracking to schedule production CA MICS processing because CA MICS will NOT automatically submit DAILYRPT, WEEKRPT, and MONTHRPT to perform daily/weekly/monthly MICF inquiry production processing. You must manually submit the DAILYRPT, WEEKRPT, and MONTHRPT jobs from prefix.MICS.CNTL.

o Submit DAILYRPT after the Operations Status and Tracking display shows DAILY processing completed.

o Submit DAILYRPT and WEEKRPT after the Operations Status and Tracking display shows WEEKLY processing completed.

o Submit DAILYRPT and MONTHRPT after the Operations Status and Tracking display shows MONTHLY processing completed.

o Submit DAILYRPT after the Operations Status and Tracking display shows YEARLY processing completed.

MANUAL SCHEDULING

Specify the correct overdue time-of-day for each CA MICS unit database on the Operational Status and Tracking display.

Authorize production control personnel to access CA MICS Operational Status and Tracking.

Authorize production control personnel to update the prefix.MICS.RESTART.CNTL data set.

Train production control personnel to use Operational Status and Tracking to control and monitor production CA MICS processing. The information in section 4.2.1 is a good starting point for training people to use the facility.

Each Day:

- Enter the DAILY, WEEKLY, MONTHLY, or YEARLY command as noted in the Current Operation column for each CA MICS unit database as soon as practical after CA MICS input data is available.

  Note: If you activated Incremental Update for one or more products in this unit database, then do NOT enter the DAILY, WEEKLY, MONTHLY, or YEARLY command until after completing all INCRccc processing for the day.

  - The current operation should have a status of DUE TODAY or OVERDUE.

  - The WEEKLY, MONTHLY, and YEARLY commands execute the DAILY job in addition to the WEEKLY, MONTHLY, or YEARLY job.

  - The DAILY, WEEKLY, and MONTHLY commands include BACKUP processing. If you want to defer BACKUP processing, enter the NOBACKUP operand on the command line (that is, DAILY p NOBACKUP where p is the unit database ID).

  - If WEEKLY, MONTHLY, or YEARLY processing is due but you wish to defer this processing to the next day, enter the DAILY command. Tomorrow, Operational Status and Tracking will once again request WEEKLY, MONTHLY, or YEARLY processing.

- Use the Operational Status and Tracking display to monitor CA MICS production processing. Initiate problem determination, correction, and restart if necessary.

If you specified AUTOSUBMIT YES in prefix.MICS.PARMS(JCLDEF), CA MICS will automatically submit the DAILYRPT, WEEKRPT, and MONTHRPT jobs using the JES internal reader.

If you specified AUTOSUBMIT NO in prefix.MICS.PARMS(JCLDEF), you should reconsider your decision to use Operational Status and Tracking for production CA MICS processing because CA MICS will NOT automatically submit DAILYRPT, WEEKRPT, and MONTHRPT to perform daily/weekly/monthly MICF inquiry production processing. You must manually submit, from prefix.MICS.CNTL, the DAILYRPT, WEEKRPT, and MONTHRPT jobs.

- Submit DAILYRPT after the Operations Status and Tracking display shows DAILY processing completed.

- Submit DAILYRPT and WEEKRPT after the Operations Status and Tracking display shows WEEKLY processing completed.

- Submit DAILYRPT and MONTHRPT after the Operations Status and Tracking display shows MONTHLY processing completed.

- Submit DAILYRPT after the Operations Status and Tracking display shows YEARLY processing completed.

If you specified the YES, JOB, and AUTOSUBMIT options for the ARCHIVE AUDIT, HISTW, and/or HISTM parameters in prefix.MICS.PARMS(JCLDEF), CA MICS will automatically submit the AUDIT, HISTW, and/or HISTM jobs using the JES internal reader. (These jobs offload archive tape processing from the WEEKLY and/or MONTHLY operational jobs to increase database availability. This processing can alternatively be executed within the WEEKLY and/or MONTHLY operational jobs.)

If you specified the YES and JOB options for the ARCHIVE AUDIT, HISTW, and/or HISTM parameters in prefix.MICS.PARMS(JCLDEF) BUT did NOT include the AUTOSUBMIT option on these parameter statements, you should reconsider your decision to use the CA MICS SCHEDULE job because CA MICS will NOT automatically submit AUDIT, HISTW, or HISTM to perform weekly and monthly archive tape processing. You must manually submit the AUDIT, HISTW, and/or HISTM jobs.

- Submit AUDIT and/or HISTW (if activated) after the Operations Status and Tracking display shows WEEKLY processing completed.

- Submit HISTM (if activated) after the Operations Status and Tracking display shows MONTHLY processing completed.

MONITORING CA MICS PRODUCTION PROCESSING

The primary Operational Status and Tracking display lists the current status for each unit database. Overdue processing and database update failures are noted.

- The STATUS command provides additional details relating to unit database status and error conditions.

- The HISTORY command displays the database update audit trail (ADMIHL file).

- The CHECKPT command displays the checkpoint database update time range records.

- The JOBLOG command displays a processing audit log.

- You can override the default overdue time-of-day (8:00 a.m.) for each CA MICS unit database on the Operational Status and Tracking display by scrolling RIGHT MAX and changing the 8 a.m. default deadline.

4.2.2.3 Manual Operation

If you choose not to use a production batch scheduling facility or the CA MICS Operational Status and Tracking application for production CA MICS processing, then you can manually submit the CA MICS production jobs as required for each unit database. All CA MICS production jobs mentioned in this section are submitted from the prefix.MICS.CNTL data set.

If you have activated incremental update for one or more products in the unit database, you will probably use a manual procedure to schedule the optional SPLITSMF job and the INCRccc jobs (one for each product using incremental update). SPLITSMF and INCRccc job scheduling is dependent upon the availability of the input data and on day-time batch processing considerations unique to your installation.

For the CA MICS end-of-day processing, you can use the CA MICS SCHEDULE job or you can submit each operational job individually. The approach you choose depends on your installation's production processing standards and procedures.

o If you use the CA MICS SCHEDULE job, then only one job is submitted each day for the CA MICS end-of-day processing. The SCHEDULE job determines what needs to be done, builds a job stream to do it, and submits the job stream to the internal reader.

o Some organizations restrict the use of the internal reader or may have standards requiring that the production control group submit each production job. In this situation, you must implement the SCHEDULE algorithm manually.

In either case, you can still monitor CA MICS processing with the Operational Status and Tracking facility.

Consult your installation's operations or production control group to find out what they will require to manage CA MICS production processing.

The remainder of this section provides more information on implementing CA MICS end-of-day production operation with manual job submission.

USING THE CA MICS SCHEDULE JOB

Submit the SCHEDULE job each day for each unit database as soon as practical after CA MICS input data is available. SCHEDULE will automatically submit a single batch job stream for DAILY, WEEKLY, MONTHLY, YEARLY, and BACKUP processing as required.

NOTE: If you activated Incremental Update for one or more products in this unit database, then the SCHEDULE job should NOT be submitted until after completing all INCRccc processing for the day.

If you specified AUTOSUBMIT YES in prefix.MICS.PARMS(JCLDEF), CA MICS will automatically submit the DAILYRPT, WEEKRPT, and MONTHRPT jobs using the JES internal reader.

If you specified AUTOSUBMIT NO in prefix.MICS.PARMS(JCLDEF), you should reconsider your decision to use the CA MICS SCHEDULE job because CA MICS will NOT automatically submit DAILYRPT, WEEKRPT, and MONTHRPT to perform daily/weekly/monthly MICF inquiry production processing.

If you specified the YES, JOB, and AUTOSUBMIT options for the ARCHIVE AUDIT, HISTW, and/or HISTM parameters in prefix.MICS.PARMS(JCLDEF), CA MICS will automatically submit the AUDIT, HISTW, and/or HISTM jobs using the JES internal reader. (These jobs offload archive tape processing from the WEEKLY and/or MONTHLY operational jobs to increase database availability. This processing can alternatively be executed within the WEEKLY and/or MONTHLY operational jobs.)

If you specified the YES and JOB options for the ARCHIVE AUDIT, HISTW, and/or HISTM parameters in prefix.MICS.PARMS(JCLDEF) BUT did NOT include the AUTOSUBMIT option on these parameter statements, you should reconsider your decision to use the CA MICS SCHEDULE job because CA MICS will NOT automatically submit AUDIT, HISTW, or HISTM.

DETAIL CA MICS JOB SUBMISSION

Primary considerations/tasks for implementing the CA MICS scheduling algorithms manually are noted below. The CA MICS processing schedule is documented in the Reference section.

If you specified AUTOSUBMIT NO in prefix.MICS.PARMS(JCLDEF):

o Submit the DAILY job each day as soon as practical after

CA MICS input data is available.

NOTE: If you activated Incremental Update for one or more products in this unit database, then the DAILY job should NOT be submitted until after completing all INCRccc processing for the day.

- Submit DAILYRPT after DAILY completes successfully.
- Submit WEEKLY after DAILY completes successfully at the beginning of a new week.
- If you specified

  ```
  ARCHIVE AUDIT YES JOB
  ```

  in prefix.MICS.PARMS(JCLDEF) without including the AUTOSUBMIT option, then schedule AUDIT after WEEKLY completes successfully.

  If you included the AUTOSUBMIT option, the WEEK300 step automatically submits AUDIT using the JES internal reader.

  If you did not specify the JOB option, then Archive Audit processing is performed in the WEEK300 step within the WEEKLY operational job.
- If you specified

  ```
  ARCHIVE HISTW YES JOB
  ```

  in prefix.MICS.PARMS(JCLDEF) without including the AUTOSUBMIT option, then schedule HISTW after WEEKLY completes successfully.

  If you included the AUTOSUBMIT option, the WEEK300 step automatically submits HISTW using the JES internal reader.

  If you did not specify the JOB option, then Archive Weekly History processing is performed in the WEEK300 step within the WEEKLY operational job.
- Submit WEEKRPT after WEEKLY completes successfully.
- Submit MONTHLY after DAILY completes successfully at the beginning of a new month.
- If you specified

```
          ARCHIVE HISTM YES JOB

     in prefix.MICS.PARMS(JCLDEF) without including the
     AUTOSUBMIT option, then schedule HISTM after MONTHLY
     completes successfully.

     If you included the AUTOSUBMIT option, the MONTH300 step
     automatically submits HISTM using the JES internal reader.

     If you did not specify the JOB option, then Archive
     Monthly History processing is performed in the MONTH300
     step within the MONTHLY operational job.

  o  Submit MONTHRPT after MONTHLY completes successfully.

  o  Submit YEARLY after MONTHLY completes successfully at the
     beginning of a new year.

  o  Submit BACKUP processing as required.  Backups should
     occur after DAILY, WEEKLY, MONTHLY, and YEARLY processing
     completes for the day.

  If you specified AUTOSUBMIT YES in prefix.MICS.PARMS(JCLDEF),
  CA MICS will automatically submit the DAILYRPT, WEEKRPT, and
  MONTHRPT jobs using the JES internal reader.  You must still
  submit the DAILY, WEEKLY, MONTHLY, YEARLY, and BACKUP jobs as
  required.

  o  Submit the DAILY job each day as soon as practical after
     CA MICS input data is available.

  o  Submit WEEKLY after DAILY completes successfully at the
     beginning of a new week.

     NOTE:  Archive Audit and Archive Weekly History processing
            considerations (the AUDIT and HISTW jobs) are the
            same as described above (under AUTOSUBMIT NO).

  o  Submit MONTHLY after DAILY completes successfully at the
     beginning of a new month.

     NOTE:  Archive Monthly History processing considerations
            (the HISTM job) is the same as described above
            (under AUTOSUBMIT NO).

  o  Submit YEARLY after MONTHLY completes successfully at the
     beginning of a new year.

  o  Submit BACKUP processing as required.  Backups should
```

occur after DAILY, WEEKLY, MONTHLY, and YEARLY processing completes for the day.

4.2.2.4 Parallel Database Update Processing

CA MICS DAILY job execution for a unit database is a serial process. Database files are updated in a fixed processing order by product. Each product must complete daily database update processing before processing for the next product can begin. The serial nature of the daily database update results from ensuring database integrity and requiring exclusive control of the database before making any modifications.

If you have installed multiple CA MICS products and/or process large daily workloads, it is possible that a single, serial database update for all products is impractical. This is one reason you might implement multiple unit databases, spreading this workload over multiple, parallel daily database update processes. Remember, the database update is ONLY serial WITHIN a unit. Each unit database is independent of all other units and thus you can execute database update jobs for multiple units at the same time.

CA MICS gives you powerful options to address your installation's unique processing requirements. The ability to install multiple unit databases is one example of CA MICS' flexibility. You may choose to implement multiple unit databases for a number of reasons, for example:

- The input data for different products may become available at different times of the day. So rather than waiting for all input data to be available before starting the daily database update, you split-out one or more products to one or more additional unit databases, thereby giving you the flexibility to begin the daily database update for some products earlier than might have been possible when all products were in a single unit. Getting started on daily update processing earlier, combined with the ability to run database updates for multiple units concurrently, lets you process more input data in less time.

- The input data for different systems processed by a single CA MICS product may become available at different times of the day. For example, the measurement log for some CICS regions may be released for CA MICS processing at a different time than the measurement log for other CICS regions. Or, you may be processing SMF data from a remote site, and this SMF data may arrive sometime after the rest of your SMF data becomes available. So, once again, rather than waiting for all input data to be available before starting the daily database update, you split-out processing for one or more systems (e.g., CICS regions or SMF data from remote sites) to one or more additional unit databases, thereby giving you the flexibility to begin the daily database update for some systems earlier than might have been possible when all systems were in a single unit. And, the database updates for these unique unit databases can run concurrently, letting you do more in less time.

Parallel database update processing is a common thread in these two examples, and multiple unit databases is the best means of achieving such parallelism. However, CA MICS incremental update facilities can also help you complete your nightly database update earlier than may be possible with standard, serial database update processing.

You should normally look to incremental update when you want to spread database update processing for one or more products over multiple, smaller incremental updates throughout the day. Standard incremental update implementation hinges on the availability of logical subsets of the total day's input data at times of the day when executing a CA MICS incremental update does not adversely impact your installation's online and batch workloads.

You can also use incremental update to achieve a significant degree of parallelism in database update processing when it is desirable to keep multiple products in a single unit database. You should note that incremental update introduces additional processing costs, operational complexity, and scheduling considerations, and incremental update provides less overall benefit than you would receive from spreading products over multiple units. However, incremental update can still be a valuable tool in configuring CA MICS to meet your unique requirements.

The remainder of this section describes daily database update processing in parallel with incremental update.

Setup and Preparation

The first step is to enable incremental update for one or more products in the unit database. Refer to the individual Product Guides for details on activating incremental update using the product's cccOPS parameter member. Please follow all instructions carefully, and run the specified generation and setup jobs. Each product that supports incremental update provides, in the Product Guide, a checklist for activating incremental update.

Examine the generated INCRccc jobs and prepare them for production execution. In particular, verify that the INPUTccc and/or INPUTSMF DDs point to the data set(s) that will contain input data for this product in this unit.

- If you specified the INCRSPLIT USE option in cccOPS, then the input data set for one or more INCRccc jobs will be created by the common SPLITSMF job. In this mode, the INCRccc job dynamically allocates the input data set (created by the SPLITSMF job), and there is no INPUTSMF DD statement in the INCRccc job.

- If you are using the SPLITSMF job, then verify that the INPUTSMF DD statement in the SPLITSMF job points to the data set that will contain the input SMF data for this unit.

Examine the re-generated DAILY job and prepare it for production execution. You will be revising the DAYnnn steps for the products for which you activated incremental update so that these job steps will process no input data. Remember to save your changes in a separate library so they are not lost when CA MICS operational jobs are regenerated.

- Add the SYSPARM=NODATA parameter to the EXEC statement for the DAYSMF step (if present) and the DAYnnn step for each product for which you activated incremental update.

- Modify the INPUTccc and/or INPUTSMF DD statements to specify DD DUMMY (no input data) in DAYnnn steps for the products for which you enabled incremental update. If this unit contains a DAYSMF step and incremental update is active for ALL products in the unit, specify DD DUMMY (no input data) on the INPUTSMF DD statement in the DAYSMF step.

Daily Operation

Each day, submit the INCRccc job, either manually or through your installation's production batch job scheduling facility, as soon as practical after input data is available for this product. Individual product

INCRccc jobs can execute in concurrently.

If you specified INCRSPLIT USE in cccOPS for one or more products, then submit the SPLITSMF job as soon as practical after SMF input data is available for this unit. Then, after SPLITSMF completes, submit the INCRccc jobs for each product for which you specified the INCRSPLIT USE option.

After all INCRccc jobs for this unit complete execution, submit the normal CA MICS daily database update processing using one of the techniques described in the prior sections. For example, you may choose to submit the CA MICS SCHEDULE job, either manually or through your installation's production batch job scheduling facility, or you might use the CA MICS Operational Status and Tracking Facility SCHEDULE command.

4.2.3 Monitoring CA MICS Operations

CA MICS is a production information system. You should monitor daily operations to ensure that the CA MICS Database is updated correctly and on time.

Use Operational Status and Tracking to monitor daily CA MICS operations. The primary Operational Status and Tracking display lists the current status for each unit database. Overdue processing and database update failures are noted.

- o The STATUS command provides additional details relating to unit database status and error conditions.

- o The HISTORY command displays the Database update audit trail (ADMIHL file).

- o The JOBLOG command displays a processing audit log.

- o You can override the default Overdue time-of-day (8:00 a.m.) for each CA MICS unit database on the Operational Status and Tracking display by scrolling RIGHT MAX and changing the 8 a.m. default deadline.

Review printed outputs from production CA MICS processing. Printed outputs include JCL logs, MICSLOG, and SAS log.

- o The Run Status Report describes the result of current processing. This information is also available with the Operational Status and Tracking STATUS command.

o The MICSLOG reports the status of each production job step. Database update steps report the number of input records processed and log the records deleted due to date-range, record length, and other validation tests. This information is also available with the Operational Status and Tracking HISTORY command.

  - Verify that the number of input records processed is reasonable for your normal processing. For example, a low value may indicate missing input data, while a high value might indicate duplicate input or new workloads.

  - Investigate the cause for deleted (or dropped) input records. For example, records deleted due to checkpoint date may indicate reprocessing of old input (duplicate) data.

Retain printed outputs from production CA MICS processing. As a guideline, consider keeping printed outputs until the corresponding input data or weekly/monthly/yearly cycle ages off of the Database.

4.2.4 Diagnosing Operational Problems

Use Operational Status and Tracking to analyze problems in DAILY, WEEKLY, MONTHLY, or YEARLY CA MICS database update processing. The primary Operational Status and Tracking display lists the current status for each unit database. Overdue processing and Database update failures are noted.

This section presents guidelines and samples for CA MICS operational problem diagnosis. Specific steps for CA MICS operational job restart are covered in the section 4.2.5 and the Reference section.

Note: This section does not directly address CA MICS incremental update processing. While the analysis concepts are the same, the actual techniques may differ. For example CA MICS Operational Status and Tracking does not support incremental update and you cannot use the STATUS command to research problems in INCRccc job execution. You can, however, get the same information from the INCRccc job MICSLOG output and the incremental update checkpoint data set.

DIAGNOSING OVERDUE PROCESSES

Operational Status and Tracking shows that a CA MICS production process (e.g., DAILY) is OVERDUE.

OVERDUE means that the processing due today has not yet been submitted via Operational Status and Tracking or has not yet executed.

Determine why the process has not run.

- o Has the CA MICS process been submitted for processing? If not, check your scheduling procedures to determine why scheduled processing has not begun.

  - If input data is not yet available, determine the cause and reschedule daily processing.

  - If input data is available, submit scheduled processing.

- o Is scheduled processing still in the JES input queue? If so, determine why the job has not yet executed and take corrective action.

- o Was scheduled processing submitted but not executed? If so, examine batch job outputs, especially the JES log and messages data set.

  - Did the operator cancel scheduled processing before execution? If so, determine why and resubmit scheduled processing.

  - Did scheduled processing fail with a JCL error prior to executing the first batch job step? If so, correct the problem and resubmit scheduled processing.

  - Did a system crash or JES failure purge scheduled processing from the system prior to executing the first batch job step? If so, resubmit scheduled processing.

DIAGNOSING INCOMPLETE PROCESSES

Operational Status and Tracking shows that a CA MICS production process (e.g., DAILY) is INCOMPLETE.

INCOMPLETE means that today's processing began execution. The operational job may still be executing.

Determine why the process is incomplete.

- o Is the operational job still executing? If so, allow it to complete.

- o If the operational job is not executing, the batch job may have been flushed by a system failure (e.g., an IPL) or may have failed with a JCL error so that the RSR (run status) step did not execute.

  - In this situation, Operational Status and Tracking will not report the correct job status and there is no Run Status Report for problem diagnosis.

    Enter the RSTATUS command. RSTATUS will update Operational Status and Tracking with the current status of CA MICS processing.

- o If, after running RSTATUS, CA MICS processing status is OVERDUE or DUE TODAY, resubmit scheduled processing. If status is FAILED, then refer to the discussion below.

DIAGNOSING FAILED PROCESSES

Operational Status and Tracking shows that a CA MICS production process (e.g., DAILY) has FAILED.

FAILED means that the today's processing terminated due to an error. You must correct the error and restart scheduled processing.

Determine why the process failed.

- The Operational Status and Tracking display lists the operational job and step where the failure occurred.

- Issue the STATUS command or review the Run Status Report to gather additional information.

  - The STATUS display lists the scheduled CA MICS operational process, last completed job step, the failing job step, and may also list the abnormal termination code (abend code).

  - If the STATUS display does NOT list the abend code, refer to the batch job printed outputs, particularly the JES log and JES messages.

- Look up the abend code in the Reference section of this guide. If the abend is not covered in the Reference section, refer to IBM documentation for information on causes and corrective action.

- Examine the MICSLOG from the failing job step for error messages documenting the cause of job termination.

- Examine the JES log and messages output from the failing job for additional information concerning an abend or other failure.

- If the failure occurred during CA MICS Database aging, call CA MICS Product Support for assistance.

- If the failure occurred in step DAY199, refer to the CA MICS Accounting and Chargeback User Guide for additional information and guidance.

The research done here determines what to do next. For the most part, you will correct the error and restart the failing job. In some cases, you may be instructed to restore the Database from a backup and rerun scheduled processing from the beginning (along with any other processing executed since the backup was taken).

Restart and recovery are discussed in section 4.2.5.

Contact CA MICS Product Support if you need assistance in problem diagnosis.

DIAGNOSIS EXAMPLE

The Operational Status and Tracking display (see sample panels on next page) shows that MONTHLY processing for the R (REMOTE) unit failed. Last completed job step was DAYALL.

The Operational Status and Tracking STATUS command shows that the DAILY job failed in step DAYSMF with an S001 abend.

IBM documentation defines an S001 abend as an I/O error.

The JES log includes a message which indicates that the I/O error occurred on the INPUTSMF DD statement.

The Reference section in this guide shows that the INPUTSMF DD statement is used in the DAYSMF step to read the input data.

Thus, the CA MICS DAILY job failed because of an I/O error on the input data set. Restart and recovery are covered in section 4.2.5, but some notes on S001 abend recovery follow to complete this example.

o If the problem is due to a hardware failure (e.g., a dirty tape drive), you can restart the DAILY job at step DAYSMF.

o If the problem is a media problem (e.g., a bad tape) and a backup copy is available, you can restart the DAILY job at step DAYSMF pointing CA MICS to the backup input tape.

o If the backup copy of the input data is on a consolidated tape (e.g., a week-to-date raw SMF tape), you can use the Select facility to select just the current data from the consolidated input tape.

o If a backup copy of the input is not available, you will need to try to salvage the tape using system utilities to copy the tape, skipping unreadable records. The data is incomplete, but this may be more tolerable than not having data at all.

```
----------------------  Operational Status and Tracking  ---------  ROW 1 OF 6
Command ===> STATUS R                                         Scroll ===> CSR


Commands:  Schedule, Daily, Weekly, Monthly, Yearly, Backup, Restore, Restart,
           Status/History/Checkpt/Joblog, Suspend/Resume, Force

           Database       Current Operation     Last Completed    Edit Suspend
  Cmd    ID  Label Type   as of 10 OCT 2001     Job/Step Date     JCL  Updates
-------- -- -------- -   -------------------  ------------------  ---    ---
________ C  CICS     U   MONTHLY  Completed   MONTHLY  10OCT2001  NO     NO
________ D  DASD     U   MONTHLY  DUE TODAY   WEEKLY   09OCT2001  NO     NO
________ I  IMS      U   MONTHLY  Completed   MONTHLY  10OCT2001  NO     NO
________ P  PRIMARY  P   MONTHLY  FAILED      DAY020   10OCT2001  NO     NO
________ R  REMOTE   U   MONTHLY  FAILED      DAYALL   10OCT2001  NO     NO
________ T  TEST     T   DAILY    OVERDUE     DAILY    19SEP2001  NO     NO
****************************** BOTTOM OF DATA  ********************************
```

. . . .

```
---------------------------  Unit Database Status  ----------------------------

 Command ===>


Database:  R  (REMOTE) - REMOTE DATA CENTER

The status information was recorded at:  10OCT01 08:45
Status of this unit Database:            NON-UPDATABLE
Status of the cycle aging process:       Completed


 CA MICS  Last Completed
 Job      Step & Date     Status of Current Operation:   MONTHLY
--------  ---  -------    -----------------------------------------------------
DAILY     ALL  10OCT01    FAILED    DAYSMF                S001
MONTHLY   900  05SEP01    HELD      Prior job failed
BACKUP    900  09OCT01    HELD      Prior job failed

                          Status of Other Jobs
                          -----------------------------------------------------
WEEKLY    900  09OCT01    Completed
YEARLY    900  10JAN01    Completed
RESTORE   900  09OCT01    Completed
```

. . . .

4.2.5 System Restart and Recovery

Restart means that after the problem is fixed, CA MICS operational processing can be resumed at the job step that failed. When CA MICS internal step restart is enabled for the failing job step, restarting the step means that processing will automatically resume at the last completed processing phase (at the last "restart checkpoint") within the database update job step.

Recovery means that after the problem is fixed, the Database must be restored from a backup and the CA MICS operational processing must be rerun from the beginning (along with any other processing executed since the backup was taken).

You can restart a failed operational job through CA MICS Operational Status and Tracking or you can restart it manually. For purposes of this discussion, manual restart includes using your installation's production scheduler restart procedures.

Note: Operational Status and Tracking does NOT support the incremental update SPLITSMF or INCRccc jobs. If the SPLITSMF or an INCRccc job fails, it must be restarted manually or through your installation's production scheduler restart procedures.

FAILURE DURING DATABASE AGING

If the update job fails during the Database aging process, call CA Technical Support for assistance before trying to restart CA MICS processing. If internal step restart is enabled for the failing job step, you will normally be able to restart the Database aging process and complete DAILY processing. However, due to the critical nature of the Database aging process, it is still wise to seek guidance from CA Technical Support.

OPERATIONAL STATUS AND TRACKING RESTART

Enter the RESTART command to restart the failing operational process at the job step that failed.

o Verify the batch job step (displayed on the RESTART panel) where processing will be restarted.

o If the "Rebuild DAYSMF step temporary work files" prompt is displayed on the RESTART panel:

  - And if the DAYSMF work files are still cataloged, reply NO (the default).

  - And if the DAYSMF work files have been deleted, reply YES to rebuild these data sets.

o If the "Restore Accounting and Chargeback audit files" prompt is displayed on the RESTART panel, review the online tutorial and respond according to your requirements. See the CA MICS Accounting and Chargeback User Guide for more information.

o Edit the generated JCL if required to point CA MICS to an alternate or backup input data source. Make other JCL changes as necessary.

o If internal step restart is enabled for the batch job step where processing will be restarted, then processing will automatically resume at the last completed processing phase in this job step.

o If you need to override automatic internal step restart and force the step to start from the beginning, specify SYSPARM=NORESTART on the JCL EXEC statement for this batch job step.

If you did NOT schedule CA MICS processing through Operational Status and Tracking or using the CA MICS batch SCHEDULE job, then verify that subsequent scheduled processing is executed. For example, if you submitted the DAILY job manually, remember to run BACKUP after DAILY completes.

MANUAL OR PRODUCTION SCHEDULER RESTART

If you are restarting the DAILY job, and you specified DAYSMF FILES TEMPORARY in the JCLDEF member of prefix.MICS.PARMS:

- And if the DAYSMF work files have been deleted, submit the job in prefix.MICS.CNTL(DAYSMFR) and wait for it to complete.

- Do NOT continue with the restart until DAYSMFR completes successfully.

If the operational job failed in the DAY199 step (CA MICS Accounting and Chargeback step):

- Submit the job in prefix.MICS.CNTL(ACTDAY1R). See the CA MICS Accounting and Chargeback User Guide for more information on restarting after failures in step DAY199.

- Do NOT continue with the restart until ACTDAY1R completes successfully.

If the failing job was submitted by Operational Status and Tracking or by the CA MICS SCHEDULE job:

- Edit prefix.MICS.RESTART.CNTL.

- Enter the RESTART= parameter on the job statement as noted on the Run Status Report -- for example,

      RESTART=(DAY030.MICS).

  If internal step restart is enabled for this batch job step, then processing will automatically resume at the last completed processing phase within this step.
- If you need to override automatic internal step restart and force the step to start from the beginning, specify SYSPARM=NORESTART on the JCL EXEC statement for this batch job step.

- Edit the DD statements if required to point CA MICS to an alternate or backup input data source.

- Submit the job stream.

If the failing job was submitted manually from prefix.MICS.CNTL or by your production scheduler:

- Edit the JCL for the failing job in prefix.MICS.CNTL or the scheduling facility.

- Enter the RESTART= parameter on the job statement if required. The correct RESTART= parameter is noted on the Run Status Report--for example, RESTART=(DAY030.MICS). If internal step restart is enabled for this batch job step, then processing will automatically resume at the last completed processing phase within this step.

- If you need to override automatic internal step restart and force the step to start from the beginning, specify SYSPARM=NORESTART on the JCL EXEC statement for this batch job step.
- Edit the DD statements if required to point CA MICS to an alternate or backup input data source. Make other JCL changes as required.

- Submit the job stream.

- CANCEL the edit session so that the RESTART= is not permanently part of the job.

- Verify that subsequent scheduled processing is executed. For example, remember to run BACKUP after DAILY completes.

FAILURE DURING INTERNAL STEP RESTART

If the restarted update job step fails, examine the CA MICS and SAS logs to determine the cause of the restart failure.

- If the CA MICS log contains,

  ```
  *** ABORT ERROR ***
  PREVIOUS DAYnnn EXECUTION FAILED DURING DATABASE AGING
  DATABASE AGING RECOVERY AND RESTART IS NOT POSSIBLE.
  PLEASE CONTACT CA TECHNICAL SUPPORT FOR ASSISTANCE.
  ```

  then the original failure occurred during the database aging process. You can not restart a job step after failure in database aging. Call CA Technical Support for assistance.

- If the CA MICS log contains,

  ```
  >ERR> Invalid checkpoint.  Unable to restart ccc
          product DAILY update......
  ```

  then the internal step restart process determined that one or more information items critical to restarting the database update job step are missing. Specify,

  ```
  SYSPARM=NORESTART
  ```

  on the JCL EXEC statement to force the job step to repeat processing from the beginning.

- If the SAS log indicates that the job failed due to a shortage of disk space on one of the WORKnn data sets (where nn is 01 - 99) or the cccXWORK data set (where ccc is the product associated with this database update step),

  - Edit the operational job JCL for the step that failed and add a PARMOVRD DD stream containing the WORK and/or RESTARTWORK parameters to temporarily override the data set allocation parameters for the failing data sets to increase the space allocation. For example,

    ```
    //PARMOVRD DD *
     WORK   SPACE=(CYL,(50,50)) STORCLAS=MICSTEMP
     RESTARTWORK SPACE=(CYL,(50,50))
     RESTARTWORK STORCLAS=MICSTEMP
    ```

  - Restart the database update job step from the beginning by specifying,

    ```
    SYSPARM=NORESTART
    ```

    on the JCL EXEC statement.

  - After the job step completes successfully, remove the PARMOVRD DD stream to resume using the data set allocation parameters you specified in prefix.MICS.PARMS(cccOPS). If you believe that the temporary change to the data set allocation parameters should be made permanent, then increase the amount of space requested on the cccOPS WORK (for WORKnn data sets) or RESTARTWORK (for the cccXWORK data set) parameter and run cccPGEN.

- If the SAS log indicates that the job failed due to a shortage of disk space on the cccXCKPT data set (where ccc is the product associated with this database update job step), call CA Technical Support for assistance.

RESTART EXAMPLE

The Operational Status and Tracking display (see sample panels on the next two pages) shows that MONTHLY processing for the P (PRIMARY) unit failed. The last completed job step was DAY020.

The Operational Status and Tracking STATUS command shows that the DAILY job failed in step DAY030 with a U310 abend.

The RESTART command invokes Operational Status and Tracking restart processing for the P (PRIMARY) unit.
The RESTART Database Update panel shows that processing will be restarted in step DAY030. DAYSMF temporary work files will not be recreated as they are still cataloged. Since Operational Status and Tracking submitted the MONTHLY processing, MONTHLY and BACKUP will automatically follow DAILY job restart.

Note: If internal step restart is enabled for this batch job step, then processing will automatically resume at the last completed processing phase within this step.

```
----------------------  Operational Status and Tracking  ---------  ROW 1 OF 6
Command ===> STATUS P                                         Scroll ===> CSR


Commands:  Schedule, Daily, Weekly, Monthly, Yearly, Backup, Restore, Restart,
           Status/History/Checkpt/Joblog, Suspend/Resume, Force

           Database       Current Operation     Last Completed    Edit Suspend
  Cmd    ID  Label Type   as of 10 OCT 2001     Job/Step Date     JCL  Updates
-------- -- -------- -   -------------------  ------------------  ---    ---
________ C  CICS     U   MONTHLY  Completed   MONTHLY  10OCT2001  NO     NO
________ D  DASD     U   MONTHLY  DUE TODAY   WEEKLY   09OCT2001  NO     NO
________ I  IMS      U   MONTHLY  Completed   MONTHLY  10OCT2001  NO     NO
________ P  PRIMARY  P   MONTHLY  FAILED      DAY020   10OCT2001  NO     NO
________ R  REMOTE   U   MONTHLY  FAILED      DAYALL   10OCT2001  NO     NO
________ T  TEST     T   DAILY    OVERDUE     DAILY    19SEP2001  NO     NO
****************************** BOTTOM OF DATA  ********************************



.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

```
---------------------------  Unit Database Status  ----------------------------

 Command ===>


Database:  P  (PRIMARY) - CA MICS PRIMARY DATABASE

The status information was recorded at:  10OCT01 08:06
Status of this unit Database:            NON-UPDATABLE
Status of the cycle aging process:       Completed


 CA MICS  Last Completed
 Job      Step & Date     Status of Current Operation:   MONTHLY
--------  ---  -------    ------------------------------------------------------
DAILY     ALL  10OCT01    FAILED    DAY030                U310
MONTHLY   900  03SEP01    HELD      Prior job failed
BACKUP    900  09OCT01    HELD      Prior job failed

                          Status of Other Jobs
                          ------------------------------------------------------
WEEKLY    900  09OCT01    Completed
YEARLY    900  10JAN01    Completed
RESTORE   900  09OCT01    Completed


Database:  P  (PRIMARY) - CA MICS PRIMARY DATABASE

The status information was recorded at:  10OCT01 08:06
Status of this unit Database:            NON-UPDATABLE
Status of the cycle aging process:       Completed


 CA MICS  Last Completed
 Job      Step & Date     Status of Current Operation:   MONTHLY
--------  ---  -------    ------------------------------------------------------
DAILY     ALL  10OCT01    FAILED    DAY030                U310
MONTHLY   900  03SEP01    HELD      Prior job failed
BACKUP    900  09OCT01    HELD      Prior job failed
```

```
                          Status of Other Jobs
                          -----------------------------------------------------
WEEKLY    900  09OCT01    Completed
YEARLY    900  10JAN01    Completed
RESTORE   900  09OCT01    Completed


.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .


----------------------  Operational Status and Tracking  ---------  ROW 1 OF 6
Command ===> RESTART P                                        Scroll ===> CSR


Commands:  Schedule, Daily, Weekly, Monthly, Yearly, Backup, Restore, Restart,
           Status/History/Checkpt/Joblog, Suspend/Resume, Force


           Database       Current Operation     Last Completed    Edit Suspend
  Cmd    ID  Label Type   as of 10 OCT 2001     Job/Step Date     JCL  Updates
-------- -- -------- -   -------------------  ------------------  ---    ---
________ C  CICS     U   MONTHLY  Completed   MONTHLY  10OCT2001  NO     NO
________ D  DASD     U   MONTHLY  DUE TODAY   WEEKLY   09OCT2001  NO     NO
________ I  IMS      U   MONTHLY  Completed   MONTHLY  10OCT2001  NO     NO
________ P  PRIMARY  P   MONTHLY  FAILED      DAY020   10OCT2001  NO     NO
________ R  REMOTE   U   MONTHLY  FAILED      DAYALL   10OCT2001  NO     NO
________ T  TEST     T   DAILY    OVERDUE     DAILY    19SEP2001  NO     NO
****************************** BOTTOM OF DATA  ********************************
```

```
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

```
--------------------------  RESTART Database Update  --------------------------

 Command ===>

Database:  P  (PRIMARY) - CA MICS PRIMARY DATABASE

The status information was recorded at:  10OCT01 08:10
The Database update job that failed:     DAILY
The job will be restarted at step:       DAY030


Edit the job stream before batch submit    ===> YES (YES/NO)
Rebuild DAYSMF step temporary work files   ===> NO  (YES/NO)




Press END (or enter the END command) to generate and submit the RESTART job.
Enter CANCEL to terminate RESTART processing for this unit Database.
```

```
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

DATABASE RECOVERY

If the update job fails due to an I/O error on the CA MICS Database or due to insufficient DASD space in the Database, you will need to recover the Database. Database recovery involves:

- Restoring the Database from a backup copy.

- Resolving CA MICS applications issues, for example Accounting and Chargeback file recovery.

- Rerunning operational processing executed since the backup was taken.

Contact CA Technical Support for assistance before recovering the CA MICS Database. The remainder of this section provides basic instructions on the kinds of issues involved.

OPERATIONAL STATUS AND TRACKING RECOVERY

Use the CA MICS Operational Status and Tracking RESTORE command to restore the Database from a standard or monthly backup.

- Review the operational processing log displayed by the RESTORE command.

- Select the standard or monthly backup that meets your requirements. Operational Status and Tracking will generate and submit the RESTORE job.
- Wait for the RESTORE job to complete.

- Examine the RESTORE job MICSLOG and SAS log outputs. If incremental update is active for one or more products in the unit database, then the RESTORE job MICSLOG messages may instruct you to run the IUDBINIT job. IUDBINIT re-initializes incremental update database files in order to correctly recover the CA MICS database.

- Identify any CA MICS operational jobs that must be rerun to recover data in the CA MICS Database since the BACKUP was taken, and run those jobs.

If CA MICS Accounting and Chargeback is installed in the unit database:

o Submit the job in prefix.MICS.CNTL(ACTDAY1R). Edit the JCL to restore the ACTAUDIT DAY1 file from the DAY2 generation that corresponds to the Database backup you selected. DO NOT SAVE THE MODIFIED JCL.

o See the CA MICS Accounting and Chargeback User Guide for more information on restoring a unit database with accounting.

o Do not restart CA MICS operational processing until the ACTDAY1R job completes.

MANUAL OR PRODUCTION SCHEDULER RECOVERY

Use the CA MICS Operational Status and Tracking JOBLOG command to review the standard and monthly backups available for use in restoring the Database.

Submit the job in prefix.MICS.CNTL(RESTORE). Edit the job if you want to restore from a backup other than the most recent (0) generation standard Database backup. DO NOT SAVE THE MODIFIED JCL.

o To restore from a backup generation other than the 0 generation, specify the desired backup generation (for example, -1) in the cataloged procedure GDG parameter.

o To restore from a monthly backup, specify the monthly backup data set name prefix of the desired backup in the cataloged procedure DSNPREF parameter.
o Wait for the RESTORE job to complete.

o Examine the RESTORE job MICSLOG and SAS log outputs. If incremental update is active for one or more products in the unit database, then the RESTORE job MICSLOG messages may instruct you to run the IUDBINIT job. IUDBINIT re-initializes incremental update database files in order to correctly recover the CA MICS database.

o Identify any CA MICS operational jobs that must be rerun to recover data in the CA MICS Database since the BACKUP was taken and run those jobs.

If CA MICS Accounting and Chargeback is installed in the unit database:

- o Submit the job in prefix.MICS.CNTL(ACTDAY1R). Edit the JCL to restore the ACTAUDIT DAY1 file from the DAY2 generation that corresponds to the Database backup you selected. DO NOT SAVE THE MODIFIED JCL.

- o See the Accounting and Chargeback User Guide for more information on restoring a unit database with accounting.

- o Do not restart CA MICS operational processing until the ACTDAY1R job completes.

RECOVERING TABLES AND SCREENS

Call CA Technical Support for assistance.

Use the Operational Status and Tracking JOBLOG command to review the standard and monthly backups available for the PRIMARY unit database. The TABLES and SCREENS data sets are backed up by the PRIMARY unit.

Determine whether or not changes have been made to the TABLES and SCREENS data sets since the last backup. See the Batch and Operations Analyzer Guide and the Accounting and Chargeback User Guide for more information on TABLES and SCREENS data set contents/changes.

Submit the job in prefix.MICS.CNTL(RSTRTBLS). Edit the JCL to restore TABLES and SCREENS from the standard or monthly backup generation that meets your requirements. DO NOT SAVE THE MODIFIED JCL.

Repeat all processing and manual data entry that updated the TABLES or modified the SCREENS data set since the date of the backup.

RECOVERING ISPTLIB

Call CA Technical Support for assistance.

Do not attempt to restore sharedprefix.MICS.ISPTLIB without first consulting CA Technical Support. You may be able to restore JUST the ISPF tables that are damaged without losing sharedprefix.MICS.ISPTLIB changes.

Use the Operational Status and Tracking JOBLOG command to review the standard and monthly backups available for the PRIMARY unit database. Sharedprefix.MICS.ISPTLIB is backed up by the PRIMARY unit.

Determine all changes made to sharedprefix.MICS.ISPTLIB since the last backup. This includes CA MICS product, parameter, and JCL generation jobs; Accounting and Chargeback parameters; MICF shared inquiries; MICF production reporting definitions; etc. Since processing by individual CA MICS products can make changes to sharedprefix.MICS.ISPTLIB, use the Subject Cross Reference facility to locate information that refers to processes that change the contents of sharedprefix.MICS.ISPTLIB.

Submit the job in prefix.MICS.CNTL(RSTRTLIB). Edit the JCL to restore sharedprefix.MICS.ISPTLIB from the standard or monthly backup generation that meets your requirements. DO NOT SAVE THE MODIFIED JCL.

Repeat all processing, data entry, and parameter changes that modified sharedprefix.MICS.ISPTLIB contents since the date of the backup.

RECOVERING INCREMENTAL UPDATE FILES

If an INCRccc job or the DAILY job fail due to I/O errors on an incremental update DETAIL or DAYS timespan file, you will generally need to rerun the INCRccc or DAILY job with all of the input data processed so far today. The incremental update data sets exist only until the next DAYnnn step completes execution and are not included in BACKUP processing.

To recover from a failure due to a damaged incremental update DETAIL or DAYS data set,

o Identify the input data that has been processed by INCRccc and/or DAYnnn jobs for this product since the last successful DAYnnn execution.

o Delete the incremental update checkpoint and database data sets.

o Use the cccIUALC job to recreate the incremental update checkpoint and database data sets.

o Restart the failing INCRccc job or the failing DAILY job DAYnnn step with ALL of the data that has been processed for this product since the last successful DAYnnn execution.

RECOVERING SPLITSMF JOB OUTPUT FILES

The SPLITSMF job dynamically allocates and populates data sets with subsets of the input SMF data for processing by the INCRccc jobs. If an INCRccc job fails due to a missing INPUTSMF data set that was originally created by the SPLITSMF job, then you will need to rerun the SPLITSMF job to recreate the input data file prior to restarting the INCRccc job. The INCRccc job deletes the INPUTSMF data set at successful termination.

4.2.6 Entering Data Into CA MICS

Most CA MICS files are derived from measurement data produced by one of the common software monitors such as SMF or CA TSO/MON PM. For some CA MICS files, data can be entered manually, either to update standard measurement data or as the sole source of input to the file.

The MWF SAS With CA MICS Libraries (MSAS) application supports manual data entry for the Operations Information Area of the Batch and Operations Analyzer. See the CA MICS Batch and Operations Analyzer Guide for information on Operations Information Area files and data entry requirements.

For instructions on manual data entry for the following CA MICS products, see the following guides:

o CA MICS Accounting and Chargeback User Guide for the Accounting Information Area of CA MICS Accounting and Chargeback

o CA MICS Capacity Planner Guide for the capacity planning application Database (CAPACITY).

PREPARATION

Authorize data entry personnel to access the SAS With CA MICS Libraries (MSAS) MWF application.

Authorize data entry personnel to update the appropriate CA MICS Database files.

Train data entry personnel in using the MSAS option for Operations Information Area data entry.

OPERATIONS INFORMATION AREA MANUAL DATA ENTRY

Select the SAS With CA MICS Libraries (MSAS) option of MWF.

Select the CA MICS Data Entry option (4) from the MSAS menu.

On the CA MICS Data Entry (SCREENS) panel,

o Enter D (Data Entry) for SAS/FSP Function.

o Enter 1 (IPL Outage Times and Reason Codes) or 2

(Environmental Change Tracking) for Data Source.

- o Leave Cycle Number and File Suffix blank.

- o Press ENTER.

On the CA MICS Database Unit Selection panel,

- o If you specified 1 (IPL Outage and Reason Codes) for Data Source, select the CA MICS unit Database (DB) ID for which you want to enter IPL outage and reason code information.

- o Press END to begin Operations Information Area data entry. See the CA MICS Batch and Operations Analyzer Guide for more information.

4.3 Operations Reference

The reference material in the following sections is not intended to be a tutorial. It is arranged by topic to provide greater detail about areas mentioned in earlier sections of this chapter.

The following sections contain reference material:

4.3.1 Processing Schedule

A processing schedule is a set of rules used to determine which operational jobs should occur on any given day. Use of a processing schedule helps to ensure that the appropriate CA MICS jobs run in a predetermined order.

CA MICS creates and maintains a processing schedule based on the specifications coded during CA MICS installation. Scheduling control parameters include the following:

- The CA MICS calendar specified with the 13MONTHYEAR statement in sharedprefix.MICS.PARMS(CPLXDEF) or prefix.MICS.PARMS(SITE)

- Backup frequency specified in prefix.MICS.PARMS(JCLDEF)

- Whether WEEKLY, MONTHLY, and/or YEARLY processing can occur during weekends specified in prefix.MICS.PARMS(EXECDEF)

The following table summarizes the standard CA MICS scheduling algorithm--when each job should be executed and what job needs to successfully complete prior to the job in question:

Operational Job	When should it be run?	Job's Dependency
SPLITSMF	One or more times per day when input data is available (Optional)	Previous INCRccc
INCRccc	One or more times per day when input data is available	Previous INCRccc or DAILY
DAILY	Each day when input data is available	Previous DAILY or INCRccc
WEEKLY	Two days after end of week (Monday)	DAILY
MONTHLY	Three days after end of month	DAILY
YEARLY	Five days after end of year	MONTHLY
BACKUP	After each day's complete updating activity	NONE

Figure 4-1. Job Scheduling and Dependencies

```
NOTE:  The CA MICS scheduling algorithm does not include the
       optional AUDIT, HISTW, and HISTM operational jobs for
       weekly/monthly archive tape processing.

o  For Archive Audit processing:

     -  If you specified
             ARCHIVE AUDIT YES JOB AUTOSUBMIT
        in prefix.MICS.PARMS(JCLDEF), CA MICS automatically
        submits the AUDIT job during weekly processing.

     -  If you specified
             ARCHIVE AUDIT YES JOB
        in prefix.MICS.PARMS(JCLDEF) without the optional
        AUTOSUBMIT parameter, you must schedule the AUDIT job
        separately.

     -  If you specified
             ARCHIVE AUDIT YES STEP
        in prefix.MICS.PARMS(JCLDEF) or accepted the  default
        (STEP), then Archive Audit processing occurs within
        the WEEKLY job WEEK300 step.

     -  If you specified
             ARCHIVE AUDIT NO
        in prefix.MICS.PARMS(JCLDEF), then Archive Audit
        processing is bypassed.

o  For Archive Weekly History processing:

     -  If you specified
             ARCHIVE HISTW YES JOB AUTOSUBMIT
        in prefix.MICS.PARMS(JCLDEF), CA MICS automatically
        submits the HISTW job during weekly processing.

     -  If you specified
             ARCHIVE HISTW YES JOB
        in prefix.MICS.PARMS(JCLDEF) without the optional
        AUTOSUBMIT parameter, you must schedule the HISTW job
        separately.

     -  If you specified
             ARCHIVE HISTW YES STEP
        in prefix.MICS.PARMS(JCLDEF) or accepted the default
        (STEP), then Archive Weekly History processing occurs
        within the WEEKLY job WEEK300 step.

     -  If you specified
```

ARCHIVE HISTW NO
in prefix.MICS.PARMS(JCLDEF), then Archive Weekly History processing is bypassed.

- For Archive Monthly History processing:

    - If you specified
      ARCHIVE HISTM YES JOB AUTOSUBMIT
      in prefix.MICS.PARMS(JCLDEF), CA MICS automatically submits the HISTM job during monthly processing.

    - If you specified
      ARCHIVE HISTM YES JOB
      in prefix.MICS.PARMS(JCLDEF) without the optional AUTOSUBMIT parameter, you must schedule the HISTM job separately.

    - If you specified
      ARCHIVE HISTM YES STEP
      in prefix.MICS.PARMS(JCLDEF) or accepted the default (STEP), then Archive Monthly History processing occurs within the MONTHLY job MONTH300 step.

    - If you specified
      ARCHIVE HISTM NO
      in prefix.MICS.PARMS(JCLDEF), then Archive Monthly History processing is bypassed.

NOTE: The CA MICS scheduling algorithm does not include the DAILYRPT, WEEKRPT, and MONTHRPT operational jobs for daily/weekly/monthly MICF production reporting.

- If you specified
  AUTOSUBMIT YES
  in prefix.MICS.PARMS(JCLDEF), CA MICS automatically submits the DAILYRPT, WEEKRPT, and MONTHRPT jobs during daily/weekly/monthly processing.

- If you specified
  AUTOSUBMIT NO
  in prefix.MICS.PARMS(JCLDEF), you must schedule the DAILYRPT, WEEKRPT, and MONTHRPT jobs separately.

CA MICS provides two automated processes to implement the processing schedule.

- Operational Status and Tracking SCHEDULE command

- The SCHEDULE job--see prefix.MICS.CNTL(SCHEDULE)

NOTE: Neither Operational Status and Tracking nor the SCHEDULE job support incremental update. You must schedule the SPLITSMF (if used) and INCRccc jobs either manually or using your installation's production batch job scheduling facility.

Both the SCHEDULE command and the SCHEDULE batch job build a tailored job stream for each unit database using the following rules:

o Start the job stream by copying the DAILY job into today's job stream.

   DAILY should be run as soon as practical after input data becomes available and after all cccINCR jobs (if any) in this unit complete successfully. This is normally done after midnight for the day's activity that is being processed. Most organizations schedule the DAILY to complete before the start of prime shift so that the reports for yesterday's activity are ready for review and the database is current.

   If a restart is required, the DAILY job must successfully complete before the start of the next scheduled update.

   NOTE: When an operational job does not successfully complete, the database is marked non-updatable. Consequently, succeeding DAILY, WEEKLY, MONTHLY, or YEARLY jobs will not be processed until the abnormally terminated jobs are successfully completed.

   For jobs using SMF data as an input source, the SUSPENDLIMIT keyword in prefix.MICS.PARMS(SMFOPS) determines how long to retain data for jobs that are still in process (for example, jobs waiting to be printed or in the system when it failed). During each day's processing, CA MICS merges suspended data with the new data in an attempt to complete suspended jobs' data. The SUSPENDLIMIT keyword sets the maximum number of days for which CA MICS retains job level data for the matching attempt. For more information on SUSPENDLIMIT, see section 7.3.1.16 of the Batch and Operations Guide.

  o SCHEDULE also prohibits a WEEKLY, MONTHLY, or YEARLY from running in combination with one another.

  o If a WEEKLY is due and one day has elapsed since it became due, copy the WEEKLY JCL into today's job stream. To determine if a WEEKLY is due, SCHEDULE computes the week start using the day of the week (keyword WEEKSTART) specified in prefix.MICS.PARMS(SITE). By default, the week starts on Sunday.

For an organization operating 7 days a week, WEEKLY should be executed following completion of the DAILY between 0:00 and 8:00 on Monday. This makes the weekly reports (for the previous week's activity) available by the beginning of prime shift on Monday.

o If a MONTHLY is due, three days have elapsed since it became due, and a WEEKLY has not already been scheduled, copy the MONTHLY job into today's job stream.

For an organization running a twelve month year, MONTHLY should be executed following completion of the DAILY between 0:00 and 8:00 on the 3rd day of the month.

o If a YEARLY is due, five days have elapsed since it became due, and a WEEKLY or MONTHLY has not already been scheduled, copy the YEARLY JCL into today's job stream.

For an organization running a standard month (30 or 31 days, except for February), YEARLY should be executed following completion of the DAILY run after the 5th day of the new year. This makes the yearly reports available in a timely fashion for the previous year's activity.

o SCHEDULE will not schedule a WEEKLY, MONTHLY, and YEARLY on the same day. If any combination of these jobs are due to be scheduled on the same day, SCHEDULE will schedule the lesser job. For example, if a MONTHLY and YEARLY are due at the same time, the MONTHLY job will be scheduled.

o If a BACKUP is required because one of the following statements is true, then copy the BACKUP job into today's job stream.

- the backup frequency specified in prefix.MICS.PARMS(JCLDEF) has been reached

- a WEEKLY job has been scheduled

- a MONTHLY job has been scheduled

- a YEARLY job has been scheduled

It is important to include BACKUP in each day's process because if the CA MICS database becomes damaged, the entire database can be restored without reprocessing raw data.

o Save today's job stream in the Schedule Restart File in prefix.MICS.RESTART.CNTL.

o Submit the generated job stream (SCHEDULE command) or write the job stream to the internal reader (SCHEDULE job--the internal reader will submit the job for execution).

```
******************** IMPORTANT NOTICE ********************
* If your installation uses the Thirteen Month Fiscal   *
* Year Option or another special CA MICS                 *
* calendar, the WEEKLY, MONTHLY, and YEARLY              *
* jobs are run after the user-defined week,              *
* month, and year ends and are not tied to the           *
* standard calendar week, month, and year.               *
*                                                        *
* You can review the processing scheduled for today and  *
* the next few days through the Operational Status and   *
* Tracking displays.                                     *
*                                                        *
* You can build a job stream for today's scheduled       *
* processing without submitting the job for execution.   *
*                                                        *
*  o  With Operational Status and Tracking,              *
*                                                        *
*     -  Enter the SCHEDULE command with the EDIT        *
*        operand.  Operational Status and Tracking will  *
*        display the generated batch job stream under    *
*        PDF Edit.                                       *
*                                                        *
*     -  You can save the generated job stream in the    *
*        Schedule Restart File by entering END on the    *
*        PDF Edit display.  Enter CANCEL to exit         *
*        without saving the generated job stream.        *
*                                                        *
*  o  With the SCHEDULE job (prefix.MICS.CNTL(SCHEDULE)),*
*                                                        *
*     -  Specify NOSUBMIT on the SCHEDULE job's EXEC     *
*        statement as follows:                           *
*                                                        *
*          //SCHEDULE EXEC MICSNDB,SYSPARM='NOSUBMIT'    *
*                                                        *
*     -  The generated job will be saved in the Schedule *
```

```
   *        Restart File (prefix.MICS.RESTART.CNTL).        *
   *                                                         *
   ***********************************************************
```

SCHEDULE DELAY

To determine when a WEEKLY, MONTHLY, or YEARLY job needs to be run, SCHEDULE uses today's date and the date the last WEEKLY, MONTHLY, and/or YEARLY job successfully completed. SCHEDULE determines if a period has changed (i.e., from one calendar week, month, or year to the next) and if the required job's delay has expired. It then constructs the JCL for the appropriate processes for execution in today's job stream.

The table at the beginning of this section summarizes the weekly, monthly, and yearly schedule delays. The schedule delay causes the job to be scheduled n number of days after its calendar scheduling date so that WEEKLY executes one day after the beginning of the week (on Monday for the default WEEKSTART specification), MONTHLY on the third day of the month, and YEARLY on the fifth day of the new year (on January 5th for the standard calendar year).

You can change the delay in scheduling the WEEKLY, MONTHLY, or YEARLY with a local modification that changes the variables WKDELAY, MNDELAY, and YRDELAY.

- Make a local modification to sharedprefix.MICS.SOURCE(SCHEDULE) for the SCHEDULE batch job.

- Make a local modification to sharedprefix.MICS.SOURCE(MAFOSCHD) for the Operational Status and Tracking SCHEDULE command. This change will be effective until after the next DAILY, WEEKLY, or MONTHLY job completes successfully.

Follow the instructions in the user modifications section of the CA MICS System Modification Guide (SMG) to implement changes to sharedprefix.MICS.SOURCE members.

4.3.2 Implementing the Processing Schedule

CA MICS provides two automated processes to implement the processing schedule.

o Operational Status and Tracking's SCHEDULE command.

o The SCHEDULE job--see prefix.MICS.CNTL(SCHEDULE).

NOTE: If you enable incremental update for one or more products in a unit database, you must schedule the optional SPLITSMF job and associated INCRccc job executions separately, either manually or with your installations production batch job scheduling facility. The CA MICS automated scheduling facilities do not handle INCRccc job scheduling.

The following diagram summarizes Operational Status and Tracking's SCHEDULE command.

```
+-----------------------------------------------------------+
|                                                           |
|      CA MICS Job Control Library (prefix.MICS.CNTL)       |
|                                                           |
|                          DAILY                            |
|                          BACKUP                           |
|                          WEEKLY                           |
|         Checkpoint       MONTHLY      Today's Date        |
|            File          YEARLY        (From MVS)         |
|             |              |               |              |
|             |              |               |              |
|             V______________V_______________V              |
|                            |                              |
|                            V                              |
|                       +----------+       +---------+      |
|     User Enters       |          |       | Online  |      |
|       SCHEDULE ------>| SCHEDULE |------>| Control |      |
|    Command Daily      |          |       | Tables  |      |
|                       +----------+       +---------+      |
|                            |                              |
|                            V                              |
|                      Today's Tailored                     |
|                         Job Stream                        |
|                            |                              |
|         +------------------+------------------+           |
|         |                  |                  |           |
|         V                  V                  V           |
|     Checkpoint            Job              Schedule       |
|        File              Input             Restart        |
|                          Queue              File          |
|                                                           |
+-----------------------------------------------------------+
```

The following diagram summarizes the SCHEDULE job.

```
+-----------------------------------------------------------+
|                                                           |
|      CA MICS Job Control Library (prefix.MICS.CNTL)       |
|                                                           |
|                          DAILY                            |
|                          BACKUP                           |
|                          WEEKLY                           |
|         Checkpoint       MONTHLY      Today's Date        |
|            File          YEARLY        (From MVS)         |
|             |              |               |              |
|             |              |               |              |
|             V______________V_______________V              |
|                            |                              |
|                            V                              |
|                       +----------+       +--------+       |
|          User         |          |       |  Run   |       |
|         Submits------>| SCHEDULE |------>| Status |       |
|        SCHEDULE       |          |       | Report |       |
|         Daily         +----------+       +--------+       |
|                            |          (only produced on   |
|                            V          SCHEDULE abnormal   |
|                          Today's         termination)     |
|                          Tailored                         |
|                         Job Stream                        |
|                            |                              |
|         +------------------+------------------+           |
|         |                  |                  |           |
|         V                  V                  V           |
|     Checkpoint           System           Schedule        |
|        File             Internal          Restart         |
|                          Reader             File          |
|                            |                              |
|                            |                              |
|                            |                              |
|                            V                              |
|                      Job Input Queue                      |
|                                                           |
+-----------------------------------------------------------+
```

Inputs to SCHEDULE are:

o The Checkpoint File - contains the dates on which each of the operational jobs last completed

- The CA MICS job control library - contains the JCL for each of the operational jobs

- Today's date - maintained and made available through the operating system

Outputs from SCHEDULE are:

- A tailored job stream containing the jobs that need to be run for the day's processing.

  - The SCHEDULE command submits the generated job stream.

  - The SCHEDULE job writes the job stream to the internal reader (the internal reader will submit the job for execution).

- The Checkpoint File - contains the flags indicating the operational jobs scheduled for today.

- The Schedule Restart File - contains a copy of the job stream that was submitted to the internal reader. You will use this file to manually restart the system if processing abends.

- Updated Operational Status and Tracking control tables (SCHEDULE command) or the Run Status Report containing job status information on each operational job (SCHEDULE job).

PRODUCTION OPERATIONS

There are a number of ways to operate CA MICS in a production mode. If use of a production scheduler, either proprietary or internally developed, is widely accepted at your installation, you should place the CA MICS operational jobs under its control.

If a production scheduler is not available, you can use either the Operational Status and Tracking facility or you can operate CA MICS in a manual mode.

In either case, you have the choice of using the automated facilities CA MICS provides to implement the processing schedule (SCHEDULE command and SCHEDULE job) or manually implementing the processing schedule. Remember that the INCRccc jobs for incremental update processing are scheduled separately and are not supported by the CA MICS automated scheduling facilities.

Refer to the Operational Guidelines section for the steps you will perform to implement the CA MICS processing schedule. The remainder of this section discusses:

- Production Job Scheduler
- Operational Status and Tracking Facility
- Manual Operation
- Reporting

PRODUCTION JOB SCHEDULER

You can use a production job scheduler to either schedule each operational job individually or schedule the CA MICS SCHEDULE job. The approach you choose depends on your installation's production processing standards and procedures. Consult your installation's operations or production control group to find out what they will require to manage CA MICS production processing.

If you enabled incremental update for one or more products in this unit database, then define the individual INCRccc jobs to your installation's production job scheduler. Each INCRccc job will execute one or more times each day to process a subset of the total day's input data according to your predefined schedule of data availability and system workloads.

If you specified AUTOSUBMIT NO in prefix.MICS.PARMS(JCLDEF), you will need to define the CA MICS DAILY, WEEKLY, MONTHLY, YEARLY, and BACKUP operational jobs to your production scheduling system.

- Do not use the SCHEDULE job when you specify AUTOSUBMIT NO because CA MICS will NOT automatically submit DAILYRPT, WEEKRPT, and MONTHRPT to perform daily/weekly/monthly MICF inquiry production processing. CA MICS depends on the AUTOSUBMIT YES option to automatically schedule MICF production processing.

- Likewise, do not use the SCHEDULE job when you specify the ARCHIVE AUDIT, ARCHIVE HISTW, and/or ARCHIVE HISTM prefix.MICS.PARMS(JCLDEF) parameters with the JOB option but without the corresponding AUTOSUBMIT option. CA MICS depends on the JOB AUTOSUBMIT option to automatically schedule stand-alone archive tape processing.

- You will replicate the CA MICS processing schedule in your production control system's definition syntax. Refer to the Processing Schedule section for the scheduling algorithm.

- You must also define DAILYRPT, WEEKRPT, and MONTHRPT to your production scheduling system.

  - Schedule DAILYRPT after DAILY completes successfully.

  - Schedule WEEKRPT after WEEKLY completes successfully.

  - Schedule MONTHRPT after MONTHLY completes successfully.

- You must also define AUDIT, HISTW, and/or HISTM to your production scheduling system if the corresponding ARCHIVE option is active (YES) and you specified the JOB option to perform archive tape processing in a stand-alone batch job.

  - Schedule AUDIT after WEEKLY completes successfully.

  - Schedule HISTW after WEEKLY completes successfully.

  - Schedule HISTM after MONTHLY completes successfully.

If you specified AUTOSUBMIT YES in prefix.MICS.PARMS(JCLDEF), and also specified the AUDIT ARCHIVE, HISTW, and/or HISTM AUTOSUBMIT option if you specified the JOB option for active archive tape processing, then choose one of the following methods to define CA MICS DAILY, WEEKLY, MONTHLY, YEARLY, and BACKUP job processing to the production job scheduling system.

- Define the CA MICS SCHEDULE job to the system

  - This option submits most CA MICS processing via the JES internal reader.

  - The production scheduler should submit the SCHEDULE job each day as soon as practical after CA MICS input data is available and, IF you enabled incremental update for one or more products in the unit database, ALL scheduled INCRccc jobs complete execution.

  - SCHEDULE will automatically submit DAILY, WEEKLY, MONTHLY, YEARLY, and BACKUP processing as required.

   - The DAY400, WEEK400, and MONTH400 steps of the respective DAILY, WEEKLY, and MONTHLY operational jobs will submit the DAILYRPT, WEEKRPT, and MONTHRPT jobs for daily/weekly/monthly reporting.

   - The WEEKLY job will submit the AUDIT and HISTW jobs as required for Archive Audit and Archive Weekly History processing (if active with the JOB option).

   - The MONTHLY job will submit the HISTM job as required for Archive Monthly History processing (if active with the JOB option).

- Define the DAILY, WEEKLY, MONTHLY, YEARLY, and BACKUP jobs to the system.

   - The DAY400, WEEK400, and MONTH400 steps of the respective DAILY, WEEKLY, and MONTHLY operational jobs will submit the DAILYRPT, WEEKRPT, and MONTHRPT jobs through the JES internal reader.

   - The WEEK300 step of the WEEKLY job will submit the AUDIT and HISTW jobs through the JES internal reader as required for Archive Audit and Archive Weekly History processing (if active with the JOB option).

   - The MONTH300 step of the MONTHLY job will submit the HISTM job through the JES internal reader as required for Archive Monthly History processing (if active with the JOB option).

OPERATIONAL STATUS AND TRACKING

The Operational Status and Tracking facility is an online tool for initiating and monitoring CA MICS Database operational processing in the DAILY, WEEKLY, MONTHLY, YEARLY, and BACKUP jobs. It is available to authorized users from the CA MICS Administrator Facility (MAF) option of the CA MICS Workstation Facility (MWF). See the online tutorials and the Operational Status and Tracking section (4.3.4) of this guide for detailed data entry instructions and command descriptions.

NOTE: If you activated incremental update for one or more products in the unit database, you will need to schedule the optional SPLITSMF job and the associated INCRccc jobs manually or using your installation's production batch job scheduling facilities. The online Operational Status and Tracking Facility does NOT support SPLITSMF or INCRccc job submission.

You can use the SCHEDULE command to automatically submit DAILY, WEEKLY, MONTHLY, or YEARLY processing according to the CA MICS processing schedule.

Alternatively, you can use Operational Status and Tracking DAILY, WEEKLY, MONTHLY, and YEARLY commands to "manually" submit DAILY, WEEKLY, MONTHLY, and YEARLY processing. Operational Status and Tracking will prompt you when WEEKLY, MONTHLY, or YEARLY processing is due--Current Operation will be DAILY, WEEKLY, MONTHLY, or YEARLY and status will be DUE TODAY or OVERDUE.

Operational Status and Tracking considers scheduled processing to be late (OVERDUE) if not completed by 8:00 AM You can override the default Overdue time-of-day on the Operational Status and Tracking panel. Scroll RIGHT MAX to display the Overdue option and specify the appropriate deadline for each unit database. Operational Status and Tracking will report status as DUE TODAY until the specified time-of-day.

Reporting considerations:

- o If you specified AUTOSUBMIT YES in prefix.MICS.PARMS(JCLDEF), CA MICS will automatically submit the DAILYRPT, WEEKRPT, and MONTHRPT jobs using the JES internal reader.

- o If you specified AUTOSUBMIT NO in prefix.MICS.PARMS(JCLDEF), you should reconsider your decision to use Operational Status and Tracking to schedule production CA MICS processing because CA MICS will NOT automatically submit DAILYRPT, WEEKRPT, and MONTHRPT to perform daily/weekly/monthly MICF inquiry production processing. You must manually submit, from prefix.MICS.CNTL, the DAILYRPT, WEEKRPT, and MONTHRPT jobs after DAILY/WEEKLY/MONTHLY processing completes.

Archive tape processing considerations:

- o If you specified the YES, JOB, and AUTOSUBMIT options for the ARCHIVE AUDIT, HISTW, and/or HISTM parameters in prefix.MICS.PARMS(JCLDEF), CA MICS will automatically submit the AUDIT, HISTW, and/or HISTM jobs using the JES internal reader. (These jobs offload archive tape processing from the WEEKLY and/or MONTHLY operational jobs to increase database availability. This processing can alternatively be executed within the WEEKLY and/or MONTHLY operational jobs.)

- If you specified the YES and JOB options for the ARCHIVE AUDIT, HISTW, and/or HISTM parameters in prefix.MICS.PARMS(JCLDEF) BUT did NOT include the AUTOSUBMIT option on these parameter statements, you should reconsider your decision to use Operational Status and Tracking to schedule production CA MICS processing because CA MICS will NOT automatically submit AUDIT, HISTW, or HISTM.

MANUAL OPERATION

If you choose not to use a production batch scheduling facility or the CA MICS Operational Status and Tracking facility for production CA MICS processing, then you can manually submit the CA MICS production jobs as required. All CA MICS production jobs are submitted from the prefix.MICS.CNTL data set.

If you enabled incremental update for one or more products in the unit, you will manually submit the SPLITSMF job (if used) and the associated INCRccc jobs as required. The optional SPLITSMF job and each INCRccc job will execute one or more times each day to process a subset of the total day's input data according to your predefined schedule of data availability and system workloads.

You can use the CA MICS SCHEDULE job or you can submit the DAILY, WEEKLY, MONTHLY, and BACKUP operational jobs individually. The approach you choose depends on your installation's production processing standards and procedures. Consult your installation's operations or production control group to find out what they will require to manage CA MICS production processing.

If you specified AUTOSUBMIT YES in prefix.MICS.PARMS(JCLDEF), and also specified the AUDIT ARCHIVE, HISTW, and/or HISTM AUTOSUBMIT option if you specified the JOB option for active archive tape processing, then you can simply submit the CA MICS SCHEDULE job each day as soon as practical after CA MICS input data is available and, IF you enabled incremental update for one or more products in the unit database, ALL scheduled INCRccc jobs complete execution.

- This option submits CA MICS processing via the JES internal reader.

- SCHEDULE will automatically submit DAILY, WEEKLY, MONTHLY, YEARLY, and BACKUP processing as required.

- The DAY400, WEEK400, and MONTH400 steps of the respective DAILY, WEEKLY, and MONTHLY operational jobs will submit the DAILYRPT, WEEKRPT, and MONTHRPT jobs for daily/weekly/monthly reporting.

- The WEEKLY job will submit the AUDIT and HISTW jobs as required for Archive Audit and Archive Weekly History processing (if active with the JOB option).

- The MONTHLY job will submit the HISTM job as required for Archive Monthly History processing (if active with the JOB option).

If you specified AUTOSUBMIT NO in prefix.MICS.PARMS(JCLDEF), you will need to submit each CA MICS operational job individually.

- Do not use the SCHEDULE job when you specify AUTOSUBMIT NO because CA MICS will NOT automatically submit DAILYRPT, WEEKRPT, and MONTHRPT to perform daily/weekly/monthly MICF inquiry production processing. CA MICS depends on the AUTOSUBMIT YES option to automatically schedule MICF production processing.

- Likewise, do not use the SCHEDULE job when you specify the ARCHIVE AUDIT, ARCHIVE HISTW, and/or ARCHIVE HISTM prefix.MICS.PARMS(JCLDEF) parameters with the JOB option but without the corresponding AUTOSUBMIT option. CA MICS depends on the JOB AUTOSUBMIT option to automatically schedule stand-alone archive tape processing.

- You will manually replicate the CA MICS processing schedule. Refer to the Processing Schedule section for the scheduling algorithm.

- You must also submit DAILYRPT, WEEKRPT, and MONTHRPT for MICF production reporting.

    - Submit DAILYRPT after DAILY completes successfully.

    - Submit WEEKRPT after WEEKLY completes successfully.

    - Submit MONTHRPT after MONTHLY completes successfully.

- You must also define AUDIT, HISTW, and/or HISTM to your production scheduling system if the corresponding ARCHIVE option is active (YES) and you specified the JOB option to perform archive tape processing in a stand-alone batch job.

- Schedule AUDIT after WEEKLY completes successfully.

- Schedule HISTW after WEEKLY completes successfully.

- Schedule HISTM after MONTHLY completes successfully.

REPORTING

The DAILY, WEEKLY, and MONTHLY processes include two types of standard CA MICS reporting:

- MICF inquiry production reporting

- Management objective (MBO) and exception reports

MICF inquiry production reporting occurs in the DAILYRPT, WEEKRPT, and MONTHRPT jobs. If you specified AUTOSUBMIT YES in prefix.MICS.PARMS(JCLDEF), then the DAILY, WEEKLY, and MONTHLY jobs automatically submit the corresponding DAILYRPT, WEEKRPT, and MONTHRPT jobs. The DAILYRPT, WEEKRPT, and MONTHRPT jobs also generate management objective (MBO) and exception reports.

If you specified AUTOSUBMIT NO in prefix.MICS.PARMS(JCLDEF), the MBO and exception reports are generated in the DAY400, WEEK400, and MONTH400 steps of the DAILY, WEEKLY, and MONTHLY jobs. If you also schedule the DAILYRPT, WEEKRPT, and MONTHRPT jobs to execute MICF inquiry production reporting, CA MICS will generate duplicate MBO and exception reports. For example, daily MBO and exception reports will be generated from both the DAILY job and the DAILYRPT job.

To solve the potential for duplicate MBO and exception reports:

- If you are NOT using MICF inquiry production reporting AND you specified AUTOSUBMIT NO in the JCLDEF member of prefix.MICS.PARMS, then do NOT schedule the DAILYRPT, WEEKRPT, and MONTHRPT jobs. DAILYRPT, WEEKRPT, and MONTHRPT will simply duplicate the MBO and exception reports generated by the DAILY, WEEKLY, and MONTHLY jobs.

- If you ARE using MICF inquiry production reporting AND you specified AUTOSUBMIT NO in the JCLDEF member of prefix.MICS.PARMS, then use the distributed MICF inquiries for MBO and exception reports. That is, you should turn off MBO and exception reporting in prefix.MICS.PARMS(EXECDEF) and include the MBO and exception report inquiries distributed with CA MICS in your MICF production reporting definitions.

See section 2.3.5 of this guide for details on EXECDEF parameters. See the Standard Reports Guide for more information on MBO and exception reports. See MICF Administration in the MICF Reference Guide for more information on inquiry production reporting.

ARCHIVE TAPE PROCESSING

The CA MICS Archive Database facility provides optional long-term retention for both detail level (audit) data and summarized (history) data. Archive Audit processing normally occurs weekly and provides long-term retention of DETAIL and/or DAYS level data to meet audit and/or detail analysis requirements. Archive History processing optionally occurs weekly and/or monthly and provides long-term retention of WEEKS and/or MONTHS level (summarized) data to address long-term reporting, capacity planning, and historical analysis requirements.

Archive tape processing can occur within the WEEKLY and/or MONTHLY operational jobs, or you can optionally offload this processing to the stand-alone AUDIT, HISTW, and HISTM jobs which access the database DISP=SHR, thereby freeing the database for concurrent execution of reporting jobs and/or online end-user inquiry and analysis, or any other "read-only" processing. See Section 2.3.3.2.1.3 for more information on parameters controlling archive tape processing.

4.3.3 Operational Processes, Jobs, and Steps

An operational process is a series of batch jobs that make up a logical unit of work. Operational processes update and maintain the CA MICS Database.

Operational jobs are the smallest executable unit of work in CA MICS. An operational job updates the CA MICS Database, generates reports, or performs a restart/recovery operation.

Job steps are a specific item of work within an operational job. A job step updates the CA MICS Database for a single product, generates reports, or performs a control function.

The following sections provide more details about:

1 - Operational Processes
2 - Operational Jobs
3 - Job Steps

4.3.3.1 Operational Processes

An operational process is a series of batch jobs that make up a logical unit of work. Operational processes update and maintain the CA MICS Database.

There are four operational processes in CA MICS:

```
o  DAILY process   - the DAILY job followed by the BACKUP
                     job,

                     or

                     one or more SPLITSMF (optional) and
                     INCRccc jobs followed by the DAILY and
                     BACKUP jobs (where ccc is a product ID).

o  WEEKLY process  - the DAILY job followed by the WEEKLY job
                     followed by the BACKUP job

o  MONTHLY process - the  DAILY  job  followed by the MONTHLY
                     job followed by the BACKUP job

o  YEARLY process  - the DAILY job followed by the YEARLY job
                     followed by the BACKUP job
```

Incremental updates supplement the DAILY job and are considered a part of the DAILY process. Each INCRccc job performs a portion of the ccc product's work that would otherwise occur in the DAILY job.

4.3.3.2 Operational Jobs

CA MICS provides standard operational jobs for updating, reporting, maintaining, and recovering its Database:

o DAILY - run each day to update the DETAIL and DAYS timespan files plus week-to-date and month-to-date files in the WEEKS and MONTHS timespans (may be supplemented by one or more INCRccc job executions)

o WEEKLY - run once each week after the DAILY job for WEEKS timespan cycle close-out, weekly archive audit, and weekly archive history processing

o MONTHLY - run once each month after the DAILY job for MONTHS timespan cycle close-out and monthly archive history processing and for updating year-to-date files

o YEARLY - run once each year after the MONTHLY job for YEARS timespan cycle close-out

o BACKUP - run daily, bi-daily, or weekly (per user-specified backup frequency) to generate a tape backup of the entire Database

o SCHEDULE - run each day to submit scheduled processing

o RESTORE - run whenever the database is damaged or must be recovered

o DAILYRPT - run after DAILY to produce daily production reports

o WEEKRPT - run after WEEKLY to produce weekly production reports

o MONTHRPT - run after MONTHLY to produce monthly production reports

o RSTRTBLS - run to restore the TABLES and SCREENS data sets

o RSTRTLIB - run to restore ISPF-based information

o DAYSMFR - run as necessary to recreate DAILY job work files

o ACTDAY1R - run as necessary to restore the CA MICS

```
                 Accounting and Chargeback DAY1 audit file

   o  RSTATUS  - run as necessary to update Operational Status
                 and Tracking control tables or to replace a
                 lost Run Status Report

   o  INCRccc  - optionally run one or more times per day to
                 process a portion of the daily input data for
                 the ccc data integration product to update the
                 product's incremental update ("today to-date")
                 DETAIL and DAYS timespan files

   o  SPLITSMF - (optional) run one or more times a day to split
                 the SMF input data into multiple files for
                 processing in the INCRccc jobs.  SPLITSMF is a
                 stand-alone version of the DAILY job's DAYSMF
                 step and applies only to those products which
                 take their input from the SMF files, and which
                 are marked as INCRUPDATE YES and INCRSPLIT USE
                 in prefix.MICS.PARMS(cccOPS).

   o  IUDBINIT - optionally run to re-initialize in-progress
                 incremental update processing after restoring
                 the unit database files (i.e., after running
                 the RESTORE job).  Messages in the RESTORE job
                 MICSLOG will prompt you to execute IUDBINIT
                 when needed.

   o  AUDIT    - (optional) run after WEEKLY to perform optional
                 Archive Audit processing.  The AUDIT job is
                 used when you specify

                      ARCHIVE AUDIT YES JOB

                 in prefix.MICS.PARMS(JCLDEF).

                 NOTE:  You can execute AUDIT more frequently
                        (e.g., twice a week or daily) if DASD
                        space is inadequate for retaining
                        sufficient DETAIL/DAYS cycles for weekly
                        audit tape creation.

                        In this case, it is necessary to add the
                        following to prefix.MICS.PARMS(EXECDEF)

                             USERDEF AUDITCWK YES

                        This parameter overrides the default
                        audit archive processing.  Data for the
```

```
                     current week is retained and copied to
                     the new audit archive tape cycle.

o  HISTW    - (optional) run after WEEKLY to perform optional
              Archive Weekly History processing.  The HISTW
              job is used when you specify

                   ARCHIVE HISTW YES JOB

              in prefix.MICS.PARMS(JCLDEF).

o  HISTM    - (optional) run after MONTHLY to perform
              optional Archive Monthly History processing.
              The HISTM job is used when you specify

                   ARCHIVE HISTM YES JOB

              in prefix.MICS.PARMS(JCLDEF).
```

The operational jobs are documented in the sections that follow.

4.3.3.2.1 DAILY Operational Job

```
+-----------------------------------------------------------+
|                                                           |
|                  Input Data Sources                       |
|                                                           |
|     |      |      |      |       |       |       |        |
|     |      |      |      |       |       |       |        |
|     +------+------+----->|<------+-------+-------+        |
|                          |                                |
|                          V                                |
|   Checkpoint          +-------+           Operational     |
|      File<---------->| DAILY |---------->Status and       |
|                      +-------+           Tracking         |
|                          |                                |
|                          |                                |
|      +--------------+---------------+--------------+      |
|      |              |               |              |      |
|      V              V               V              V      |
|  Exception      Management      MICF Rpts.      Online    |
|  Reports        Objective       & Graphics      Database  |
|                 Reports                                   |
|                                                           |
+-----------------------------------------------------------+
```

Run the DAILY job each day.

The JCL for the DAILY job is in prefix.MICS.CNTL(DAILY). It is tailored to the components in the unit database.

The DAILY job operates as follows:

o If incremental update is active for one or more products, verify that no INCRccc jobs are still running or awaiting restart/recovery.

o If stand-alone Archive Audit processing is enabled, verify that previously submitted AUDIT jobs completed successfully.

o Select input data for installed products.

o Create information area files in the DETAIL and DAYS timespans.

  Note: If incremental update is enabled for one or more products, some DAILY job steps will,

  - perform INCRccc processing (see section 4.3.3.2.16)

  - roll-up the incremental update DETAIL and DAYS timespan files to the CA MICS database DETAIL and DAYS timespan data sets

  - initialize the incremental update checkpoint and database files in preparation for the next day's INCRccc processing

o Update week-to-date and month-to-date files in the WEEKS and MONTHS timespans, respectively.

o Perform daily accounting processing if CA MICS Accounting and Chargeback is installed in the unit database.

o If prefix.MICS.PARMS(JCLDEF) specifies AUTOSUBMIT YES, copy prefix.MICS.CNTL(DAILYRPT) to the internal reader (submit the DAILYRPT job).

- o If prefix.MICS.PARMS(JCLDEF) specifies AUTOSUBMIT NO, produce management objective (MBO) and exception reports if the products in the unit support these reports -- see prefix.MICS.PARMS(EXECDEF).

- o Optionally execute user (non-CA MICS) reporting.

- o Update Operational Status and Tracking control tables.

DATABASE UPDATES

A new file cycle (level 01) is added to the Database for each of the files maintained in the DETAIL and DAYS timespans. This 01 level cycle contains the data processed from the input data sets. The files that are maintained in the WEEKS and MONTHS timespans are updated on a week-to-date and month-to-date basis, respectively.

MICF INQUIRY PRODUCTION REPORTS AND GRAPHICS

Daily MICF inquiry production interface processing in the DAILYRPT job executes shared MICF inquiries. Printed reports and color graphics are saved in the associated MICF production inquiry output catalog(s) for online review and printing using the MICF Reports and Graphics option. See MICF Administration in the MICF Reference Guide for more information on MICF inquiry production reporting.

STANDARD REPORTS

If prefix.MICS.PARMS(EXECDEF) exceptions and MBO parameters are set to valid values other than NONE, the DAILY job produces the following reports for the previous day's activity.

- o Daily Management Objective (MBO) Reports by Information Area

- o Daily Exception Reports

Refer to section 2.3.5 for more information on EXECDEF.

MAINTENANCE REPORTS

The DAILY job generates JES, MICSLOG, and SAS log output documenting performance of the daily update.

- The Run Status Report describes the result of current processing. This information is also available with the Operational Status and Tracking STATUS command.

- The MICSLOG reports the status of each production job step. Database update steps report the number of input records processed and log the records deleted due to date-range, record length, and other validation tests. This information is also available with the Operational Status and Tracking HISTORY command.

DEPENDENCIES

If the DAILY job malfunctions, no other update functions will be able to execute. The DAILY job must successfully complete before other functions can start.

The DAILY job will not be able to execute if any of the latest DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP or RESTORE jobs have completed unsuccessfully.

If incremental update is enabled for one or more products, then the DAILY job will not be able to execute if any of the latest INCRccc jobs have completed unsuccessfully.

If the DAILY job is accidentally submitted for execution twice in the same day (with the same data in the input files), the second run abends (U300) because the duplicate data check mechanisms in CA MICS recognize that the same data is being submitted again.

4.3.3.2.2 WEEKLY Operational Job

```
+-----------------------------------------------------------+
|                                                           |
|                       Online                              |
|                      Database                             |
|                          |                                |
|                          |                                |
|                          V                                |
|   Checkpoint        +---------+          Operational      |
|      File<--------->| WEEKLY  |--------->Status and       |
|                     +---------+          Tracking         |
|                          |                                |
|                          |                                |
|     +-------------+-----------+----------+----------+     |
|     |             |           |          |          |     |
|     V             V           V          V          V     |
| Management    MICF Rpts.    Online    Archive    Archive  |
| Objective     & Graphics   Database   Audit      Weeks    |
| Reports                               Data       History  |
|                                                  Data     |
+-----------------------------------------------------------+
```

The WEEKLY job runs at the start of each new week after the DAILY for the last day of the week completes successfully. The standard CA MICS schedule algorithm delays weekly processing for 2 days after the end of the week. This execution pattern provides the reports for the previous week's activity early in the new week (e.g., on Monday) and has the database fully updated for adhoc reporting and problem tracking, while still allowing for suspended input data (e.g., data for batch jobs that have not yet printed).

The JCL for the WEEKLY job is in prefix.MICS.CNTL(WEEKLY). It is tailored to the components in the unit.

The WEEKLY job operates as follows:

o If stand-alone Archive Audit processing is enabled, verify that the previous week's AUDIT job completed successfully.

o If stand-alone Archive Weekly History processing is enabled, verify that the previous week's HISTW job completed successfully.

o Update the online Database by aging the in-process week-to-date cycles to full week 01 cycles.

o Submit the AUDIT job, or create the weekly archive audit Database.

o Submit the HISTW job, or update the weeks archive history Database.

o If prefix.MICS.PARMS(JCLDEF) specifies AUTOSUBMIT YES, copy prefix.MICS.CNTL(WEEKRPT) to the internal reader (submit the WEEKRPT job).

o If prefix.MICS.PARMS(JCLDEF) specifies AUTOSUBMIT NO, produce management objective (MBO) reports if the products in the unit support these reports -- see prefix.MICS.PARMS(EXECDEF).

o Optionally execute user (non-CA MICS) reporting.

o Update Operational Status and Tracking control tables.

DATABASE UPDATES

The week-to-date files are updated each day by the DAILY job. The WEEKLY job extracts prior weeks data from the week-to-date cycles to become the first level WEEKS cycles (level 01). Data from the current week remains in the week-to-date cycles as the basis for a new week-to-date accumulation.

For each file maintained in the WEEKS timespan, one cycle is created for each week's data processed.

ARCHIVE AUDIT DATA

Archive audit data is created as a new entry in the audit generation data group. The audit data is generated for files that contain job, program, command, or application unit execution-related data by copying data from the last ten DAILY or DETAIL timespan cycles maintained online for the applicable files. If the file is active in the DETAIL timespan, DETAIL is used. If the file is NOT active in the DETAIL timespan, then DAYS is used.

- If you specified

  ```
  ARCHIVE AUDIT YES JOB
  ```

  in prefix.MICS.CNTL(JCLDEF), then Archive Audit processing is performed in the stand-alone AUDIT job. If you specified the AUTOSUBMIT parameter on the ARCHIVE statement, then the WEEKLY job WEEK300 step automatically submits the AUDIT job through the OS/390 internal reader. If you specified NOAUTOSUBMIT (or took the default) then you must schedule the AUDIT job externally, either manually or through your installation's production batch job scheduling facilities.

  NOTE: The AUDIT job must complete before the next scheduled DAILY job will be allowed to execute.

- If you specified

  ```
  ARCHIVE AUDIT YES
  ```

  in prefix.MICS.CNTL(JCLDEF), without the JOB parameter (i.e., you took the default or specified STEP), then Archive Audit processing is performed in the WEEK300 step of the WEEKLY job.

The files included in archive audit processing are specified in prefix.MICS.PARMS(DBMODEL) and sharedprefix.MICS.GENLIB member cccGENIN, where ccc is the three-character component identifier. Archive audit tape retention (number of generations) is specified in prefix.MICS.PARMS(JCLDEF). See the CA MICS Structure and Content Guide for more information on the Archive Database.

ARCHIVE HISTORY DATA

The archive weekly history data is updated by creating a new generation data group entry using the most recent WEEKS timespan data (cycle 01) and the history-to-date contained in the current (0) generation archive for each file maintained in the WEEKS timespan archives. Historical data and new data are read in and any data that is older than the user-specified archive retention age is deleted.

- If you specified

```
     ARCHIVE HISTW YES JOB
```

  in prefix.MICS.CNTL(JCLDEF), then Archive Weekly History processing is performed in the stand-alone HISTW job. If you specified the AUTOSUBMIT parameter on the ARCHIVE statement, then the WEEKLY job WEEK300 step automatically submits the HISTW job through the OS/390 internal reader. If you specified NOAUTOSUBMIT (or took the default) then you must schedule the HISTW job externally, either manually or through your installation's production batch job scheduling facilities.

  NOTE: The HISTW job must complete before the next scheduled WEEKLY job will be allowed to execute.
- If you specified

```
     ARCHIVE HISTW YES
```

  in prefix.MICS.CNTL(JCLDEF), without the JOB parameter (i.e., you took the default or specified STEP), then Archive Weekly History processing is performed in the WEEK300 step of the WEEKLY job.

The WEEKS timespan files included in weekly archive history processing and the archive retention are specified in prefix.MICS.PARMS(DBMODEL) and sharedprefix.MICS.GENLIB member cccGENIN, where ccc is the three-character component identifier. See the CA MICS Structure and Content Guide for more information on the Archive Database.

MICF INQUIRY PRODUCTION REPORTS AND GRAPHICS

Weekly MICF inquiry production interface processing in the WEEKRPT job executes shared MICF inquiries. Printed reports and color graphics are saved in the associated MICF production inquiry output catalog(s) for online review and printing using the MICF Reports and Graphics option. See MICF Administration in the MICF Reference Guide for more information on MICF inquiry production reporting.

STANDARD REPORTS

Based on the REPORT MBOWEEKLY parameter in prefix.MICS.PARMS member EXECDEF, the WEEKLY job produces the following reports for the previous week's activity.

o Weekly Management Objective (MBO) Reports by Information Area

MAINTENANCE REPORTS

The WEEKLY job generates JES, MICSLOG, and SAS log output documenting performance of the weekly update.

o The Run Status Report describes the result of current processing. This information is also available with the Operational Status and Tracking STATUS command.

o The MICSLOG reports the status of each production job step.

DEPENDENCIES

If the WEEKLY job malfunctions, no other update functions will be able to execute. The WEEKLY job must successfully complete before other functions can start.

The WEEKLY job will not be able to execute if any of the latest DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP or RESTORE jobs have completed unsuccessfully.

4.3.3.2.3 MONTHLY Operational Job

```
+-----------------------------------------------------------+
|                                                           |
|                       Online                              |
|                      Database                             |
|                          |                                |
|                          |                                |
|                          V                                |
|   Checkpoint        +---------+          Operational      |
|      File<--------->| MONTHLY |--------->Status and       |
|                     +---------+          Tracking         |
|                          |                                |
|                          |                                |
|       +--------------+--------------+-------------+       |
|       |              |              |             |       |
|       V              V              V             V       |
|   Management     MICF Rpts.       Online       Archive    |
|   Objective      & Graphics      Database      Monthly    |
|   Reports                                      History    |
|                                                 Data      |
+-----------------------------------------------------------+
```

The MONTHLY job runs at the start of each new month after the DAILY for the last day of the month completes successfully. The standard CA MICS schedule algorithm delays monthly processing for 3 days after the end of the month. This execution pattern provides the reports for the previous month's activity early in the new month (e.g., on the 3rd) and has the database fully updated for adhoc reporting, while still allowing for suspended input data (e.g., data for batch jobs that have not yet printed).

The JCL for the MONTHLY job is in prefix.MICS.CNTL(MONTHLY). The JCL is tailored to the components in the unit.

The MONTHLY job operates as follows:

o If stand-alone Archive Monthly History processing is enabled, verify that the previous month's HISTM job completed successfully.

o Update the online database by aging the in-process month-to-date cycles to full month 01 cycles.

o Optionally update the year-to-date files in the YEARS timespan from the new month 01 cycles.

o Update the months archive history database.

- o If prefix.MICS.PARMS(JCLDEF) specifies AUTOSUBMIT YES, copy prefix.MICS.CNTL(MONTHRPT) to the internal reader (submit the MONTHRPT job).

- o If prefix.MICS.PARMS(JCLDEF) specifies AUTOSUBMIT NO, produce management objective (MBO) reports if the products in the unit support these reports -- see prefix.MICS.SOURCE(EXECDEF).

- o Optionally execute user (non-CA MICS) reporting.

- o Update Operational Status and Tracking control tables.

- o If prefix.MICS.PARMS(JCLDEF) specifies AUTOSUBMIT YES, copy the monthly backup job to the internal reader (submit the monthly backup).

- o If prefix.MICS.PARMS(JCLDEF) specifies AUTOSUBMIT NO, perform monthly backup processing.

DATABASE UPDATES

The month-to-date files are updated each day by the DAILY job. The MONTHLY job extracts prior months data from the month-to-date cycles to become the first level MONTHS cycles (level 01). Data from the current month remains in the month-to-date cycles as the basis for a new month-to-date accumulation.

For each file maintained in the MONTHS timespan, one cycle is created for each month's data processed.

If the YEARS timespan is active, year-to-date files in the YEARS timespan are updated using the newly-created 01 MONTHS timespan cycles. See the YEARS TIMESPAN ACTIVE/INACTIVE parameter in sharedprefix.MICS.PARMS(CPLXDEF) and prefix.MICS.PARMS(SITE) to determine if the YEARS timespan is active or inactive.

ARCHIVE HISTORY DATA

Archive monthly history data is updated by creating a new generation data group entry using the most recent MONTHS timespan data (cycle 01) and the history-to-date contained in the current (0) generation archive for each file maintained in the MONTHS timespan archives. The historical and new data are read in and any data that is older than the user-specified archive retention age is deleted.

o If you specified

```
        ARCHIVE HISTM YES JOB
```

   in prefix.MICS.CNTL(JCLDEF), then Archive Monthly History processing is performed in the stand-alone HISTM job. If you specified the AUTOSUBMIT parameter on the ARCHIVE statement, then the MONTHLY job MONTH300 step automatically submits the HISTM job through the OS/390 internal reader. If you specified NOAUTOSUBMIT (or took the default) then you must schedule the HISTM job externally, either manually or through your installation's production batch job scheduling facilities.

   NOTE: The HISTM job must complete before the next scheduled MONTHLY job will be allowed to execute.

o If you specified

```
        ARCHIVE HISTM YES
```

   in prefix.MICS.CNTL(JCLDEF), without the JOB parameter (i.e., you took the default or specified STEP), then Archive Monthly History processing is performed in the MONTH300 step of the MONTHLY job.

The MONTHS timespan files included in monthly archive history processing and the archive retention are specified in prefix.MICS.PARMS(DBMODEL) and sharedprefix.MICS.GENLIB member cccGENIN, where ccc is the three-character component identifier.

MICF INQUIRY PRODUCTION REPORTS AND GRAPHICS

Monthly MICF inquiry production interface processing in the MONTHLY job executes shared MICF inquiries. Printed reports and color graphics are saved in the associated MICF production inquiry output catalog(s) for online review and printing using the MICF Reports and Graphics option. See MICF Administration in the MICF Reference Guide for more information on MICF inquiry production reporting.

STANDARD REPORTS

Based on the REPORT MBOMONTHLY parameter in prefix.MICS.PARMS member EXECDEF, the MONTHLY job produces the following reports for the previous month's activity.

o Monthly Management Objective (MBO) Reports by Information Area

MAINTENANCE REPORTS

The MONTHLY job generates JES, MICSLOG, and SAS log output documenting performance of the monthly update.

o The Run Status Report describes the result of current processing. This information is also available with the Operational Status and Tracking STATUS command.

o The MICSLOG reports the status of each production job step.

MONTHLY BACKUP

A MONTHLY backup of the database may be taken. The monthly backup provides recovery capability to the month level.

o If you specified

```
     MONTHLY BACKUP YES
```

or allowed this to default, the MONTHLY job will include step BKUP900M which submits the monthly backup job in member prefix.MICS.CNTL(BACKUPM).

o If you specified

```
     MONTHLY BACKUP NO
```

the MONTHLY job will not include the BKUP900M step. In this case, you must submit the BACKUPM job by some other means in order to keep a monthly backup of your database.

DEPENDENCIES

If the MONTHLY job malfunctions, no other update functions will be able to execute. The MONTHLY job must successfully complete before other functions can start.

The MONTHLY job will not be able to execute if any of the

```
latest DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, or RESTORE
jobs have completed unsuccessfully.
```

4.3.3.2.4 YEARLY Operational Job

```
+-----------------------------------------------------------+
|                                                           |
|                       Online                              |
|                      Database                             |
|                          |                                |
|                          |                                |
|                          V                                |
|   Checkpoint        +---------+          Operational      |
|      File<--------->| YEARLY  |--------->Status and       |
|                     +---------+          Tracking         |
|                          |                                |
|                          |                                |
|                          |                                |
|                          V                                |
|                       Online                              |
|                      Database                             |
|                                                           |
+-----------------------------------------------------------+
```

The YEARLY job runs at the start of each new year after the MONTHLY for the last month of the year completes successfully. The standard CA MICS schedule algorithm delays yearly processing for 5 days after the end of the year. This execution pattern has the Database fully updated for adhoc reporting early in the new year (e.g., January 5th) while still allowing for suspended input data (e.g., data for batch jobs that have not yet printed). Refer to Section 4.3.12.4 for special year-end processing considerations.

The JCL for the YEARLY job is in prefix.MICS.CNTL(YEARLY). It is tailored to the components in the unit.

The YEARLY job operates as follows:

- o Update the Database by aging the in-process year-to-date cycles to full year cycles (01 level).

- o Update Operational Status and Tracking control tables.

DATABASE UPDATES

The year-to-date files are updated each month by the MONTHLY job. The YEARLY job extracts prior years data from the year-to-date cycles to become the first level YEARS cycles (level 01). Data from the current year remains in the year-to-date cycles as the basis for a new year-to-date accumulation.

For each file maintained in the YEARS timespan, one cycle is created for each year's data processed.

MAINTENANCE REPORTS

The YEARLY job generates JES, MICSLOG, and SAS log output documenting performance of the weekly update.

- o The Run Status Report describes the result of current processing. This information is also available with the Operational Status and Tracking STATUS command.

- o The MICSLOG reports the status of each production job step.

DEPENDENCIES

If the YEARLY job malfunctions, no other update functions will be able to execute. The YEARLY job must successfully complete before other functions can start.

The YEARLY job will not be able to execute if any of the latest DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, and RESTORE jobs have been completed unsuccessfully.

4.3.3.2.5 BACKUP Operational Job

```
+-----------------------------------------------------------+
|                                                           |
|                                  Checkpoint               |
|            Online                  File                   |
|           Database                   |                    |
|               |                      |                    |
|               +----------+-----------+                    |
|                          |                                |
|                          |                                |
|                          V                                |
|   Checkpoint        +---------+         Operational      |
|      File<--------->| BACKUP  |--------->Status and       |
|                     +---------+         Tracking         |
|                          |                                |
|                          |                                |
|                          |                                |
|                          V                                |
|                       Backup                              |
|                        Data                               |
|                        Files                              |
|                                                           |
+-----------------------------------------------------------+
```

The BACKUP job runs after the last update job completes successfully. BACKUP frequency (daily, bi-daily, or weekly) is specified in prefix.MICS.PARMS(JCLDEF).

The JCL for the BACKUP job is in prefix.MICS.CNTL(BACKUP). It is tailored to the components in the unit.

The BACKUP job creates a copy of the Database in six generation data groups, named:

```
        tapeprefix.MICS.BACKUP.CHECKPT
        tapeprefix.MICS.BACKUP.DETAIL
        tapeprefix.MICS.BACKUP.DAYS
        tapeprefix.MICS.BACKUP.WEEKS
        tapeprefix.MICS.BACKUP.MONTHS
        tapeprefix.MICS.BACKUP.YEARS
```

The BACKUP job for the PRIMARY unit database also backs up:

```
        sharedprefix.MICS.SCREENS
        sharedprefix.MICS.TABLES
        sharedprefix.MICS.ISPTLIB
```

If the CA MICS Capacity Planner is installed in the PRIMARY unit, the PRIMARY unit BACKUP job also backs up:

```
        sharedprefix.MICS.CAPACITY
```

The BACKUP job tracks compressed data on an inventory so that compressed data can be restored in compressed mode. (The SAS System is not directed to write compressed data to tape in compressed mode.)

DEPENDENCIES

If the BACKUP job malfunctions, no other update functions will be able to execute. The BACKUP job must successfully complete before other functions can start.

The BACKUP job will not be able to execute if any of the latest DAILY, WEEKLY, MONTHLY, YEARLY, and RESTORE jobs have been completed unsuccessfully. The BACKUP job will be able to execute if a prior BACKUP job failed -- that is, you can restart the BACKUP job by rerunning the BACKUP job.

4.3.3.2.6 SCHEDULE Operational Job

```
+-----------------------------------------------------------+
|                                                           |
|        MICS Job Control Library (prefix.MICS.CNTL)        |
|                                                           |
|                          DAILY                            |
|                          BACKUP                           |
|                          WEEKLY                           |
|         Checkpoint       MONTHLY      Today's Date        |
|            File          YEARLY        (From MVS)         |
|             |              |               |              |
|             |              |               |              |
|             V______________V_______________V              |
|                            |                              |
|                            V                              |
|                      +----------+       +--------+       |
|                      |          |       |  Run   |       |
|                      | SCHEDULE |------>| Status |       |
|                      |          |       | Report |       |
|                      +----------+       +--------+       |
|                            |          (only produced on   |
|                            V          SCHEDULE abnormal   |
|                          Today's         termination)     |
|                          Tailored                         |
|                         Job Stream                        |
|                            |                              |
|        +------------------+------------------+           |
|        |                  |                  |           |
|        V                  V                  V           |
|    Checkpoint           System           Schedule        |
|       File             Internal          Restart         |
|                          Reader           File           |
|                            |                              |
|                            |                              |
|                            |                              |
|                            V                              |
|                     Job Input Queue                       |
|                                                           |
+-----------------------------------------------------------+
```

The SCHEDULE operational job is one of two tools for implementing the CA MICS operational schedule. You can use the Operational Status and Tracking SCHEDULE command or the batch SCHEDULE job to automatically submit the proper DAILY, WEEKLY, MONTHLY, YEARLY, and/or BACKUP processing due today.

The SCHEDULE job examines the Checkpoint File and determines the CA MICS Database update processing that should occur today. SCHEDULE then builds a composite batch job stream from the DAILY, WEEKLY, MONTHLY, YEARLY, and/or BACKUP job JCL in prefix.MICS.CNTL. The composite job is copied to the JES internal reader and also saved on the prefix.MICS.RESTART.CNTL data set.

See the Processing Schedule and Implementing the Processing Schedule sections (4.3.1 and 4.3.2) of this guide for more information.

4.3.3.2.7 RESTORE Operational Job

```
+-----------------------------------------------------------+
|                                                           |
|                       Backup                              |
|                        Data                               |
|                        Files                              |
|                          |                                |
|                          |                                |
|                          |                                |
|                          V                                |
|   Checkpoint        +---------+          Operational      |
|      File<--------->| RESTORE |--------->Status and       |
|                     +---------+          Tracking         |
|                          |                                |
|                          |                                |
|              +----------+-----------+                     |
|              |                      |                     |
|              V                      V                     |
|           Online                Checkpoint                |
|          Database                  File                   |
|                                                           |
+-----------------------------------------------------------+
```

Run the RESTORE job to recover the Database.

The JCL for the RESTORE job is in prefix.MICS.CNTL(RESTORE). It is tailored to the components in the unit, and if compression was active for the files when they were compressed, the data is restored in compressed format.

The RESTORE job defaults to the current (0) generation backup. You can also restore from another daily backup generation or from a monthly backup.

Recovery to the day level (from a standard daily backup generation) places the CA MICS Database in the status it was in when the specified Daily backup was executed.

Recovery to the month level (from a monthly backup generation) protects the CA MICS Database from the vulnerability of bad data being processed for a period of time greater than the number of generations of daily backup tapes. The monthly backup also provides an alternative to restoring from a daily backup prior to the last monthly and having to rerun a monthly process.

If you have activated incremental update for any product in the unit database, the RESTORE job may issue messages warning that you may need to re-initialize the incremental update database files. If one or more INCRccc jobs completed prior to running the RESTORE job, then the incremental update database files contain information based on the unit database state prior to the RESTORE. Since the unit database contents have changed, the "today to-date" files in the incremental update database are invalid. Execute the IUDBINIT operational job to re-initialize the incremental update database files and position your unit database for correctly recovering the data processed since the last backup was taken.

If CA MICS Accounting and Chargeback is installed in the unit database, you will need to run the ACTDAY1R job after completing RESTORE processing. See the CA MICS Accounting and Chargeback User Guide for special considerations relating to accounting files.

After RESTORE completes, and after you have run the IUDBINIT and/or ACTDAY1R jobs (if required), rerun ALL database update processing performed since the backup was taken.

RESTORING THE DATABASE

Use the Operational Status and Tracking RESTORE command. You can select the current daily backup generation, another daily backup generation, or a monthly backup.

You can manually restore the Database using the job in prefix.MICS.CNTL(RESTORE). RESTORE defaults to the current (0) daily backup generation. You can restore from another backup generation or monthly backup by changing RESTORE job symbolic parameters. DO NOT SAVE THE MODIFIED RESTORE JCL. The RESTORE parameters are:

```
  o DSNPREF - Backup prefix.  Defaults to daily backup DSN.

  o GDG     - Backup cycle.  Defaults to 0 (current).
```

The RESTORE job prints a copy of the checkpoint file for the online database and the checkpoint file for the tape backup database before beginning the restore process. This serves two purposes. First, it allows comparison of the state of the database at the time it was damaged with the backup version used in the RESTORE. Second, it ensures that the backup tape does indeed exist and that at least one file of it can be read. If the backup checkpoint file cannot be printed, the RESTORE job will stop and the damaged database will not be deleted.

If you specified RESTORE BACKUP YES in prefix.MICS.PARMS(JCLDEF), the RESTORE job attempts to execute a database BACKUP before the RESTORE. If the RESTORE job fails because the BACKUP is unsuccessful, the RESTORE can be completed by

- o replying NO to the Operational Status and Tracking RESTORE command pre-restore backup prompt

  or

- o manually restarting the RESTORE job in the RSTR900 step

The data sets produced by the pre-restore backup are written to the same generation data groups that are used for normal backups, but are not catalogued. This means that if the RESTORE job fails, you must ensure that the tapes are kept and catalogued before they are scratched. If the restore is successful, then you may elect to let the tapes be scratched so that they will not affect the normal sequence of your CA MICS backup generations.

DEPENDENCIES

If the RESTORE job malfunctions, no other update functions will be able to execute. The RESTORE job must successfully complete before other functions can start.

The RESTORE job is allowed to execute even if any of the latest DAILY, WEEKLY, MONTHLY, YEARLY, or BACKUP jobs failed. In this case, recognize that any processing that may have taken place since the date of the backup that is being restored will have to be repeated again.

Remember that the RESTORE job does not touch the incremental update database files. If any INCRccc jobs have run in this unit for the current day's data, you will need to run the IUDBINIT job to re-initialize the incremental update database files.

4.3.3.2.8 DAILYRPT Operational Job

```
+-----------------------------------------------------------+
|                                                           |
|                       Online                              |
|                      Database                             |
|                          |                                |
|                          |                                |
|                          V                                |
|                     +----------+                          |
|                     | DAILYRPT |                          |
|                     +----------+                          |
|                          |                                |
|                          |                                |
|           +--------------+---------------+                |
|           |              |               |                |
|           V              V               V                |
|       Exception      Management      MICF Reports         |
|       Reports        Objective        & Graphics          |
|                      Reports                              |
|                                                           |
+-----------------------------------------------------------+
```

DAILYRPT runs after the DAILY job.

The JCL for the DAILYRPT job is in prefix.MICS.CNTL(DAILYRPT).

The DAILYRPT job operates as follows:

- o Execute daily MICF inquiry production reporting for this unit database.

- o Produce management objective (MBO) and exception reports if the products in the unit support these reports -- see prefix.MICS.PARMS(EXECDEF).

MICF INQUIRY PRODUCTION REPORTS AND GRAPHICS

Daily MICF inquiry production interface processing in the DAILYRPT job executes shared MICF inquiries. Printed reports and color graphics are saved in the associated MICF production inquiry output catalog(s) for online review and printing using the MICF Reports and Graphics option. See MICF Administration in the MICF Reference Guide for more information on MICF inquiry production reporting.

STANDARD REPORTS

If prefix.MICS.PARMS(EXECDEF) exceptions and MBO parameters are set to valid values other than NONE, the DAILYRPT job produces the following reports for the previous day's activity.

- Daily Management Objective (MBO) Reports by Information Area

- Daily Exception Reports

4.3.3.2.9 WEEKRPT Operational Job

```
+-----------------------------------------------------------+
|                                                           |
|                       Online                              |
|                      Database                             |
|                          |                                |
|                          |                                |
|                          V                                |
|                     +----------+                          |
|                     | WEEKRPT  |                          |
|                     +----------+                          |
|                          |                                |
|                          |                                |
|                  +-------+-------+                        |
|                  |               |                        |
|                  V               V                        |
|              Management      MICF Reports                 |
|              Objective        & Graphics                  |
|              Reports                                      |
|                                                           |
+-----------------------------------------------------------+
```

WEEKRPT runs after the WEEKLY job.

The JCL for the WEEKRPT job is in prefix.MICS.CNTL(WEEKRPT).

The WEEKRPT job operates as follows:

o Execute weekly MICF inquiry production reporting for this unit database.

o Produce management objective (MBO) reports if the products in the unit support these reports -- see prefix.MICS.PARMS(EXECDEF).

MICF INQUIRY PRODUCTION REPORTS AND GRAPHICS

Weekly MICF inquiry production interface processing in the WEEKRPT job executes shared MICF inquiries. Printed reports and color graphics are saved in the associated MICF production inquiry output catalog(s) for online review and printing using the MICF Reports and Graphics option. See MICF Administration in the MICF Reference Guide for more information on MICF inquiry production reporting.

STANDARD REPORTS

Based on the REPORT MBOWEEKLY parameter in prefix.MICS.PARMS member EXECDEF, the WEEKRPT job produces the following reports for the previous week's activity.

- Weekly Management Objective (MBO) Reports by Information Area

4.3.3.2.10 MONTHRPT Operational Job

```
+-----------------------------------------------------------+
|                                                           |
|                       Online                              |
|                      Database                             |
|                          |                                |
|                          |                                |
|                          V                                |
|                     +----------+                          |
|                     | MONTHRPT |                          |
|                     +----------+                          |
|                          |                                |
|                          |                                |
|                  +-------+-------+                        |
|                  |               |                        |
|                  V               V                        |
|              Management      MICF Reports                 |
|              Objective        & Graphics                  |
|              Reports                                      |
|                                                           |
+-----------------------------------------------------------+
```

MONTHRPT runs after the MONTHLY job.

The JCL for the MONTHRPT job is in prefix.MICS.CNTL(MONTHRPT).

The MONTHRPT job operates as follows:

- Execute monthly MICF inquiry production reporting for this unit database.

- Produce management objective (MBO) reports if the products in the unit support these reports -- see prefix.MICS.PARMS(EXECDEF).

MICF INQUIRY PRODUCTION REPORTS AND GRAPHICS

Monthly MICF inquiry production interface processing in the MONTHRPT job executes shared MICF inquiries. Printed reports and color graphics are saved in the associated MICF production inquiry output catalog(s) for online review and printing using the MICF Reports and Graphics option. See MICF Administration in the MICF Reference Guide for more information on MICF inquiry production reporting.

STANDARD REPORTS

Based on the REPORT MBOMONTHLY parameter in prefix.MICS.PARMS member EXECDEF, the MONTHRPT job produces the following reports for the previous month's activity.

- Monthly Management Objective (MBO) Reports by Information Area

4.3.3.2.11 RSTRTBLS Operational Job

```
+-----------------------------------------------------------+
|                                                           |
|                    Backup TABLES and                      |
|                     SCREENS Files                         |
|                          |                                |
|                          |                                |
|                          |                                |
|                          V                                |
|                     +---------+                           |
|                     |RSTRTBLS |                           |
|                     +---------+                           |
|                          |                                |
|                          |                                |
|                          |                                |
|                          V                                |
|                       Online                              |
|                  TABLES and SCREENS                       |
|                        Files                              |
|                                                           |
+-----------------------------------------------------------+
```

The RSTRTBLS job is used to restore the TABLES and SCREENS data sets. These data sets contain SAS data files, but unlike other CA MICS SAS data files, they are shared across all database units. They are backed up as part of the normal BACKUP job for the PRIMARY unit database.

The JCL for the RSTRTBLS job is in prefix.MICS.CNTL(RSTRTBLS).

Some files in the TABLES data set are updated by CA MICS Accounting and Chargeback monthly processing. Other files are updated manually. All files in the SCREENS data set are updated manually. Changes to TABLES or SCREENS made after the last Database update and backup for the PRIMARY unit database are not reflected in the backup copy of the TABLES and SCREENS data sets. Therefore, TABLES and SCREENS are not included in the normal RESTORE job.

The RSTRTBLS job restores the TABLES and SCREENS data sets from the most recent (0) generation TABLES/SCREENS backup file. After RSTRTBLS processing completes, you will need to recreate all changes to the TABLES and SCREENS data sets made after the backup was taken.

You can restore from an older generation TABLES/SCREENS backup file modifying the RSTRTBLS JCL to reference an alternative backup file generation (e.g., -3). DO NOT SAVE THE MODIFIED RSTRTBLS JCL.

Refer to the CA MICS Accounting and Chargeback User Guide for additional information on the RSTRTBLS operational job.

4.3.3.2.12 RSTRTLIB Operational Job

```
+-----------------------------------------------------------+
|                                                           |
|                     Backup ISPF                           |
|                     Tables Files                          |
|                          |                                |
|                          |                                |
|                          |                                |
|                          V                                |
|                     +---------+                           |
|                     |RSTRTLIB |                           |
|                     +---------+                           |
|                          |                                |
|                          |                                |
|                          V                                |
|                     Online ISPF                           |
|                     Tables Files                          |
|                                                           |
+-----------------------------------------------------------+
```

The RSTRTLIB job is used to restore the online ISPF tables data sets (sharedprefix.MICS.ISPTLIB). This data set contains ISPF control tables and is shared across all database units and CA MICS users. It is backed up as part of the normal BACKUP job for the PRIMARY unit Database.

The JCL for the RSTRTLIB job is in prefix.MICS.CNTL, member RSTRTLIB.

ISPF tables are updated by CA MICS generation jobs, by CA MICS and MICF administrative activities, and by other CA MICS online applications. Changes to ISPF tables made after the last Database update and backup for the PRIMARY unit database are not reflected in the backup copy of the ISPF tables data set. Therefore, sharedprefix.MICS.ISPTLIB is not included in the normal RESTORE job.

Call CA MICS Product Support for assistance before restoring sharedprefix.MICS.ISPTLIB.

The RSTRTLIB job restores sharedprefix.MICS.ISPTLIB from the most recent (0) generation backup file. After RSTRTLIB processing completes, you will need to recreate all changes to sharedprefix.MICS.ISPTLIB made after the backup was taken.

You can restore from an older generation ISPTLIB backup file modifying the RSTRTLIB JCL to reference an alternative backup file generation (e.g., -3). DO NOT SAVE THE MODIFIED

RSTRTLIB JCL.

See MICF Administration in the MICF Reference Guide and other CA MICS product guides for more information on sharedprefix.MICS.ISPTLIB contents.

4.3.3.2.13 DAYSMFR Operational Job

The DAYSMFR job is used to allocate and rebuild the SMF temporary work files used by the DAILY job.

The JCL for the DAYSMFR job is in prefix.MICS.CNTL(DAYSMFR).

The DAYSMFR job replicates the function of the DAILY job's DAYSMF step when you specify DAYSMF FILES TEMPORARY in prefix.MICS.PARMS(JCLDEF).

If you modified the DAILY job JCL to point to an alternate input file, you will need to make the same modification to the DAYSMFR JCL. DO NOT SAVE THE MODIFIED DAYSMFR JCL.

If incremental update is active for one or more products in this unit database, then DAYSMFR may fail with MICSLOG messages stating that incremental or daily update processing is in progress or requires restart/recovery for one or more products. You can override this checkpoint validity test and force DAYSMFR to execute by specifying,

```
    SYSPARM=FORCE
```

on the DAYSMF step EXEC statement in the DAYSMFR job JCL. SYSPARM=FORCE overrides the abort for DAYnnn step failures when incremental update is active, and allows DAYSMFR job processing to continue.

4.3.3.2.14 ACTDAY1R Operational Job

The ACTDAY1R job applies only to CA MICS Accounting and Chargeback. Run the ACTDAY1R job as necessary to restore the Accounting and Chargeback ACTAUDIT DAY1 file to a status synchronized with the online CA MICS Database.

The JCL for the ACTDAY1R job is in prefix.MICS.CNTL, member ACTDAY1R.

The ACTDAY1R job is used when restarting the DAILY job in the DAY199 step or after running the RESTORE job. ACTDAY1R copies the current (0) generation of the ACTAUDIT DAY2 file file into the current (0) generation of the DAY1 file. If you restart DAY199 or restore the Database without running ACTDAY1R, the ACTAUDIT files (DAY1 and DAY2) will contain duplicate data. See the Accounting and Chargeback User Guide for more information on the ACTAUDIT files.

If your DAY1 tape is not usable (for example, damaged), you must modify the ACTDAY1R JCL to create a new output DAY1 GDG or use the job in prefix.MICS.CNTL(ACTINITA) to catalog a new DAY1 GDG prior to running the ACTDAY1R. See the Accounting and Chargeback User Guide for more information.

You can restore from an older generation of the DAY2 file by modifying the ACTDAY1R JCL to reference an alternative DAY2 file generation (e.g., -3). DO NOT SAVE THE MODIFIED ACTDAY1R JCL.

4.3.3.2.15 RSTATUS Operational Job

```
+-----------------------------------------------------------+
|                                                           |
|   Checkpoint        +---------+          Operational      |
|      File<--------->| RSTATUS |--------->Status and       |
|                     +---------+          Tracking         |
|                                                           |
+-----------------------------------------------------------+
```

Run the RSTATUS job as necessary to update Operational Status and Tracking control tables or to replace a lost Run Status Report.

The JCL for the RSTATUS job is in prefix.MICS.CNTL(RSTATUS).

The DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, RESTORE, AUDIT, HISTW and HISTM database update jobs each have a RSTATUS step -- DAYRSR, WEEKRSR, MONTHRSR, YEARRSR, BKUPRSR, or RSTRRSR. If a Database update job fails without executing the RSTATUS step, you will be unable to use Operational Status and Tracking or the Run Status Report for problem diagnosis and job restart until you either use the Operational Status and Tracking RSTATUS command or submit the job in prefix.MICS.CNTL(RSTATUS).

If the RSTATUS job finds the database in a non-updatable condition, the RSTATUS job will terminate with a U999 abend. When the RSTATUS job terminates with a U999 abend, use the Operational Status and Tracking STATUS command or review the printed Run Status Report to analyze the database problem.

For more information on the Run Status Report, see Section 4.3.6 of this guide.

4.3.3.2.16 INCRccc Operational Jobs

```
+------------------------------------------------------------+
|                                                            |
|            ccc Product Input Data Sources                  |
|                                                            |
|     |      |      |      |       |       |       |         |
|     |      |      |      |       |       |       |         |
|     +------+------+----->|<------+-------+-------+         |
|                          |                                 |
|                          V                                 |
|     Unit            +---------+           ccc Product      |
|  Checkpoint-------->| INCRccc |<--------->Incremental      |
|     File            +---------+           Update           |
|                          |                Checkpoint       |
|                          |                                 |
|                          V                                 |
|                     ccc Product                            |
|                     Incremental                            |
|                       Update                               |
|                          |                                 |
|                          |                                 |
|                 +-----------------+                        |
|                 |                 |                        |
|                 V                 V                        |
|            DETAIL Files      DAYS Files                    |
|                                                            |
+------------------------------------------------------------+
```

If incremental update is enabled for the ccc product, optionally run the INCRccc job one or more times each day. When incremental update is enabled, the first phase of the DAILY job step for the ccc product is the equivalent of an INCRccc job execution.

The JCL for the INCRccc jobs is in prefix.MICS.CNTL(INCRccc). Each INCRccc job is tailored to the associated data integration product.

The INCRccc job operates as follows:

o If prefix.MICS.PARMS(cccOPS) for this product specifies INCRSPLIT USE, dynamically allocate the input prefix.MICS.ccc.IUSPLTDS data set created by the SPLITSMF job.

o If this is the FIRST INCRccc job of the day (for the ccc product), initialize the ccc product incremental update checkpoint file and incremental update database suspend files from the unit database checkpoint and DETAIL and/or DAYS timespan files.

- Set incremental update checkpoint file SELECT and FORCE specifications from the unit checkpoint file.

- Select input data for the ccc product.

- Create information area files in the ccc product incremental update DETAIL and DAYS timespan data sets.

- If this is NOT the first INCRccc job of the day (for this product), combine the new DETAIL and DAYS level data with accumulated DETAIL and DAYS "today to-date" incremental update database files.

- If prefix.MICS.PARMS(cccOPS) for this product specifies INCRSPLIT USE, delete the input prefix.MICS.ccc.IUSPLTDS data set created by the SPLITSMF job.

DATABASE UPDATES

The incremental update database 01 level file cycle is created or updated for each of the files maintained in the DETAIL and DAYS timespans. This 01 level cycle contains the data processed from the input data sets for this INCRccc job and all prior INCRccc executions for this product since the last DAILY job execution.

DEPENDENCIES

If the INCRccc job malfunctions, no other INCRccc jobs for this product, or the DAILY job will be able to execute. Also, if prefix.MICS.PARMS(cccOPS) for this product specifies INCRSPLIT USE, then the SPLITSMF job will not be able to execute. The INCRccc job must successfully complete before subsequent INCRccc jobs for this product, or the DAILY job, can start.

The INCRccc job will not be able to execute if any of the latest DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP or RESTORE jobs, or a prior INCRccc job for this product, have completed unsuccessfully.

Note: The INCRccc jobs for different products are totally independent and can execute concurrently. For example, the INCRTSO, INCRSMF, and INCRRMF jobs can execute in parallel.

In addition, the failure of one INCRccc job has no effect the INCRccc job for any other product. However, if any INCRccc job fails, the DAILY job will be unable to execute until the problem is resolved.

If an INCRccc job is accidentally submitted for execution twice in the same day with the same data in the input files, the second run abends (U300) because the duplicate data check mechanisms in CA MICS recognize that the same data is being submitted again.

In addition, if the DAILY job is accidentally submitted for execution with the same input files that have already been processed by an INCRccc job, the DAILY job step for the ccc product will abend (U300) unless SYSPARM=NODATA is specified on the job step EXEC JCL statement.

4.3.3.2.17 SPLITSMF Operational Job

```
+-----------------------------------------------------------+
|                                                           |
|                SMF Input Data Source                      |
|                CA SMF Director Data Source                |
|                                                           |
|                          |                                |
|                          |                                |
|                          V                                |
|     Unit            +---------+           ccc Product     |
|  Checkpoint-------->| SPLITSMF|<--------->Incremental     |
|     File            +---------+           Update          |
|                          |                Checkpoint      |
|                          |                files           |
|                          |                                |
|     +--------------------------------------------+        |
|     |              |              |              |        |
|     V              V              V              V        |
| ccc product    ccc product    ccc product    ccc product  |
|  IUSPLTDS       IUSPLTDS       IUSPLTDS       IUSPLTDS    |
| data file      data file      data file      data file    |
|                                                           |
+-----------------------------------------------------------+
```

If incremental update is enabled for one or more products that take their input data from the common SMF data files, then you may optionally run the SPLITSMF job one or more times each day to prepare input data for the INCRccc jobs. The SPLITSMF job is enabled based on the products active in the unit database, the prefix.MICS.PARMS(JCLDEF) SMFRECORDING option, and the individual prefix.MICS.PARMS(cccOPS) INCRSPLIT USE parameter specifications.

The SPLITSMF job JCL is in prefix.MICS.CNTL(SPLITSMF). The SPLITSMF job is tailored to the prefix.MICS.PARMS(INPUTRDR) member for the unit database.

The SPLITSMF job operates as follows:

o Determine applicable products by first selecting the products that input the common SMF input data (just like the DAILY job DAYSMF step) and then trim this list to just those products with INCRUPDATE YES and INCRSPLIT USE specified in the product's prefix.MICS.PARMS(cccOPS) parameter.

o For each applicable product, allocate the product's incremental update checkpoint file and verify that previous INCRccc or DAYnnn processing completed successfully.

- Construct a composite, temporary checkpoint file by combining the contents of the unit checkpoint file and each applicable product's incremental update checkpoint file. This composite checkpoint file reflects the input data processed by prior INCRccc jobs executed today, and is used to drop any previously processed input data from the input SMF data files.

- For each applicable product, allocate the prefix.MICS.ccc.IUSPLTDS data set to contain the product's subset of the SMF input data. Data set allocation parameters for these data sets are specified on the INCRSPLIT USE parameter. Data set allocation parameters default to UNIT=&DASDUNIT (see JCLDEF DASDUNIT parameter) and SPACE=(TRK,(700,100)).

  The ccc.IUSPLTDS data sets are normally allocated DISP=(NEW,CATLG). However, if a product's ccc.IUSPLTDS already exists, the existing data set will be allocated DISP=MOD and the new input data will be appended to any existing data. This is done to avoid losing data when SPLITSMF is run prior to completing all associated INCRccc job executions. Any duplicate input data is dropped by standard CA MICS checkpoint and SORT NODUPS processing.

  A warning message will be written to the MICSLOG when the prefix.MICS.ccc.IUSPLTDS data set already exists.

  Note: The USER.IUSPLTDS data set name and data set allocation parameters may be overridden by defining the &IUSUSERD and/or &IUSUSERA SAS global variables with the desired values. The following is a sample prefix.MICS.USER.SOURCE(#BASEXIT) specification excerpt.

  ```
  %LET IUSUSERD = MYPREFIX.MYUSER.IUSPLTDS
  %LET IUSUSERA = STORCLAS=WORKDATA SPACE=(CYL,(20,5)
  ```

- Read the INPUTSMF data sets and split-out the input data to the appropriate prefix.MICS.ccc.IUSPLTDS data sets. This is the same basic logic as it appears in the DAILY job's DAYSMF step.

DATABASE UPDATES

The SPLITSMF job does NOT update the unit or incremental update checkpoint or database files. SPLITSMF simply creates the input data files for INCRccc jobs.

DEPENDENCIES

If the SPLITSMF job malfunctions, the INCRccc jobs that input the data sets created by the SPLITSMF job may be unable to execute. The SPLITSMF job must successfully complete before associated INCRccc jobs can start.

The SPLITSMF job will not be able to execute if any of the latest DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP or RESTORE jobs, or a prior INCRccc job for any product marked for INCRSPLIT USE, have completed unsuccessfully.

If you need to force SPLITSMF job execution to recreate a missing prefix.MICS.ccc.IUSPLTDS data set in order to recover from an INCRccc job failure, specify,

```
    SYSPARM=FORCE
```

on the SPLITSMF step EXEC statement. SYSPARM=FORCE overrides the abort for INCRccc job failures and allows SPLITSMF job processing to continue.

Note: The SPLITSMF job is independent of any INCRccc jobs for products marked as INCRSPLIT IGNORE (the default) in prefix.MICS.PARMS(cccOPS). INCRccc job failures for these products have NO impact on SPLITSMF job execution.

If the SPLITSMF job is accidentally submitted for execution twice in the same day with the same data in the input files, the second set of INCRccc jobs may abend (U300) because the duplicate data check mechanisms in CA MICS recognize that the same data is being submitted again.

4.3.3.2.18 IUDBINIT Operational Job

The IUDBINIT job applies only where you have activated incremental update for one or more products in the unit database. You execute IUDBINIT to re-initialize in-progress incremental update processing after restoring the unit database files (i.e., after running the RESTORE job). Messages in the RESTORE job MICSLOG will prompt you to execute IUDBINIT when needed.

The JCL for the IUDBINIT job is in prefix.MICS.CNTL, member IUDBINIT.

The IUDBINIT job loops through the products installed in the unit database to locate those products for which you specified INCRUPDATE YES in prefix.MICS.PARMS(cccOPS).

o IUDBINIT allocates each product's incremental update checkpoint file to determine if one or more INCRccc jobs executed (or are in progress) since the last DAILY job execution.

o If you specified INCRDB PERM (or took the default) in prefix.MICS.PARMS(cccOPS), then IUDBINIT allocates and clears the product's IUDETAIL and IUDAYS incremental update database files.

o If you specified INCRDB TAPE or INCRDB PERM in prefix.MICS.PARMS(cccOPS), then IUDBINIT deletes the in-progress IUDETAIL and IUDAYS incremental update data sets.

o Then IUDBINIT initializes the incremental update checkpoint file, preparing it for the next INCRccc or DAILY job execution.

4.3.3.2.19 AUDIT Operational Job

The AUDIT job applies only where you have activated stand-alone Archive Audit tape processing by specifying

```
     ARCHIVE AUDIT YES JOB
```

in prefix.MICS.PARMS(JCLDEF).

The AUDIT job performs the Archive Audit tape processing. This processing alternatively occurs within the WEEKLY job's WEEK300 step.

Archive audit data is created as a new entry in the audit generation data group. The audit data is generated for files that contain job, program, command, or application unit execution-related data by copying data from the last ten DAILY or DETAIL timespan cycles maintained online for the applicable files. If the file is active in the DETAIL timespan, DETAIL is used. If the file is NOT active in the DETAIL timespan, then DAYS is used.

The files included in archive audit processing are specified in prefix.MICS.PARMS(DBMODEL) and sharedprefix.MICS.GENLIB member cccGENIN, where ccc is the three-character component identifier. Archive audit tape retention (number of generations) is specified in prefix.MICS.PARMS(JCLDEF). See the CA MICS Structure and Content Guide for more information on the Archive Database.

The AUDIT job may be scheduled automatically, or you may need to schedule the job using your installation's production batch job scheduling facilities.

o If you specified

```
        ARCHIVE AUDIT YES JOB
```

  in prefix.MICS.CNTL(JCLDEF), then Archive Audit processing is performed in the stand-alone AUDIT job.

  - If you specified the AUTOSUBMIT parameter on the ARCHIVE statement, then the WEEKLY job WEEK300 step automatically submits the AUDIT job through the OS/390 internal reader.

  - If you specified NOAUTOSUBMIT (or took the default) then you must schedule the AUDIT job externally, either manually or through your installation's production batch job scheduling facilities.

  NOTE: The AUDIT job must complete before the next scheduled DAILY job will be allowed to execute.

o If you specified

```
ARCHIVE AUDIT YES
```

   in prefix.MICS.CNTL(JCLDEF), without the JOB parameter (i.e., you took the default or specified STEP), then Archive Audit processing is performed in the WEEK300 step of the WEEKLY job. In this situation, you will NOT execute the AUDIT job.

You will normally execute the AUDIT job once each week, after the WEEKLY job completes successfully. However, you can choose to run the AUDIT job more often. Each time AUDIT executes, it will create a new audit tape cycle from the contents of the last 10 online cycles of the database files for which audit is enabled.

o Note, by default, audit archive processing drops data from the current week in order to reduce processing time and conserve CA MICS tape data base space. Thus, even though 10 cycles of data are input, depending upon when audit archive processing is performed, the full 10 online cycles my not be included in the current audit archive tape cycle.

o You may find that you can reduce the number of cycles retained online by executing the AUDIT job more frequently (e.g., 4 times per week) and then referring to the audit tapes when detail data analysis is required.

   Note: If you choose to take advantage of this technique, and execute the AUDIT job more often than once each week, then the following must be added to prefix.MICS.PARMS(EXECDEF)

```
 USERDEF AUDITCWK YES
```

   This parameter overrides the default audit archive processing so that data for the current week is retained and copied to the new audit archive tape cycle.

   Concurrently, don't forget to increase the number of cycles of the archive audit tapes you retain (see the GDG= subparameter of the AUDITPARM statement in prefix.MICS.CNTL(JCLDEF).

4.3.3.2.20 HISTW Operational Job

The HISTW job applies only where you have activated stand-alone Archive Weekly History tape processing by specifying,

```
     ARCHIVE HISTW YES JOB
```

in prefix.MICS.PARMS(JCLDEF).

The HISTW job performs the Archive Weekly History tape processing. This processing alternatively occurs within the WEEKLY job's WEEK300 step.

The archive weekly history data is updated by creating a new generation data group entry using the most recent WEEKS timespan data (cycle 01) and the history-to-date contained in the current (0) generation archive for each file maintained in the WEEKS timespan archives. Historical data and new data are read in and any data that is older than the user-specified archive retention age is deleted.

The WEEKS timespan files included in weekly archive history processing and the archive retention are specified in prefix.MICS.PARMS(DBMODEL) and sharedprefix.MICS.GENLIB member cccGENIN, where ccc is the three-character component identifier. See the CA MICS Structure and Content Guide for more information on the Archive Database.

The HISTW job may be scheduled automatically, or you may need to schedule the job using your installation's production batch job scheduling facilities.

o If you specified

```
        ARCHIVE HISTW YES JOB
```

   in prefix.MICS.CNTL(JCLDEF), then Archive Weekly History processing is performed in the stand-alone HISTW job.

   - If you specified the AUTOSUBMIT parameter on the ARCHIVE statement, then the WEEKLY job WEEK300 step automatically submits the HISTW job through the OS/390 internal reader.

   - If you specified NOAUTOSUBMIT (or took the default) then you must schedule the HISTW job externally, either manually or through your installation's production batch job scheduling facilities.

```
   NOTE:  The HISTW job must complete before the next
          scheduled WEEKLY job will be allowed to execute.

o  If you specified

        ARCHIVE HISTW YES

   in prefix.MICS.CNTL(JCLDEF), without the JOB parameter
   (i.e., you took the default or specified STEP), then
   Archive Weekly History processing is performed in the
   WEEK300 step of the WEEKLY job.  In this situation, you
   will NOT execute the HISTW job.
```

You will normally execute the HISTW job once each week, after the WEEKLY job completes successfully. To preserve archive database integrity and avoid duplicate data, the HISTW job will abort if you attempt to run it again before the next WEEKLY job completes. Likewise, the next WEEKLY job will not be allowed to execute until the HISTW job completes successfully.

4.3.3.2.21 HISTM Operational Job

The HISTM job applies only where you have activated stand-alone Archive Monthly History tape processing by specifying,

```
     ARCHIVE HISTM YES JOB
```

in prefix.MICS.PARMS(JCLDEF).

The HISTM job performs the Archive Monthly History tape processing. This processing alternatively occurs within the MONTHLY job's MONTH300 step.

The archive monthly history data is updated by creating a new generation data group entry using the most recent MONTHS timespan data (cycle 01) and the history-to-date contained in the current (0) generation archive for each file maintained in the MONTHS timespan archives. Historical data and new data are read in and any data that is older than the user-specified archive retention age is deleted.

The MONTHS timespan files included in monthly archive history processing and the archive retention are specified in prefix.MICS.PARMS(DBMODEL) and sharedprefix.MICS.GENLIB member cccGENIN, where ccc is the three-character component identifier. See the CA MICS Structure and Content Guide for more information on the Archive Database.

The HISTM job may be scheduled automatically, or you may need to schedule the job using your installation's production batch job scheduling facilities.

o If you specified

```
        ARCHIVE HISTM YES JOB
```

   in prefix.MICS.CNTL(JCLDEF), then Archive Monthly History processing is performed in the stand-alone HISTM job.

   - If you specified the AUTOSUBMIT parameter on the ARCHIVE statement, then the MONTHLY job MONTH300 step automatically submits the HISTM job through the OS/390 internal reader.

   - If you specified NOAUTOSUBMIT (or took the default) then you must schedule the HISTM job externally, either manually or through your installation's production batch job scheduling facilities.

NOTE: The HISTM job must complete before the next scheduled MONTHLY job will be allowed to execute.

o If you specified

```
     ARCHIVE HISTM YES
```

in prefix.MICS.CNTL(JCLDEF), without the JOB parameter (i.e., you took the default or specified STEP), then Archive Monthly History processing is performed in the MONTH300 step of the MONTHLY job. In this situation, you will NOT execute the HISTM job.

You will normally execute the HISTM job once each month, after the MONTHLY job completes successfully. To preserve archive database integrity and avoid duplicate data, the HISTM job will abort if you attempt to run it again before the next MONTHLY job completes. Likewise, the next MONTHLY job will not be allowed to execute until the HISTM job completes successfully.

4.3.3.3 Job Steps

The CA MICS Database update jobs are each composed of several job steps.

Job steps are named to identify the function being accomplished. Job step names are prefixed by INCR, DAY, WEEK, MONTH, or YEAR to identify within which job they reside. The job step name is suffixed by a three digit code that indicates the job step function or purpose.

- ALL - First step in the DAILY job
- SMF - Selects and splits SMF input data to work files for processing by Database update steps
- 001-099 - CA MICS data integration applications--refer to sharedprefix.MICS.GENLIB(COMPTDEF) for a current list
- 100-190 - CA MICS field-developed applications
- 199 - Accounting and Chargeback
- 200 - Update CA MICS exception files (ADMEXC)
- 300 - Update the CA MICS archive database
- 400 - Standard reports and MICF inquiry production reporting
- 500 - User-written reports (non-MICF)
- 900 - Database update termination
- RSR - Run Status Report

The hierarchy of job steps within each operational job is shown in the charts below.

Job steps 001 through 199 are executed only if the associated product is installed. These steps are enclosed in parentheses. If the prerequisite step is for a product that is not installed, move up the column until an installed product step or a non-product step is found.

The charts use "nnn" to indicate one of the 001-190 steps. The prerequisite for step nnn is the next lower step number for products installed in the unit database. For example, the prerequisite step to DAY030 (Batch and Operations Analyzer) is step DAY020 (Hardware and SCP Analyzer Analyzer) if both products are installed in the unit.

Note: The optional INCRccc jobs contain a single INCRnnn step. The prerequisite for INCRnnn is either DAY900 or the previous INCRnnn step (for this product) executed since the last DAILY job execution.

If incremental update is enabled for a product, then INCRnnn or the previous DAYnnn are also prerequisite for DAYnnn execution.

These relationships are shown in the table that follows figure 4-2 below.

JCLGEN processing generates the hierarchy uniquely for each unit database.

CA MICS Operational Job		Checkpoint Conditions to be Met					
		Prerequisite Stepnames Checked					
Job Name	Step Name	DAILY	WEEKLY	MONTHLY	YEARLY	BACKUP	RESTORE
DAILY	DAYALL	900	900	900	900	900	900
DAILY	(DAYSMF)	(000)	900	900	900	900	900
DAILY	(DAYnnn)	(nnn)	900	900	900	900	900
DAILY	(DAY199)	(nnn)	900	900	900	900	900
DAILY	DAY200	(199)	900	900	900	900	900
DAILY	DAY400	200	900	900	900	900	900
DAILY	DAY500	400	900	900	900	900	900
DAILY	DAY900	500	900	900	900	900	900

Figure 4-2. Operational Job Step Hierarchy (Part 1 of 2)

CA MICS Operational Job		Checkpoint Conditions to be Met					
		Prerequisite Stepnames Checked					
Job Name	Step Name	DAILY	WEEKLY	MONTHLY	YEARLY	BACKUP	RESTORE
WEEKLY		900	900	900	900	900	900
WEEKLY	(WEEKnnn)	900	(nnn)	900	900	900	900
WEEKLY	(WEEK199)	900	(nnn)	900	900	900	900
WEEKLY	WEEK300	900	199	900	900	900	900
WEEKLY	WEEK400	900	300	900	900	900	900
WEEKLY	WEEK500	900	400	900	900	900	900
WEEKLY	WEEK900	900	500	900	900	900	900
MONTHLY		900	900	900	900	900	900
MONTHLY	(MONTHnnn)	900	900	(nnn)	900	900	900
MONTHLY	(MONTH199)	900	900	(nnn)	900	900	900
MONTHLY	MONTH200	900	900	(199)	900	900	900
MONTHLY	MONTH300	900	900	200	900	900	900
MONTHLY	MONTH400	900	900	300	900	900	900
MONTHLY	MONTH500	900	900	400	900	900	900
MONTHLY	MONTH900	900	900	500	900	900	900
YEARLY		900	900	900	900	900	900
YEARLY	(YEARnnn)	900	900	900	(nnn)	900	900
YEARLY	(YEAR199)	900	900	900	(nnn)	900	900
YEARLY	YEAR500	900	900	900	(199)	900	900
YEARLY	YEAR900	900	900	900	500	900	900
BACKUP		900	900	900	900	*np	900
SPLITSMF		900	900	900	900	900	900
DAILYRPT		*np	*np	*np	*np	*np	*np
WEEKRPT		*np	*np	*np	*np	*np	*np
MONTHRPT		*np	*np	*np	*np	*np	*np
RESTORE		*np	*np	*np	*np	*np	*np
RSTRTBLS		*np	*np	*np	*np	*np	*np
RSTRTLIB		*np	*np	*np	*np	*np	*np
DAYSMFR		*np	*np	*np	*np	*np	*np
ACTDAY1R		*np	*np	*np	*np	*np	*np
IUDBINIT		*np	*np	*np	*np	*np	*np
RSTATUS		*np	*np	*np	*np	*np	*np

*np - No prerequisite step names checked.

Figure 4-2. Operational Job Step Hierarchy (Part 2 of 2)

If incremental update is enabled for one or more products, then the following additional checkpoint condition checks must be satisfied.

CA MICS Operational Job		Checkpoint Conditions to be Met	
		Prerequisite Stepnames Checked	
Job Name	Step Name	Unit Checkpoint File (DAILY)	ccc Product Incremental Update Checkpoint File
INCRccc	INCRnnn	900	900 or nnn
DAILY	DAYnnn	nnn	900 or nnn
SPLITSMF		900	900 or nnn

The operational job steps are documented in the sections that follow:

1 - DAILY Job Steps
2 - WEEKLY Job Steps
3 - MONTHLY Job Steps
4 - YEARLY Job Steps
5 - BACKUP Job Steps
6 - RESTORE Job Steps
7 - INCRccc Job Steps

4.3.3.3.1 DAILY Job Steps

The actions being performed within each of the steps in the DAILY job are discussed below.

Step DAYALL: Allocate DAILY job work data sets and initiate daily processing.

1. Verify that prerequisite step completed successfully.
2. If incremental update is enabled for one or more products,
   a. Verify that the incremental update checkpoint file for each incremental update enabled product is available and that any INCRccc job for the corresponding product executed since the last DAILY job execution completed successfully.
   b. Verify that the unit checkpoint file will be able to hold any new ORGSYSID, product entries from the incremental update checkpoint files.
3. If ARCHIVE AUDIT YES JOB is specified in JCLDEF (stand-alone Archive Audit processing is enabled), then verify that the AUDIT job for the previous week (from the WEEKLY job) completed successfully.
4. Update the Checkpoint File--DAYALL completed.
5. Update Operational Status and Tracking control tables (DAILY in progress).

Step DAYSMF: Select and split SMF input data to work files for processing by database update steps.

1. Verify that prerequisite step completed successfully.
2. Update the Checkpoint File--Daily (000) started.
3. If incremental update is enabled for one or more products,
   a. Verify that the incremental update checkpoint file for each incremental update enabled product is available and that any INCRccc job for the corresponding product executed since the last DAILY job execution completed successfully.
   b. Generate a composite, temporary checkpoint file reflecting the results of INCRccc jobs executed since the last DAILY job execution. This temporary checkpoint file will be used by the DAYSMF step to drop input data that has already been processed by an INCRccc job execution. The composite checkpoint file is generated by updating a temporary copy of the unit checkpoint file with new and/or modified ORGSYSID, product entries from the incremental update checkpoint files. The temporary checkpoint

```
          file is generated in the temporary data set
          allocated to the IUTEMPCK DD statement.
   4.  Process the input data and output:
       - Output ccc Analyzer work data set for ccc products
         having SMF input.
       - Optionally output user-defined records to the USER DD
         allocation.
   5.  Update the Checkpoint File--DAYSMF completed.


 Step DAYnnn:  Update information areas in the database for
               products installed in this unit database.
               There is one step (001-190) for each product.

   1.  Verify that prerequisite step completed successfully.
   2.  If incremental update is enabled for this product,
       a. Perform INCRnnn processing (see section 4.3.3.3.7)
          for this product.
       b. Delete each file's highest defined DETAIL cycle.
       c. Delete each file's DETAIL and DAYS cycle 00.
       d. Roll-up incremental update DETAIL and DAYS level
          files to create DETAIL and DAYS cycle 00 files
       Otherwise (incremental update is NOT enabled),
       a. Read raw input data, convert to SAS format, and
          output to intermediate work files.
       b. Delete each file's highest defined DETAIL cycle and
          cycle 00.
       c. Sort, analyze, summarize, and enhance intermediate
          work file contents to create DETAIL cycle 00 files.
       d. Delete each file's DAYS cycle 00 and build the new
          DAYS cycle 00 files.
   3.  Build week-to-date and/or month-to-date temporary files
       (cycles W0 and M0) for files maintained in the WEEKS
       and/or MONTHS timespan from new data for today and
       existing WEEKS and/or MONTHS timespan 00 cycle files.
   4.  Update time ranges in the SYSID/COMPONENT section of
       the Checkpoint File and Input History Log.
   5.  Perform daily file aging.
       a. Rename new DETAIL/DAYS cycles (00) to 01 level.
       b. Rename DETAIL/DAYS cycles 01-nn to cycles 02-(nn+1).
       c. Rename WEEKS cycle W0 to 00.
       d. Rename MONTHS cycle M0 to 00.
   6.  If incremental update is enabled for this product,
       initialize the incremental update checkpoint and
       database files in preparation for the next day's
       incremental update processing.
   7.  Update the Checkpoint File--DAYnnn completed.
```

Step DAY199: Accounting and Chargeback--see the Accounting and Chargeback User Guide for more information on processing performed in the DAY199 step.

1. Verify that prerequisite step completed successfully.
2. Delete each journal file's highest defined DETAIL cycle.
3. Delete the ledger file's DAYS cycle 00.
4. Build DAYS ledger cycle 00 file.
5. Update month-to-date tape audit files.
6. Build month-to-date temporary ledger file (cycle M0) in the MONTHS timespan from new data for today and existing MONTHS timespan 00 cycle ledger file.
7. Perform daily file aging for the DETAIL journal files and the DAILY ledger file.
   a. Rename new DETAIL/DAYS cycles (00) to 01 level.
   b. Rename DETAIL/DAYS cycles 01-nn to cycles 02-(nn+1).
   c. Rename MONTHS cycle M0 to 00.
8. Update the Checkpoint File--DAY199 completed.

Step DAY200: Update CA MICS exception files.

1. Verify that prerequisite step completed successfully.
2. Delete each DAYS exception file's highest defined cycle.
3. Delete each DAYS exception file's cycle 00.
4. Process component-specific files and build interim exception files based prefix.MICS.PARMS(EXECDEF) specifications.
5. Build DAYS exception file (cycle 00) if exception processing is active for any product in the unit database--see prefix.MICS.PARMS(EXECDEF).
6. Build month-to-date temporary exception file (cycle M0) from new data for today and existing MONTHS timespan 00 cycle exception file.
7. Perform daily exception file aging.
   a. Rename new DAYS cycles (00) to 01 level.
   b. Rename DAYS cycles 01-nn to cycles 02-(nn+1).
   c. Rename MONTHS cycle M0 to 00.
8. Update the Checkpoint File--DAY200 completed.

Step DAY400: Standard reports and MICF inquiry production reporting.

1. Verify that prerequisite step completed successfully.
2. If you specified AUTOSUBMIT YES in prefix.MICS.PARMS(JCLDEF), submit the DAILYRPT job;

```
          copy prefix.MICS.CNTL(DAILYRPT) to the JES internal
          reader.
      3.  If you specified AUTOSUBMIT NO in
          prefix.MICS.PARMS(JCLDEF), produce management objective
          and exception reports--see report control parameters in
          prefix.MICS.PARMS(EXECDEF).
      4.  Update the Checkpoint File--DAY400 completed.


    Step DAY500:  User-written reports (non-MICF).

      1.  Verify that prerequisite step completed successfully.
      2.  Perform user-defined processing--see
          prefix.MICS.USER.SOURCE(DYUSER).
      3.  Update the Checkpoint File--DAY500 completed.


    Step DAY900:  Database update termination.

      1.  Verify that prerequisite step completed successfully.
      2.  Delete interim work data sets.
      3.  Update the Checkpoint File--DAY900 completed.


    Step DAYRSR:  Run Status Report.

      1.  Generate CA MICS Run Status Report.
      2.  Update Operational Status and Tracking control tables.
          a. Store DAILY job status.
          b. Copy ADMIHL01 file to the history table.
          c. Update the processing schedule for the next 7 days.
          d. Update the operational job log.
```

4.3.3.3.2 WEEKLY Job Steps

The actions being performed within each step of the WEEKLY job are discussed below.

Step WEEKnnn: Update information areas in the Database for products installed in this unit database. There is one step (001-190) for each product.

1. Verify that prerequisite step completed successfully.
2. If this is the first step of the WEEKLY job, then
   a. If ARCHIVE AUDIT YES JOB is specified in JCLDEF (stand-alone Archive Audit processing is enabled), then verify that the AUDIT job for the previous week completed successfully.
   b. If ARCHIVE HISTW YES JOB is specified in JCLDEF (stand-alone Archive Weekly History processing is enabled), then verify that the HISTW job for the previous week completed successfully.
3. Delete each file's highest defined cycle.
4. Split week-to-date cycles (00) into current week (W0) and prior weeks (W1) cycles.
5. Perform weekly file aging for each file.
   a. Rename cycles 01-nn to cycles 02-(nn+1).
   b. Replace cycle 00 with cycle W0 (new week-to-date).
   c. Rename cycle W1 to 01.
6. Update the Checkpoint File -- WEEKnnn completed.

Step WEEK199: Accounting and Chargeback -- see the Accounting and Chargeback User Guide for information on monthly accounting processing.

Step WEEK300: Update the CA MICS archive Database.

1. Verify that prerequisite step completed successfully.
2. If you specified ARCHIVE AUDIT YES in prefix.MICS.PARMS(JCLDEF),
   a. If you specified the STEP option (or took the default), then create the new generation of the archive audit files.
   b. Or, if you specified ARCHIVE AUDIT YES JOB, then
      1) Mark checkpoint to indicate that Audit Archive processing is required (AUDIT job).
      2) If ARCHIVE AUDIT YES JOB AUTOSUBMIT is specified, write the AUDIT operational job to the OS/390 internal reader.

3. If you specified ARCHIVE HISTW YES in prefix.MICS.PARMS(JCLDEF),

```
      a.  If you specified the STEP option (or took the
          default), then update weekly archive history files.
          Combine the current (0) generation weekly archive
          history files with the new 01 cycle WEEKS timespan
          files to create the new (+1) generation weekly
          archive history files.  See weekly archive history
          eligibility and retention parameters in
          prefix.MICS.PARMS(DBMODEL) and the specifications
          in sharedprefix.MICS.GENLIB(cccGENIN).
      b.  Or, if you specified ARCHIVE HISTW YES JOB, then
          1)  Mark checkpoint to indicate that Audit Weekly
              History processing is required (HISTW job).
          2)  If ARCHIVE HISTW YES JOB AUTOSUBMIT is
              specified, write the HISTW operational job to
              the OS/390 internal reader.
  4.  Update the Checkpoint File -- WEEK300 completed.


Step WEEK400:  Standard reports and MICF inquiry production
               reporting.

  1.  Verify that prerequisite step completed successfully.
  2.  If you specified AUTOSUBMIT YES in
      prefix.MICS.PARMS(JCLDEF), submit the WEEKRPT job; copy
      prefix.MICS.CNTL(WEEKRPT) to the JES internal reader.
  3.  If you specified AUTOSUBMIT NO in
      prefix.MICS.PARMS(JCLDEF), produce management objective
      reports -- see report control parameters in
      prefix.MICS.PARMS(EXECDEF).
  4.  Update the Checkpoint File -- WEEK400 completed.


Step WEEK500:  User-written reports (non-MICF).

  1.  Verify that prerequisite step completed successfully.
  2.  Perform user-defined processing -- see
      prefix.MICS.USER.SOURCE(WKUSER).
  3.  Update the Checkpoint File -- WEEK500 completed.


Step WEEK900:  Database update termination.

  1.  Verify that prerequisite step completed successfully.
  2.  Update the Checkpoint File -- WEEK900 completed.

Step WEEKRSR:  Run Status Report.

  1.  Generate CA MICS Run Status Report.
  2.  Update Operational Status and Tracking control tables.
```

a. Store WEEKLY job status.
b. Update the processing schedule for the next 7 days.
c. Update the operational job log.

4.3.3.3.3 MONTHLY Job Steps

The actions being performed within each step of the MONTHLY job are discussed below.

Step MONTHnnn: Update information areas in the database for products installed in this unit database. There is one step (001-190) for each product.

1. Verify that prerequisite step completed successfully.
2. If this is the first step of the MONTHLY job and ARCHIVE HISTM YES JOB is specified in JCLDEF (stand-alone Archive Monthly History processing is enabled), then verify that the HISTM job for the previous month completed successfully.
3. Delete each file's highest defined cycle.
4. Split month-to-date cycles (00) into current month (M0) and prior months (M1) cycles.
5. If YEARS timespan is active, build year-to-date temporary files (cycle Y0) for files maintained in the YEARS timespan from the new MONTHS M1 cycle and existing YEARS timespan 00 cycle files. See sharedprefix.MICS.PARMS(CPLXDEF), prefix.MICS.PARMS(SITE), and prefix.MICS.PARMS(DBMODEL).
6. Perform monthly cycle aging for each file.
   a. Rename cycles 01-nn to cycles 02-(nn+1).
   b. Replace cycle 00 with cycle M0 (new month-to-date).
   c. Rename cycle M1 to 01.
   d. Rename YEARS cycle Y0 to 00.
7. Update the Checkpoint File -- MONTHnnn completed.

Step MONTH199: Accounting and Chargeback -- see the Accounting and Chargeback User Guide for information on monthly accounting processing.

Step MONTH200: Update CA MICS exception files.

1. Verify that prerequisite step completed successfully.
2. Delete the highest defined cycle.
3. Split month-to-date cycle (00) into current month (M0) and prior months (M1) cycles.
4. Perform monthly cycle aging.
   a. Rename cycles 01-nn to cycles 02-(nn+1).
   b. Replace cycle 00 with cycle M0 (new month-to-date).
   c. Rename cycle M1 to 01.
5. Update the Checkpoint File -- MONTH200 completed.

Step MONTH300: Update the CA MICS archive database.

1. Verify that prerequisite step completed successfully.
2. If you specified ARCHIVE HISTM YES in prefix.MICS.PARMS(JCLDEF),
   a. If you specified the STEP option (or took the default), then update monthly archive history files. Combine the current (0) generation monthly archive history files with the new 01 cycle MONTHS timespan files to create the new (+1) generation monthly archive history files. See monthly archive history eligibility and retention parameters in prefix.MICS.PARMS(DBMODEL) and the specifications in sharedprefix.MICS.GENLIB(cccGENIN).
   b. Or, if you specified ARCHIVE HISTM YES JOB, then
      1) Mark checkpoint to indicate that Audit Monthly History processing is required (HISTM job).
      2) If ARCHIVE HISTM YES JOB AUTOSUBMIT is specified, write the HISTM operational job to the OS/390 internal reader.
3. Update the Checkpoint File -- MONTH300 completed.

Step MONTH400: Standard reports and MICF inquiry production reporting.

1. Verify that prerequisite step completed successfully.
2. If you specified AUTOSUBMIT YES in prefix.MICS.PARMS(JCLDEF), submit the MONTHRPT job; copy prefix.MICS.CNTL(MONTHRPT) to the JES internal reader.
3. If you specified AUTOSUBMIT NO in prefix.MICS.PARMS(JCLDEF), produce management objective reports -- see report control parameters in prefix.MICS.PARMS(EXECDEF).
4. Update the Checkpoint File -- MONTH400 completed.

Step MONTH500: User-written reports (non-MICF).

1. Verify that prerequisite step completed successfully.
2. Perform user-defined processing -- see prefix.MICS.USER.SOURCE(MNUSER).
3. Update the Checkpoint File -- MONTH500 completed.

Step MONTH900: Database update termination.

1. Verify that prerequisite step completed successfully.
2. Update the Checkpoint File -- MONTH900 completed.

Step MONTHRSR: Run Status Report.

1. Generate CA MICS Run Status Report.
2. Update Operational Status and Tracking control tables.
   a. Store MONTHLY job status.
   b. Update the processing schedule for the next 7 days.
   c. Update the operational job log.

Step BACKUP: Produce CA MICS Monthly backup.

1. Execute or submit the Monthly backup job.

The monthly backup steps are described below:

Step BKUP900M: Database backup.

1. Backup the Checkpoint File.
2. Copy online Database files to timespan monthly backup data sets.
3. Update Operational Status and Tracking control tables.
   a. Store monthly backup (MBACKUP) job status.
   b. Update the operational job log.

Step BKTLIB: Backup sharedprefix.MICS.ISPTLIB (PRIMARY unit
database only).

1. Copy sharedprefix.MICS.ISPTLIB to the monthly ISPTLIB backup file.

Step IDCAMS: Produce CA MICS Catalog List.

1. Generate Catalog List for all 'prefix.MICS' entries.

4.3.3.3.4 YEARLY Job Steps

The actions being performed within each step of the YEARLY job are discussed below.

Step YEARnnn: Update information areas in the database for products installed in this unit database. There is one step (001-190) for each product.

1. Verify that prerequisite step completed successfully.
2. Delete each file's highest defined cycle.
3. Split year-to-date cycles (00) into current year (Y0) and prior years (Y1) cycles.
4. Perform yearly cycle aging for each file.
   a. Rename cycles 01-nn to cycles 02-(nn+1).
   b. Replace cycle 00 with cycle Y0 (new year-to-date).
   c. Rename cycle Y1 to 01.
5. Update the Checkpoint File -- MONTHnnn completed.

Step YEAR199: Accounting and Chargeback -- see the Accounting and Chargeback User Guide for information on monthly accounting processing.

Step YEAR500: User-written reports (non-MICF).

1. Verify that prerequisite step completed successfully.
2. Perform user-defined processing -- see prefix.MICS.USER.SOURCE(YRUSER).
3. Update the Checkpoint File -- YEAR500 completed.

Step YEAR900: Database update termination.

1. Verify that prerequisite step completed successfully.
2. Update the Checkpoint File -- YEAR900 completed.

Step YEARRSR: Run Status Report.

1. Generate CA MICS Run Status Report.
2. Update Operational Status and Tracking control tables.
   a. Store YEARLY job status.
   b. Update the processing schedule for the next 7 days.
   c. Update the operational job log.

4.3.3.3.5 BACKUP Job Steps

The actions being performed within each step of the BACKUP job are discussed below.

Step BKUP900: Database backup.

1. Verify that prerequisite step completed successfully.
2. Update the Checkpoint File -- BKUP900 in progress.
3. Back up the Checkpoint File.
4. Create and save the CMPSFILE macro after the checkpoint data. (CMPSFILE contains a list of all files present in the database that are compressed.)
5. Copy online Database files to timespan backup data sets.
6. Update Operational Status and Tracking control tables.
   a. Store BACKUP job status.
   b. Update the backup log.
7. Update the Checkpoint File -- BKUP900 completed.

Step BKTLIB: Backup sharedprefix.MICS.ISPTLIB (PRIMARY unit Database only).

1. Copy sharedprefix.MICS.ISPTLIB to the ISPTLIB backup file.

Step BKUPRSR: Run Status Report.

1. Generate CA MICS Run Status Report
2. Update Operational Status and Tracking control tables.
   a. Store BACKUP job status.
   b. Update the processing schedule for the next 7 days.
   c. Update the operational job log.

Step IDCAMS: Produce CA MICS Catalog List.

1. Generate Catalog List for all 'prefix.MICS' entries.

4.3.3.3.6 RESTORE Job Steps

The actions being performed within each step of the RESTORE job are discussed below.

If you specified RESTORE BACKUP YES in prefix.MICS.PARMS(JCLDEF), the RESTORE job begins by executing a Database backup as described below.

Step BKUP900: Database backup.

1. Verify that prerequisite step completed successfully.
2. Update the Checkpoint File -- BKUP900 in progress.
3. Backup the Checkpoint File
4. Copy online database files to timespan backup data sets.

Step BKTLIB: Backup sharedprefix.MICS.ISPTLIB (PRIMARY unit database only).

1. Copy sharedprefix.MICS.ISPTLIB to the ISPTLIB backup file.

Step BKUPRSR: Run Status Report.

1. Generate CA MICS Run Status Report

Step IDCAMS: Produce CA MICS Catalog List.

1. Generate Catalog List for all 'prefix.MICS' entries.

RESTORE processing follows the backup. The RESTORE steps are described below.

If you specified RESTORE BACKUP NO in prefix.MICS.PARMS member JCLDEF, the RESTORE job only executes the following steps.

Step RSTR900: Database Restore.

1. Print current Checkpoint File contents.
2. Update the Checkpoint File -- RSTR900 in progress.
3. Print the backup Database Checkpoint File.
4. Perform Checkpoint File restore.
5. Update the restored Checkpoint File -- RSTR900 is still in progress.
6. Perform PROC DATASETS KILL on each timespan to delete the existing online database files.
7. Copy timespan backup data sets to online Database

files. Uncompressed files are restored, and if you have compressed files, they are restored in a second pass of the tape.

8. If incremental update is active for any product in the unit database, examine the product's incremental update checkpoint file and issue messages warning that running IUDBINIT may be required to re-initialize in-progress or completed incremental update processing prior to resuming database update processing.
9. Update Operational Status and Tracking control tables.
   a. Store RESTORE job status.
   b. Update the operational job log.
10. Update the Checkpoint File--RSTR900 completed.

Step RSTRRSR: Run Status Report.

1. Generate CA MICS Run Status Report.
2. Update Operational Status and Tracking control tables.
   a. Store RESTORE job status and update status for all operational jobs.
   b. Update the processing schedule for the next 7 days.
   c. Update the operational job log.

4.3.3.3.7 INCRccc Job Steps

The actions being performed within each of the steps in the INCRccc jobs are discussed below.

Step INCRnnn: Update information areas in the incremental update database for the ccc product. There is one step corresponding to the product's DAYnnn (001-190) step.

1. Verify that prerequisite steps completed successfully.
   a. Previous DAY900 step
   b. Previous INCRnnn step (if any)
2. If the INCRSPLIT USE option is set for this product, dynamically allocate the input SMF data file (prefix.MICS.ccc.IUSPLTDS) created by the SPLITSMF job.
3. If this is the first INCRnnn execution since the last DAILY job completed,
   a. Initialize the incremental update checkpoint file from the unit checkpoint.
   b. Initialize incremental update database suspend files for the ccc product (if any) from the CA MICS database DETAIL and/or DAYS timespans.
4. Read raw input data, convert to SAS format, and output to intermediate work files.
5. Sort, analyze, summarize, and enhance intermediate work file contents to create incremental update DETAIL cycle 00 files.
6. Build incremental update DAYS cycle 00 files.
7. Combine the new 00 cycle files with 01 cycle files (if any) from the prior INCRnnn execution to create the updated incremental update DETAIL and DAYS cycle I0 files.
8. Update time ranges in the SYSID/COMPONENT section of the incremental update checkpoint file and input history log.
9. Perform incremental update file aging.
   a. Rename incremental update DETAIL cycle I0 to 01.
   b. Rename incremental update DAYS cycle I0 to 01.
10. Update the incremental update checkpoint file-- INCRnnn completed.
11. If the INCRSPLIT USE option is set for this product, delete the prefix.MICS.ccc.IUSPLTDS data set that was used as input by this INCRccc execution.

4.3.4 Operational Status and Tracking

Operational Status and Tracking provides online access to the CA MICS operational jobs and to CA MICS Database status. You can use Operational Status and Tracking to:

- Interrogate Database status and/or processing history.

- Submit regularly scheduled production CA MICS processing.

- Perform special CA MICS processing, including Database recovery and production job restart.

- Suspend Database update processing while you apply maintenance.

Note: Operational Status and Tracking does not support CA MICS incremental update facilities. This means that you can NOT submit INCRnnn jobs through Operational Status and Tracking facilities, and the Operational Status and Tracking commands do NOT support the incremental update checkpoint files.

However, FORCE and SELECT specifications established through the Operational Status and Tracking FORCE command ALSO APPLY to Incremental Update SPLITSMF and INCRccc job processing.

Operational Status and Tracking presents a scrollable status display of the unit databases in your CA MICS complex. You can scroll RIGHT to display the CA MICS processing schedule and Operational Status and Tracking parameters.

Online tutorials provide detailed documentation on display content, commands, usage considerations, and data entry instructions.

The following sections document:

1 - Operational Status and Tracking Display
2 - Database Status
3 - CA MICS Operations
4 - Operational Status and Tracking Commands

4.3.4.1 Operational Status and Tracking Display

```
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .


Command ===>                                                  Scroll ===> CSR



Commands:  Schedule, Daily, Weekly, Monthly, Yearly, Backup, Restore, Restart,
           Status/History/Checkpt/Joblog, Suspend/Resume, Force

           Database       Current Operation     Last Completed    Edit Suspend
  Cmd    ID  Label Type   as of 10 OCT 2001     Job/Step  Date    JCL  Updates
-------- -- -------- -   -------------------  ------------------  ---    ---
________ C  CICS     U   MONTHLY  Completed   MONTHLY  10OCT2001  NO     NO
________ D  DASD     U   MONTHLY  DUE TODAY   WEEKLY   09OCT2001  NO     NO
________ I  IMS      U   MONTHLY  Completed   MONTHLY  10OCT2001  NO     NO
________ P  PRIMARY  P   MONTHLY  FAILED      DAY020   10OCT2001  NO     NO
________ R  REMOTE   U   MONTHLY  FAILED      DAYALL   10OCT2001  NO     NO
________ T  TEST     T   DAILY    OVERDUE     DAILY    19SEP2001  NO     NO
****************************** BOTTOM OF DATA  ********************************
```

. . . .

Operational Status and Tracking presents a scrollable display of the unit databases in your CA MICS complex. There is one line on the display for each unit database.

- o The primary display shows the current status of each unit database update.

- o Secondary displays, accessed by scrolling RIGHT, show the CA MICS processing schedule and Operational Status and Tracking parameters:

  - Use the RIGHT primary command to access the secondary displays. Scroll right once for the schedule forecast display and scroll right again for the parameter display.

  - Use the LEFT primary command to return to the primary (status) display. LEFT MAX returns immediately to the primary display.

Note: Operational Status and Tracking does not fully support CA MICS incremental update facilities. The JOBLOG command will report on INCRccc and SPLITSMF executions. However, the SCHEDULE command cannot be used to schedule the INCRccc and SPLITSMF jobs because their scheduling is dependent on the availability of the input data and on day-time batch processing considerations unique to your data center. You can use the primary and/or line commands for INCRUPDT and SPLITSMF to manually submit these jobs.

This section describes the content of the three Operational Status and Tracking displays.

- o Database Status Display
- o Schedule Forecast Display
- o Parameter Display

The Database Status, Schedule Forecast, and Parameter displays begin with the database heading: three columns of database identification.

- o ID - The CA MICS unit database (DB) ID you specified when you defined the unit database.
- o Label - The 1-8 character unit DB label you specified when you defined the unit database.
- o Type - The type of unit database.
  P is the PRIMARY database.
  U is a standard production UNIT database.
  T is a TEST database.
  S is a SPECIAL database.

DATABASE STATUS DISPLAY

The Database Status display presents information on the current operation, last completed processing, and the database update status. It is the default display when you access Operational Status and Tracking.

Current Operation as of - The date of the CA MICS database update process and status. (Two columns of information are displayed.)

- o The panel initially lists the processing scheduled for today. If you enter the DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, RESTORE, or RSTATUS commands, then Current Operation becomes the processing you requested. The display also reacts to processing scheduled outside of Operational Status and Tracking. Once database update processing begins, Current Operation is changed to reflect the processing that is in progress or completed.
- o Current operation includes:

  DAILY - Daily CA MICS database update.

  WEEKLY - Daily database update and weekly cycle closeout.

  MONTHLY - Daily database update and monthly cycle closeout.

  YEARLY - Daily database update and yearly cycle closeout.

  BACKUP - Database backup was explicitly requested or

```
                   failed.

         RESTORE - Database restore was explicitly requested
                   or failed.

         RSTATUS - Database run status processing was
                   explicitly requested.


    o Processing status includes:

         Completed  - Processing executed today has completed
                      successfully.

         DUE TODAY  - Scheduled processing has not yet
                      occurred.

         OVERDUE    - Scheduled processing is late.

                      Today's processing is OVERDUE if not
                      complete by the user-specified deadline
                      (hh:mm).  The default is 8 a.m.

         Submitted  - Scheduled or requested processing has
                      been submitted for execution but is not
                      yet complete.

         INCOMPLETE - Database update processing began
                      execution but has not finished.

                      If the job is NOT still executing, then
                      it may have been canceled by the
                      operator, failed with a JCL error, or
                      terminated by an MVS system crash.
                      RESTART may be required.

         FAILED     - Scheduled or requested processing
                      terminated abnormally.

                      Use the STATUS command for more
                      information.  Use the RESTART command to
                      restart database update processing.

                      You can enter the BACKUP or RESTORE
                      command to restart backup/restore
                      processing that failed.

  Last Completed - The last CA MICS process completed for this
                   unit database.  Two columns of information
```

are displayed.

- o Job/Step - The last production job (DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, or RESTORE) completed if no errors occurred.

  If database update processing failed or is still in progress, the last completed step name is displayed.

  For example, if daily processing failed updating the SMF product files, then DAY020 would appear to show that CA MICS completed the DAY020 (update RMF product files) step. In this situation, restart would be at the DAY030 step.

  If BACKUP or RESTORE processing failed, Job/Step is not changed. That is, Last Completed still reflects the last completed processing before BACKUP or RESTORE began.

- o Date - The date that Job/Step completed.

Edit JCL - Use this column to specify whether you want to edit generated production job JCL before the job is submitted for processing.

Suspend Updates - The database update status.

YES - Database update processing has been suspended.

This generally means that maintenance is being applied. CA MICS will NOT allow you to execute any database update processing.

Enter the RESUME command to activate database update processing.

NO - Database update processing is available. This is the normal status.

Enter the SUSPEND command to block database update processing.

The Operational Status and Tracking display is refreshed when you execute any Operational Status and Tracking command. Use the REFRESH command to update the display without invoking other functions.

```
SCHEDULE FORECAST DISPLAY

----------------------  Operational Status and Tracking  ---------  ROW 1 OF 6


Command ===>                                                Scroll ===> CSR



Commands:  Schedule, Daily, Weekly, Monthly, Yearly, Backup, Restore, Restart,
           Status/History/Checkpt/Joblog, Suspend/Resume, Force


           Database
  Cmd    ID  Label Type              Seven Day Schedule Forecast
-------- -- -------- -   ---------------------------------------------------
________ C  CICS     U   10OCT2001 D.M.B D...B D...B D...B D...B D...B D...B
________ D  DASD     U   09OCT2001 DW..B D.M.B D...B D...B D...B D...B D...B
________ I  IMS      U   10OCT2001 D.M.B D...B D...B D...B D...B D...B D...B
________ P  PRIMARY  P   10OCT2001 D.M.B D...B D...B D...B D...B D...B D...B
________ R  REMOTE   P   10OCT2001 D.M.B D...B D...B D...B D...B D...B D...B
________ T  TEST     T   19SEP2001 D...B D.... D...B D.... D...B DW..B D...B
****************************** BOTTOM OF DATA  ********************************
```

. . . .

Access the Seven Day Schedule Forecast by scrolling RIGHT from the Database Status display.

This panel displays the CA MICS processing schedule for a seven day period. The first column is the schedule beginning date. This is the date of the first schedule forecast entry.

Seven schedule forecast entries follow the schedule beginning date. Each forecast entry has the format dwmyb for DAILY, WEEKLY, MONTHLY, YEARLY, and BACKUP, respectively.

The schedule forecast entries indicate which CA MICS process is scheduled for a specific day (the beginning date and the next six days): DAILY, DAILY/BACKUP, DAILY/WEEKLY/BACKUP, and so on. A period (.) indicates that the corresponding job is NOT scheduled.

In the example above, the following processing is scheduled for the D (DASD) unit:

- October 9 - WEEKLY process (DAILY/WEEKLY/BACKUP)
- October 10 - MONTHLY process (DAILY/MONTHLY/BACKUP)
- October 11 through 15 - DAILY process (DAILY/BACKUP)

PARAMETER DISPLAY

```
----------------------  Operational Status and Tracking  ---------  ROW 1 OF 6


Command ===>                                                     Scroll ===> CSR




Commands:  Schedule, Daily, Weekly, Monthly, Yearly, Backup, Restore, Restart,
           Status/History/Checkpt/Joblog, Suspend/Resume, Force

           Database      Overdue
  Cmd    ID Label   Type (hh mm)
-------- -- -------- -    -- --
________ C  CICS     U    08 00
________ D  DASD     U    08 00
________ I  IMS      U    08 00
________ P  PRIMARY  P    08 00
________ R  REMOTE   P    08 00
________ T  TEST     T    08 00
****************************** BOTTOM OF DATA  ********************************



.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Access the Parameter display by scrolling RIGHT from the Schedule Forecast display.

This panel lists the overdue time for each unit database. Use the Overdue column to override the 08:00 default daily CA MICS database update processing deadline.

- Daily processing is considered late and status is set to OVERDUE if not completed before this user-specified time-of-day.

- Status is DUE TODAY until the overdue time.

4.3.4.2 Database Status

The Operational Status and Tracking display shows you the current status of each unit database.

- The Current Operation column shows the processing scheduled for (or executed) today and the status of this processing -- Completed, DUE TODAY, OVERDUE, Submitted, INCOMPLETE, or FAILED.

- The Last Completed column shows the last process that executed and the completion date, or if the last update job failed, the last completed job step appears.

You can use Operational Status and Tracking commands to view more detailed Database status information. Four commands -- STATUS, HISTORY, CHECKPT, and JOBLOG -- are provided.

STATUS displays more detailed status information.

- The STATUS panel displays specific indicators of Database update and cycle aging status.

- The last step completed and completion date is shown for each of the DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, and RESTORE jobs.

- Jobs scheduled for today (e.g., DAILY, WEEKLY, and BACKUP) are listed separate from the other jobs.

  Note: The STATUS display does NOT reflect the incremental update (SPLITSMF or INCRccc job) processing

HISTORY lets you review CA MICS Database update history.

- The HISTORY display lists run date and checkpoint timestamps for each CA MICS product that updates the CA MICS Database.

- The HISTORY display is initially subset to the most recent update for each product. You enter SHOW ALL on the HISTORY display to view all previous Database updates (up to the INPUTLOGLIMIT parameter you specified in prefix.MICS.PARMS(EXECDEF)).

- You can select a row to see more detail, such as the number of input records processed, number of records deleted, and force/select specifications in effect during the Database update.

Note: The HISTORY display does NOT reflect incremental update (INCRccc job) processing for the current day. However, once the DAILY job completes, the HISTORY display will contain entries for EACH incremental update.

CHECKPT displays the CA MICS Checkpoint File Database update time records. You can use this option to review the date range processed to date by each CA MICS product and application system (e.g., SYSID and IMSID).

Note: The HISTORY display does NOT reflect incremental update (INCRccc job) processing for the current day. However, once the DAILY job completes, the HISTORY display will contain entries for EACH incremental update.

JOBLOG displays an audit log of your Database update, BACKUP, and RESTORE processing.

o Both the standard and special backups taken for this CA MICS unit database are listed.

o Unit database update processing is listed.

o Operational jobs submitted by Operational Status and Tracking are listed.

o Checkpoint changes made by Operational Status and Tracking are also listed.

Note: The JOBLOG display does NOT reflect incremental update (INCRccc job) processing.

See the Operational Status and Tracking Commands section (4.3.4.4) for more information on each command.

4.3.4.3 CA MICS Operations

CA MICS maintains a processing schedule based on the specifications you entered in sharedprefix.MICS.PARMS (defaults for all units) and in prefix.MICS.PARMS for each unit database. You specified backup frequency in prefix.MICS.PARMS(JCLDEF) and indicated in prefix.MICS.PARMS(EXECDEF) whether WEEKLY, MONTHLY, and/or YEARLY processing can occur during weekends. You also defined your installation's calendar (e.g., a 13 month fiscal calendar) and week (e.g., the week starts on Sunday vs. Monday) through the sharedprefix.MICS.PARMS(CPLXDEF) or prefix.MICS.PARMS(SITE) parameter members.

You can simply enter the SCHEDULE command to execute the CA MICS processing scheduled for today (see the Current Operation column). You can scroll RIGHT on the Operational Status and Tracking display to see the seven day CA MICS processing schedule forecast.

You can also schedule CA MICS processing with specific commands for the DAILY, WEEKLY, MONTHLY, and YEARLY processing cycles. Each of these options performs a DAILY update and a database BACKUP. The WEEKLY/MONTHLY/YEARLY options follow the DAILY update with the requested WEEKLY/MONTHLY/YEARLY close-out cycle and then the BACKUP.

It is always a good idea to back up the database before making any changes. You can enter the BACKUP command for a special, ad hoc database backup. You will be prompted to provide a brief explanation of the purpose of the backup so that CA MICS can label the backup cycle for reference should you need to RESTORE.

Occasionally, problems occur in CA MICS processing. Operational Status and Tracking presents information on database update status and provides tools for restart and recovery. Use the RESTART command to restart CA MICS processing. If the failure occurred during the CA MICS aging process, call CA MICS Product Support.

There will also be times where you will want to block database update processing. For example, you may need to suspend Database updating while you apply selected PSP maintenance. Operational Status and Tracking provides the SUSPEND command to block database update processing until you are ready to continue with normal CA MICS operations. The RESUME command returns a unit database to a normal operational state.

Operational Status and Tracking also simplifies special recovery operations. The FORCE command tells CA MICS to select specific input data and/or accept input data which would normally fail CA MICS date/time checks. For example, if a tape of SMF data was missed, you can FORCE this missing data into the database even though it pre-dates the most recent CA MICS checkpoint dates. You can even select the missing data from a composite (e.g., week-to-date) input data tape.

See the Operational Status and Tracking Commands section (4.3.4.4) for more information on each command.

4.3.4.4 Operational Status and Tracking Commands

You will use Operational Status and Tracking commands to request CA MICS operations and processing status services.

You can enter Operational Status and Tracking commands as line commands or as primary commands.

- Line commands are entered in the Cmd column and apply ONLY to the corresponding CA MICS unit database.

- Primary commands are entered in the Command area at the top of the panel and can apply to one or more unit databases, with the exception of the INCRUPDT and SPLITSMF commands. The INCRUPDT and SPLITSMF commands require the component as well as the database ID and therefore only support one component/database ID combination per execution of the command.

Detailed online tutorials document each command and the panels the command displays. The online tutorials describe display content, usage considerations, and data entry instructions.

Each command refreshes the Operational Status and Tracking display.

You can also use standard CA MICS commands to subset the Operational Status and Tracking display (SHOW command), find a specific value in the display (FIND command), or position to a row of the display (LOCATE). See the online tutorials for more information on primary command considerations and syntax.

The Operational Status and Tracking commands are documented in the sections that follow. Command syntax is documented below.

COMMAND SYNTAX

Enter Operational Status and Tracking commands in the Cmd column (line commands) to request processing for a specific unit database. There are no line command parameters; simply enter the command name.

Enter Operational Status and Tracking commands in the Command area as follows:

```
command UNIT(S) dbidlist/ALL BACKUP/NOBACKUP EDIT/NOEDIT

where:

  o command  - The Operational Status and Tracking command
               name (for example, SCHEDULE).

  o UNIT(S)  - Optional keyword UNIT or UNITS to indicate
               that you are specifying one or more CA MICS
               unit database (DB) IDs for the command.

  o dbidlist - One or more CA MICS unit database (DB) IDs
               separated by blanks.  Command will be
               performed for each DB ID in the list.

    or

  o ALL      - Specify ALL instead of dbidlist if you want to
               perform the command for each production unit
               database.  ALL does NOT apply to TEST or
               SPECIAL unit databases.

               For example, enter SCHEDULE ALL to submit
               today's scheduled processing for all
               production unit databases.

               ALL will honor the display subset you define
               with the SHOW command.  For example, if
               several unit database updates failed
               overnight, you can use ALL to RESTART each of
               the failing processes.
                - Enter SHOW FAILED to subset the display.
                - Enter RESTART ALL to restart processing for
                  just the unit database updates that
                  failed.

  o BACKUP/NOBACKUP - Optional parameter for the SCHEDULE,
                      DAILY, WEEKLY, MONTHLY, and YEARLY
                      commands.

               Specify BACKUP with the SCHEDULE command to
               override the processing schedule and include a
               backup in the requested processing, even if no
               backup is currently scheduled.

               Specify NOBACKUP to omit backup processing
               even if normally included in the requested
               processing.  DAILY, WEEKLY, MONTHLY, and
               YEARLY will always include a backup unless you
```

specify NOBACKUP.

You can also specify NOBACKUP on the RESTORE command to override your prefix.MICS.PARMS(JCLDEF) specification of RESTORE BACKUP YES. CA MICS will restore the database without a pre-restore unit database backup.

Note: You cannot override a RESTORE BACKUP NO specification in JCLDEF. Do not specify BACKUP on the RESTORE command.

o EDIT/NOEDIT - Optional parameter for the SCHEDULE, DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, RESTORE, RESTART, and RSTATUS commands.

Specify EDIT to override your specification in the Edit JCL column and display each generated job under PDF Edit before submitting the job for batch execution.

Specify NOEDIT to submit generated jobs without a pre-submit Edit, even if you specified YES in the Edit JCL column.

The operational status and tracking commands are documented in the sections that follow:

 1 - BACKUP Command
 2 - CHECKPT Command
 3 - DAILY Command
 4 - FORCE Command
 5 - HISTORY Command
 6 - JOBLOG Command
 7 - MONTHLY Command
 8 - REFRESH Command
 9 - RESTART Command
10 - RESTORE Command
11 - RESUME Command
12 - RSTATUS Command
13 - SCHEDULE Command
14 - STATUS Command
15 - SUSPEND Command
16 - WEEKLY Command
17 - YEARLY Command
18 - INCRUPDT Command
19 - SPLITSMF Command
20 - AUDIT Command

```
21 - HISTM Command
22 - HISTW Command
```

4.3.4.4.1 BACKUP Command

```
--------------------------  BACKUP Unit Database  ---------------------------
Command ===>

Database:  C  (CICS) - CA MICS CICS DATABASE


Enter a brief description of this backup job to help identify it in a
subsequent Database RESTORE command.

===> Backup before applying CICS upgrade on 10/10


Edit the job stream before batch submit ===> YES (YES/NO)





Press END (or enter the END command) to generate and submit the BACKUP job.
Enter CANCEL to terminate BACKUP processing for this unit Database.
```

. . . .

Use the BACKUP command to take a backup of unit database contents.

BACKUP generates and submits a batch job to back up a unit database. BACKUP also saves a copy of the generated batch job in the unit's prefix.MICS.RESTART.CNTL data set.

The BACKUP Unit Database panel is displayed for the BACKUP command. Use the BACKUP Unit Database panel to document the backup you are requesting -- the description you enter will be displayed by the RESTORE command to help you select the proper backup generation should you need to recover the CA MICS Database.

CA MICS automatically labels standard daily and monthly backups. You should always take a special Database backup before making changes that might impact Database integrity. (This includes applying PSP maintenance.) By entering meaningful text on this panel, you will be able to easily distinguish the standard and special backup generations available should you need to RESTORE the Database.

The BACKUP Unit Database panel also gives you the option to review and/or modify the generated batch job before it is submitted for execution. You can also cancel job submission processing after reviewing the generated JCL.

See the online tutorial for detailed data entry instructions.

4.3.4.4.2 CHECKPT Command

```
---------------------------------  Checkpoint  -------------------------------
Command ===>                                                     Scroll ===> CSR

Database:  C  (CICS) - CA MICS CICS DATABASE

ORGSYSID  Comp  Subsys      Lowest Timestamp        Highest Timestamp
  ----    ---   ----       -------------------     -------------------
  ONL1    CIC   ACTR       20JUL87 08:32:05.50     10OCT01 00:00:00.13
  ONL1    CIC   MRP        15DEC88 21:00:32.03     10OCT01 00:00:01.21
  ONL1    CIC   PRD1       25SEP88 00:00:15.67     10OCT01 00:02:21.91
  ONL1    CIC   PRD2       25SEP88 00:00:13.00     10OCT01 00:02:32.11
  ONL1    CIC   PROD       07MAY84 00:00:02.21     25SEP98 15:23:51.34
  ONL1    CIC   TEST       07MAY84 00:07:42.03     09OCT01 21:00:00.52
****************************** BOTTOM OF DATA  ********************************
```

```
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Use the CHECKPT command to display the Database update time range records for the CA MICS unit database checkpoint file.

Note: The CHECKPT display does NOT reflect incremental update (INCRccc job) processing.

The rows of the Checkpoint panel describe each combination of original SYSID identifier (ORGSYSID), product identifier, and subsystem name in the Database and are listed in the same order they appear in the checkpoint file.

See the CA MICS Checkpoint File section (4.3.7) of this guide for more information on Checkpoint File contents and processing.

The scrollable Checkpoint panel displays:

```
  o ORGSYSID          - Original system identifier.

  o Comp              - Product (or component) identifier.
```

o Subsys - Subsystem name for products such as IMS and CICS where CA MICS product identification in the checkpoint is extended to include the application subsystem (e.g., IMSID/CICSID/etc.).

o Lowest Timestamp - the date/time of the first (oldest) data loaded into the Database for an ORGSYSID, product, and subsystem name.

o Highest Timestamp - the date/time of the most recent (newest) data loaded into the Database for an ORGSYSID, product, and subsystem name.

The CA MICS checkpoint date range check will drop input data for an ORGSYSID, product, and subsystem if the data is older (or less) than highest timestamp.

See the online tutorial for information on using CA MICS commands (e.g., FIND, LOCATE, SHOW, SORT, etc.) with the Checkpoint display.

4.3.4.4.3 DAILY Command

Use the DAILY command to explicitly request daily CA MICS Database update processing.

DAILY generates and submits a batch job to perform daily CA MICS Database update processing and a Database backup.

Specify the NOBACKUP parameter to delete Database backup processing from the generated batch job.

DAILY also saves a copy of the generated Database update job in the prefix.MICS.RESTART.CNTL data set for the unit Database.

4.3.4.4.4 FORCE Command

```
--------------------  FORCE/SELECT Input Data Definitions  --------------------
Command ===>

Unit:  C  (CICS) - CA MICS CICS DATABASE

Enter FORCE/SELECT date/time ranges that will apply to the indicated systems:

                    Lowest Timestamp           Highest Timestamp
                 -----------------------    -----------------------
                 dd mon yy   hh:mm:ss.th    dd mon yy   hh:mm:ss.th
     Force  ===> 09 OCT 01   15 33 00.00    09 OCT 01   18 45 59.99
     Select ===> 09 OCT 01   00 00 00.00    09 OCT 01   23 59 59.99

Enter system, product, and subsystem combinations (or * for ALL) to apply
FORCE/SELECT date/time ranges or to ERASE prior FORCE/SELECT specifications.

     ORGSYSID     ===> *    ____ ____ ____ ____ ____ ____ ____ ____ ____
     Product ID   ===> CIC  ___  ___  ___  ___  ___  ___  ___  ___  ___
     Subsystem ID ===> P*   ACTR ____ ____ ____ ____ ____ ____ ____ ____

Press END to apply and review FORCE/SELECT date/time ranges.
Enter ERASE to remove FORCE/SELECT specifications from the indicated entries.
Enter CANCEL to exit without changing the unit Database checkpoint.
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

CA MICS Database protection features are designed to minimize the possibility of destroying data. A number of rigid input checks are required to achieve this protection. However, there are times when the Administrator needs to safely bypass these protection features.

Under some circumstances, it may be necessary to subset the input data being used to update the Database. For example, the CA MICS system may have to use a cumulative input tape which has more days data than can be run in in a single update. You can use the FORCE command to restrict CA MICS processing to a subset of the input data. That is, you have the option to select a portion of a cumulative input file as input to the Database.

Under extraordinary circumstances, it may be necessary to deactivate the Database time range data check in order to input data that may have been previously omitted. For example, a portion of the CA MICS input data may have been unavailable for the standard daily update or CA MICS might not have been submitted for a day, but was run on the following day. Because the missing data was within the timespan of the data already used to update the Database, it would no longer be allowed to pass the standard time-range tests on a succeeding update. Although great care should be taken when overriding the standard CA MICS protection features, you can use the FORCE command to force CA MICS to accept data already encompassed by the Database's current time-ranges.

You can also use the FORCE command to review force/select criteria you entered earlier. You can use the ERASE command on the FORCE/SELECT Input Data Definitions panel to clear previous force/select specifications. In either case, force/select specifications are automatically cleared (erased) when daily Database update processing updates the unit database checkpoint file.

NOTE: Specifications you make through the FORCE command apply to BOTH the DAILY database update and ALSO to the incremental update INCRccc job for the associated product. FORCE and SELECT specifications are saved in the unit checkpoint data set and will remain in effect until cleared through successful DAILY or INCRccc job execution.

Be aware that the INCRccc job allocates the unit checkpoint data set DISP=SHR and uses CA MICS ENQ/DEQ services to lock the unit checkpoint while updates are being made. If the unit checkpoint is locked by another task, the INCRccc job will wait until the ENQ is obtained. If the maximum wait limit is exceeded (see the JCLDEF DYNAMWAIT parameter), then the INCRccc job will abort. In this situation, you should manually clear the FORCE/SELECT specifications to prevent their reuse by the next INCRccc job or DAYnnn step for this product.

FORCE/SELECT INPUT DATA DEFINITIONS PANEL

The FORCE/SELECT Input Data Definitions panel is displayed for the FORCE command.

You will use the FORCE/SELECT Input Data Definitions panel to specify force and/or select timestamps to be applied to the CA MICS checkpoint entries for the system, product, and subsystem combinations you identify. When you press END, CA MICS will apply your specifications to the unit database checkpoint file and display the Checkpoint Data Set Time Range Records panel.

You can review the current unit database checkpoint data set force/select specifications by pressing END before entering data on this panel. That is, if you leave ALL fields blank, then the Checkpoint Data Set Time Range Records panel displays the current unit database checkpoint file. No new FORCE/SELECT date/time ranges are added.

If you enter the ERASE command, then CA MICS will clear force/select time ranges from the checkpoint entries that match your system, product, and subsystem combinations.

FORCE/SELECT date/time ranges consist of a low (or start) timestamp and a high (or stop/end) timestamp in ddmonyy hh:mm:ss.th format (e.g., 09OCT01 15:30:00.00). Lowest and Highest Timestamp define an inclusive date/time range -- Lowest Timestamp through Highest timestamp.

```
    Force  - Date/time range (Lowest Timestamp through
             Highest Timestamp) of data to be forced into the
             unit database even though this data is
             encompassed by the Database's current time-
             ranges.  If you also specify Select, then the
             Force date/time range is applied ONLY to data
             within the Select date/time range.

    Select - Date/time range (Lowest Timestamp through
             Highest Timestamp) of data to be processed by
             the CA MICS Database update. Only data that
             falls within the Select date/time range is made
             available to CA MICS.
```

System, product, and subsystem combinations identify the unit Database checkpoint entries to which the Force/Select date/time ranges will apply or from which Force/Select specifications will be erased.

- o Each value you enter for ORGSYSID, Product ID, or Subsystem ID must participate in at least one successful match with the unit database checkpoint entries.

  A successful match will include at least one combination of ORGSYSID, Product ID, and Subsystem ID. That is, each ORGSYSID value must be found on at least one checkpoint entry that contains one of the Product ID and Subsystem ID values you specify.

- o You can specify * (asterisk) to select all systems, products, or subsystems. That is, * means ALL.

You can also terminate an ORGSYSID, Product ID, or Subsystem ID with an * to select all entries that begin with the characters you specify. For example, PRD* would match all entries beginning with the characters PRD.

Refer to the online tutorials for detailed data entry instructions.

The example panel above defines specifications that will,

- o Select input data for October 9, 2001.

- o Force October 9th data from 15:33 (3:33 p.m.) through 18:45:59.99 (that is up to but not including 6:46 p.m.) into the database.

The Force/Select criteria will apply to CICS product processing for the ACTR and Pxxx CICSID's (subsystems), where xxx is any 3 characters. Force/Select processing applies to all processors (ORGSYSID values).

The Checkpoint Data Set Time Range Records panel is displayed when you press END on the FORCE/SELECT Input Data Definitions panel. The sample panel on the next page shows the result of applying the previous example to the Checkpoint File.

```
-------------------  Checkpoint Data Set Time Range Records  ------------------
Command ===>

Unit:  C  (CICS) - CA MICS CICS DATABASE

Line Cmds:  C Copy, M Move, O Over, E Erase

Cmd  Sys  CCC Subsys             Lowest Timestamp         Highest Timestamp
 -   ---- --- ----            ----------------------   ----------------------
 _   ONL1 CIC ACTR            20 JUL 87  08 32 05.50   10 OCT 01 00 00 00.13
                  Force  ===> 09 OCT 01  15 33 00.00   09 OCT 01 18 45 59.99
                  Select ===> 09 OCT 01  00 00 00.00   09 OCT 01 23 59 59.99
 _   ONL1 CIC MRP             15 DEC 88  21 00 32.03   10 OCT 01 00 00 01.21
                  Force  ===> __ ___ __  __ __ _____   __ ___ __ __ __ _____
                  Select ===> __ ___ __  __ __ _____   __ ___ __ __ __ _____
 _   ONL1 CIC PRD1            25 SEP 88  00 00 15.67   10 OCT 01 00 02 21.91
                  Force  ===> 09 OCT 01  15 33 00.00   09 OCT 01 18 45 59.99
                  Select ===> 09 OCT 01  00 00 00.00   09 OCT 01 23 59 59.99
 _   ONL1 CIC PRD2            25 SEP 88  00 00 13.00   10 OCT 01 00 02 32.11
                  Force  ===> 09 OCT 01  15 33 00.00   09 OCT 01 18 45 59.99
                  Select ===> 09 OCT 01  00 00 00.00   09 OCT 01 23 59 59.99
 _   ONL1 CIC TEST            17 JUN 87  00 12 32.13   10 OCT 01 00 02 15.55
```

```
                  Force  ===> __ ___ __  __ __ _____   __ ___ __ __ __ _____
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

The Checkpoint Data Set Time Range Records panel is displayed when you press END on the FORCE/SELECT Input Data Definitions panel. This panel shows the result of applying your Force/Select specifications to the current CA MICS unit database checkpoint file.

The first line of each row of this scrollable panel shows:

- ORGSYSID - Original system identifier.

- Comp - Product (or component) identifier.

- Subsys - Subsystem name for products such as IMS and CICS where CA MICS product identification in the checkpoint is extended to include the application subsystem (e.g., IMSID/CICSID/etc.).

- Lowest Timestamp - Date/time of the first (oldest) data loaded into the Database for an ORGSYSID, product, and subsystem name.

- Highest Timestamp - Date/time of the most recent (newest) data loaded into the Database for an ORGSYSID, product, and subsystem name.

  The CA MICS checkpoint date range check will drop input data for an ORGSYSID, product, and subsystem if the data is older (or less) than highest timestamp.

The second and third lines of each row show the Force/Select date/time range specified for the ORGSYSID, product, and subsystem. The Force and Select values are the result of the Force, Select, ORGSYSID, Product ID, and/or Subsystem ID values (and/or the ERASE command) you specified on the FORCE/SELECT Input Data Definitions panel combined with the current unit database checkpoint file contents.

- You can change Force/Select Lowest Timestamp and/or Highest Timestamp.

- You can use the E (Erase) line command to clear both the Force and Select values for an entry.

- You can also use the C (Copy) and M (Move) line commands to modify Force/Select specifications.

  - Copy replicates Force/Select specifications onto the row you flag with the O (Over) line command.

  - Move replicates Force/Select specifications onto the O (Over) line and then clears them from the M (Move) line -- that is, Move is like entering Copy and then Erase.

When you are satisfied with your Force and/or Select specifications, press END to update the unit database checkpoint file.

Press CANCEL to terminate FORCE command processing without updating the unit database checkpoint file.

Refer to the online tutorials for detailed data entry instructions and for information on using CA MICS commands (e.g., FIND, LOCATE, SHOW, SORT, etc.) on this panel.

4.3.4.4.5 HISTORY Command

```
-----------------------------  Input History Log  -----------------------------
Command ===>                                                      Scroll ===> CSR


Database:  C  (CICS) - CA MICS CICS DATABASE

Line Cmds:  S Select

Cmd Sys CCC Subsys  Run Date       Lowest Timestamp       Highest Timestamp
 - ---- --- ----    ---------   ----------------------  ----------------------
 _ ONL1 CIC ACTR    10OCT2001   09OCT2001  00:00:12.19  10OCT2001  00:00:00.13
 _ ONL1 CIC MRP     10OCT2001   09OCT2001  00:00:02.47  10OCT2001  00:00:01.21
 _ ONL1 CIC PRD1    10OCT2001   09OCT2001  00:05:39.49  10OCT2001  00:02:21.91
 _ ONL1 CIC PRD2    10OCT2001   09OCT2001  00:04:12.53  10OCT2001  00:02:32.11
 _ ONL1 CIC TEST    10OCT2001   09OCT2001  06:15:00.41  09OCT2001  21:00:00.52
****************************** BOTTOM OF DATA  ********************************









 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

The HISTORY command displays the Database update audit trail, the CA MICS ADMIHL file.

The rows of the Input History Log panel describe the Database updates you have processed for each combination of original system identifier (ORGSYSID), product, and application subsystem.

Note: The HISTORY display does NOT reflect incremental update (INCRccc job) processing for the CURRENT day. However, once the DAILY job completes, the HISTORY display will contain entries for EACH incremental update (INCRccc job execution).

The display is initially subset to show the most recent Database update. Enter SHOW ALL to display the full history. The Input History Log display is limited by the INPUTLOGLIMIT parameter you specified in prefix.MICS.PARMS(EXECDEF).

The scrollable Input History Log panel displays:

```
o Sys               - Original system identifier.

o CCC               - Product (or component) identifier.

o Subsys            - Subsystem name for products such as
                      IMS and CICS where CA MICS product
                      identification in the checkpoint is
                      extended to include the application
                      subsystem (e.g., IMSID/CICSID/etc.).

o Run Date          - The Database update date.

o Lowest Timestamp  - Date/time of the first (oldest) data
                      loaded into the Database for an
                      ORGSYSID, product, subsystem name,
                      and run date.

o Highest Timestamp - Date/time of the most recent (newest)
                      data loaded into the Database for an
                      ORGSYSID, product, and subsystem
                      name, and run date.
```

See the online tutorial for information on using CA MICS commands (e.g., FIND, LOCATE, SHOW, SORT, etc.) with the Input History Log display.

You can select an entry from the display to see detailed information such as the number of input records processed; the number of records deleted due to date range, option values, short records, user exit, and total deleted; and the number of records kept. The Detail Input History Log (below) also lists any Force/Select criteria you specified for this Database update.

```
--------------------------  Detail Input History Log  -------------------------
Command ===>

Database:  C  (CICS) - CA MICS CICS DATABASE


ORGSYSID  Comp  Subsys        Run Timestamp         Routine
  ----    ---   ----      ---------------------     --------
  ONL1    CIC   MRP       10OCT2001:03:45:15.98     DYCICFMT

                                                          Range       Input
          Lowest Timestamp       Highest Timestamp     DAYS/HH:MM    Records
        ---------------------   ---------------------   -----------   --------
INPUT   09OCT2001 00:00:02.47   10OCT2001 00:00:01.21       0/23:59    1017791
FORCE            .           .            .           .       ./    .          0
SELECT           .           .            .           .       ./    .

Date checked deleted:           0
Option deleted:                 0
Short record deleted:          15
User exit deleted:           1024         Total records deleted:        1039
                                          Total records kept:        1016752

.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

The Detail Input History Log panel displays details for a single observation of the Database update audit trail -- the CA MICS ADMIHL file. You can use the UP, DOWN, FIND, or LOCATE commands to scroll to other entries.

The first line of the display identifies the entry by listing:

    ORGSYSID   - Original system identifier.

Comp - Product (or component) identifier.

Subsys - Subsystem name for products such as IMS and CICS where CA MICS product identification in the checkpoint is extended to include the application subsystem (e.g., IMSID/CICSID/etc.).

Run Timestamp - Database update date and time.

Routine - CA MICS processing routine that produced the entry.

The next section of the display describes the input data processed for this update cycle with three lines of information.

- o INPUT - The total amount of input data seen by CA MICS.
- o FORCE - Input data processed because of your FORCE specification.
- o SELECT - Input data subset by your SELECT specification.

The INPUT line describes the range of input data that was processed. The date/time range is shown by listing the oldest date/time found in the data (Lowest Timestamp), the most recent date/time found (Highest Timestamp), and the Range -- the number of days, hours, and minutes between the low and high timestamps). The total number of input records is also listed.

The FORCE line is normally blank. If you used the CA MICS checkpoint Force option to update the Database with data CA MICS would normally drop due to the checkpoint date range check, then the FORCE line will document that processing. Lowest and Highest Timestamps are the values you specified for Force processing and the number of records column shows the number of records you forced into the Database.

The SELECT line is normally blank. If you used the CA MICS checkpoint Select option to subset the input data for CA MICS processing, the SELECT line documents your specifications. Lowest and Highest Timestamps are the values you specified for SELECT processing and the number of records column shows the number of records found within the SELECT date/time range.

The bottom portion of the display documents the number of records actually added to the Database. The number of records deleted from CA MICS update processing are documented by delete reason indicated below.

- Date checked deleted - Duplicate data; date/time is within the date/time range that has already been processed.

- Option deleted - Deleted by the SELECT specification.

- Short record deleted - Invalid records; records are too short to contain valid data.

- User exit deleted - Deleted by user exit processing.

Total records deleted and total records kept (input records minus deleted records) are shown in the lower right-hand corner.

See the online tutorial for information on using CA MICS commands (e.g., FIND, LOCATE, SHOW, SORT, etc.) with the Detail Input History Log display.

4.3.4.4.6 JOBLOG Command

```
----------------------------  CA MICS Processing Log ----------------------------
Command ===>                                                    Scroll ===> CSR

Database:  C  (CICS) - CA MICS CICS DATABASE

    Date     Time                 Description                        Generation
  ---------  -----  ---------------------------------------------  ----------
  10OCT2001   4:41  BACKUP Completed (after Monthly)                BACKUP   0
  10OCT2001   4:09  MONTHLY BACKUP Completed                        MBACK    0
  10OCT2001   4:06  MONTHLY Completed
  10OCT2001   3:58  DAILY Completed (Monthly in progress)
  10OCT2001   0:01  DAILY, Monthly, Backup submitted by OPRUSR1
  09OCT2001   7:15  BACKUP Completed (after Weekly)                 BACKUP  -1
  09OCT2001   7:02  WEEKLY Completed
  09OCT2001   6:24  DAILY Completed (Weekly in progress)
  09OCT2001   4:25  DAILY, Weekly, Backup submitted by OPRUSR1
  09OCT2001   4:15  RESTORE from BACKUP 08OCT2001 03:34
  09OCT2001   3:42  DAILY FAILED U101 in DAY040
  09OCT2001   0:15  DAILY, Weekly, Backup submitted by OPRUSR1
  08OCT2001   3:34  BACKUP Completed (after Daily)                  BACKUP  -2
  08OCT2001   3:15  DAILY Completed
  07OCT2001   2:56  BACKUP Completed (after Daily)                  BACKUP  -3
  07OCT2001   2:32  DAILY Completed
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

JOBLOG displays the CA MICS processing audit log.

The rows of the CA MICS Processing Log panel describe your Database update, BACKUP, and RESTORE processing.

- o Both the standard and special backups taken for this CA MICS unit database are listed.

- o Database update processing is listed.

- o Operational jobs submitted by Operational Status and Tracking are listed.

- o Checkpoint changes made by Operational Status and Tracking are also listed.

Note: The JOBLOG display does NOT reflect incremental update (INCRccc job) processing.

The display is sorted with the most recent processing first and in the order that processing/changes occurred. The most recent Database and/or checkpoint updates appear first.

Backup generations are identified by the date and time the backup was taken, the generation number, and a description. Standard CA MICS daily and monthly backups are identified by name -- BACKUP (after Daily), MONTHLY BACKUP, etc. Special backup generations are identified by the description you entered on the BACKUP Unit Database panel.

BACKUP entries are dropped from the audit log when the corresponding tape GDG entry expires -- you specified BACKUP retention in prefix.MICS.PARMS(JCLGEN) with the BCKUPGDG parameter. Entries for Database update processing, job submission, and checkpoint changes that occurred prior to an expired backup are also dropped from the display. That is, once the backup tapes begin to expire (the BCKUPGDG limit is reached), the Database update processing, job submission, and checkpoint change entries older than the oldest backup entry are dropped.

CA MICS retains six monthly backup (MBACK) generations in the audit log.

Refer to the online tutorials for detailed data entry instructions and for information on using CA MICS commands (e.g., FIND, LOCATE, SHOW, SORT, etc.) on this panel.

Information reported by the example above includes:

- The MONTHLY process submitted by OPRUSR1 executed on October 10, 2001. DAILY, MONTHLY, Monthly backup (MBACK), and BACKUP processing completed successfully.

- The WEEKLY process was submitted by OPRUSR1 on October 9, 2001 using the Operational Status and Tracking facility.

  - The DAILY portion of this processing failed with a U101 abend in step DAY040.

  - The Database was restored and WEEKLY processing was resubmitted. WEEKLY completed successfully.

4.3.4.4.7 MONTHLY Command

Use the MONTHLY command to explicitly request monthly CA MICS Database update processing.

MONTHLY generates and submits a batch job to perform daily CA MICS Database update processing, monthly cycle Database close-out, and a Database backup.

Specify the NOBACKUP parameter to delete Database backup processing from the generated batch job.

MONTHLY also saves a copy of the generated Database update job in the prefix.MICS.RESTART.CNTL data set for the unit database.

4.3.4.4.8 REFRESH Command

Use the REFRESH command to update the Operational Status and Tracking display with the latest unit database processing status.

CA MICS updates Operational Status and Tracking entries before processing primary or line commands against a unit database. The REFRESH command is provided to give you the option of updating the display without visiting a subordinate panel. For example, both HISTORY and REFRESH update the Operational Status and Tracking display, but HISTORY also displays the Input History Log panel.

The REFRESH command defaults to the ALL option. That is, you can enter REFRESH ALL or just REFRESH in the Command area to update the Operational Status and Tracking entry for each production unit database.

Remember that ALL does NOT apply to TEST or SPECIAL unit databases and that ALL honors the display subset you define with the SHOW command. Enter REFRESH as a line command or use the primary command dbidlist option (e.g., enter REFRESH T S) to update Operational Status and Tracking entries for TEST and/or SPECIAL unit databases.

4.3.4.4.9 RESTART Command

```
--------------------------  RESTART Database Update  --------------------------

 Command ===>


Database:  P  (PRIMARY) - CA MICS PRIMARY DATABASE

The status information was recorded at:  10OCT01 08:10
The Database update job that failed:     DAILY
The job will be restarted at step:       DAY030


Edit the job stream before batch submit    ===> YES (YES/NO)
Rebuild DAYSMF step temporary work files   ===> NO  (YES/NO)







Press END (or enter the END command) to generate and submit the RESTART job.
Enter CANCEL to terminate RESTART processing for this unit Database.




.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Use the RESTART command to restart CA MICS database update processing after a processing failure.

RESTART generates and submits a batch job to restart database update processing that failed. RESTART also saves a copy of the generated batch job in the prefix.MICS.RESTART.CNTL data set for the unit database.

Note: The RESTART command does NOT support the incremental update INCRccc job. You must restart INCRccc processing through your installation's production batch job scheduling facility or manually submitting prefix.MICS.CNTL(INCRccc).

The RESTART Database Update panel displays the name of the CA MICS database update job that failed and the job step where CA MICS processing will resume. If internal step restart is enabled for this job step, then processing will automatically resume at the last completed processing phase in this job step. Review the information on this panel carefully.

If the restart job and/or step are incorrect -- that is, if the information on this panel does NOT match your expectations -- enter the CANCEL command to terminate RESTART processing. You should enter the STATUS command before invoking RESTART processing to review and analyze CA MICS operational job status.

The data entry fields on the Database Update panel will vary based on unit database contents and the failing job step.

- The Edit the job stream before batch submit option is always present.

- The Rebuild DAYSMF step temporary work files option appears if you specified DAYSMF FILES TEMPORARY in prefix.MICS.PARMS(JCLDEF) for this unit database AND if the failure occurred during DAILY processing.

  When you specify DAYSMF FILES TEMPORARY in JCLDEF, CA MICS deletes the DAYSMF work files after daily processing completes. The DAYSMF work files are kept when a failure occurs during daily processing and are normally available for RESTART processing.

  If the DAYSMF work files are accidentally scratched, the Operational Status and Tracking RESTART command can recreate the DAYSMF work files as part of RESTART processing. Operational Status and Tracking adds the prefix.MICS.CNTL(DAYSMFR) batch job to the front of the generated RESTART job stream.

- The Restore Accounting and Chargeback audit file option appears if the failure occurred during the DAY199 step -- that is, if Daily Accounting and Chargeback processing failed.

  The Accounting and Chargeback audit tapes provide a detailed audit of daily processing. Each daily update adds to the end of the ACTAUDIT DAY1 file (i.e., DISP=MOD). After daily processing, a new generation of the ACTAUDIT DAY2 tape is created from the DAY1 file. If you restart daily Accounting and Chargeback processing without restoring the ACTAUDIT DAY1 file, the audit files WILL contain duplicate data. See the Accounting and Chargeback User Guide for more information on the ACTAUDIT files.

Operational Status and Tracking adds the prefix.MICS.CNTL(ACTDAY1R) batch job to the front of the generated RESTART job stream.

The RESTART Database Update panel also gives you the option to review and/or modify the generated batch job before it is submitted for execution. You can cancel job submission processing after reviewing the generated JCL.

If you need to override automatic internal step restart and force the step to start from the beginning,

- o Specify YES to edit the job stream before execution.

- o When the job stream edit panel appears, scroll to the JCL EXEC statement for the job step where execution will resume and specify SYSPARM=NORESTART on this EXEC statement.

Refer to the online tutorials for detailed data entry instructions.

4.3.4.4.10 RESTORE Command

```
---------------------------  RESTORE Unit Database  -------------  ROW 1 OF 27
Command ===>                                                     Scroll ===> CSR

Database:  C  (CICS) - CA MICS CICS DATABASE

Edit the job stream before batch submit ===> YES (YES/NO)

Line Cmds:  S Select

Cmd    Date    Time  Generation                Description
 -   --------- ----- ----------  --------------------------------------------
 _   10OCT2001  4:41 BACKUP   0  BACKUP Completed (after Monthly)
 _   10OCT2001  4:09 MBACK    0  MONTHLY BACKUP Completed
 _   10OCT2001  4:06             MONTHLY Completed
 _   10OCT2001  3:58             DAILY Completed (Monthly in progress)
 _   10OCT2001  0:01             DAILY, Monthly, Backup submitted by OPRUSR1
 _   09OCT2001  7:15 BACKUP  -1  BACKUP Completed (after Weekly)
 _   09OCT2001  7:02             WEEKLY Completed
 _   09OCT2001  6:24             DAILY Completed (Weekly in progress)
 _   09OCT2001  4:25             DAILY, Weekly, Backup submitted by OPRUSR1
 _   09OCT2001  4:15          0  RESTORE from BACKUP 08OCT2001 02:46
 _   09OCT2001  3:42             DAILY FAILED U101 in DAY040
 _   09OCT2001  0:15             DAILY, Weekly, Backup submitted by OPRUSR1
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Use the RESTORE command to restore unit database contents from a selected unit database backup.

RESTORE generates and submits a batch job to restore unit database files. RESTORE also saves a copy of the generated batch job in the prefix.MICS.RESTART.CNTL data set for the unit database.

The RESTORE Unit Database panel displays an audit trail of your unit Database update, BACKUP, and RESTORE processing.

Note: The display does NOT reflect incremental update (INCRccc job) processing.

Select a BACKUP entry from this display to RESTORE the Database or enter CANCEL to terminate RESTORE processing.

If you specified YES for the RESTORE BACKUP parameter in the JCLDEF member of prefix.MICS.PARMS, CA MICS will take a unit database backup before beginning the restore. This option gives you the opportunity to return the Database to the condition that it was in prior to the restore.

- If you specify NOBACKUP when you enter the RESTORE command on the Operational Status and Tracking panel, CA MICS will bypass the pre-restore backup.

- CA MICS will also bypass the pre-restore backup if the Database is not updatable--that is, if prior CA MICS processing failed. CA MICS will NOT backup a bad Database.

- See the RESTORE Operational Job section (4.3.3) of this guide for more information on pre-restore backups.

The RESTORE Unit Database panel also gives you the option to review and/or modify the generated batch job before it is submitted for execution. You can cancel job submission processing after reviewing the generated JCL.

The rows of the RESTORE Unit Database panel describe your Database update, BACKUP, and RESTORE processing.

- Both the standard and special backups taken for this CA MICS unit database are listed.

- Unit database update processing is listed (incremental update processing is not included).

- Operational jobs submitted by Operational Status and Tracking are listed.

- Checkpoint changes made by Operational Status and Tracking are listed.

The display is sorted with the most recent processing first and in the order that processing/changes occurred. The most recent Database and/or checkpoint updates appear first.

Backup generations are identified by the date and time the backup was taken, the generation number, and a description. Standard CA MICS daily and monthly backups are identified by name -- BACKUP (after Daily), MONTHLY BACKUP, etc. Special backup generations are identified by the description you entered on the BACKUP Unit Database panel.

BACKUP entries are dropped from the audit log when the corresponding tape GDG entry expires -- you specified BACKUP retention in the prefix.MICS.PARMS(JCLGEN) using the BCKUPGDG parameter. Entries for Database update processing, job submission, and checkpoint changes that occurred prior to an expired backup are also dropped from the display. That is, once the backup tapes begin to expire (the BCKUPGDG limit is reached), the Database update processing, job submission, and checkpoint change entries older than the oldest backup entry are dropped.

CA MICS retains six monthly backup (MBACK) generations in the audit log.

Refer to the online tutorials for detailed data entry instructions and for information on using CA MICS commands (e.g., FIND, LOCATE, SHOW, SORT, etc.) on this panel.

Information reported by the example above includes:

- The most recent backup was created on October 10, 2001.

- The MONTHLY process executed on October 10. DAILY, MONTHLY, Monthly backup (MBACK), and BACKUP processing completed successfully.

- The WEEKLY process was submitted through Operational Status and Tracking on October 9.

  - The DAILY portion of this processing failed with a U101 abend in step DAY040.

  - The Database was restored and WEEKLY processing was resubmitted. WEEKLY completed successfully.

4.3.4.4.11 RESUME Command

Use the RESUME command to reactivate unit database update processing that was blocked by the SUSPEND command.

RESUME returns a CA MICS unit database to a normal operational status so that Database update processing can continue.

Use the RESUME command to reactivate normal CA MICS operations after you finish installing CA MICS PSP maintenance.

4.3.4.4.12 RSTATUS Command

Use the RSTATUS command to synchronize the Operational Status and Tracking control tables with the CA MICS unit database checkpoint file and to produce the CA MICS Run Status Report.

Normal unit database update, backup, and restore processing will keep Operational Status and Tracking synchronized with the checkpoint file. However, run status processing may not execute if Database update processing is canceled by the console operator, fails with a JCL error, or is terminated by an MVS system crash. Under these conditions, Operational Status and Tracking will not be synchronized with the unit database checkpoint file and RSTATUS will be required before Operational Status and Tracking can restart the Database update.

RSTATUS submits the prefix.MICS.CNTL(RSTATUS) batch job to update Operational Status and Tracking entries from the unit database checkpoint file. RSTATUS also saves a copy of the generated run status job in the prefix.MICS.RESTART.CNTL data set for the unit database.

If the RSTATUS job finds the database in a non-updatable condition, the RSTATUS job will terminate with a U999 abend. When the RSTATUS job terminates with a U999 abend, use the Operational Status and Tracking STATUS command or review the printed Run Status Report to analyze the database problem.

4.3.4.4.13 SCHEDULE Command

Use the SCHEDULE command to execute the CA MICS processing scheduled for today.

CA MICS maintains a processing schedule based on the specifications you entered in sharedprefix.MICS.PARMS (defaults for all units) and in prefix.MICS.PARMS for each unit database. You specified backup frequency in prefix.MICS.PARMS(JCLDEF) and indicated in prefix.MICS.PARMS(EXECDEF) whether WEEKLY, MONTHLY, and/or YEARLY processing can occur during weekends. You also defined your installation's calendar (e.g., a 13 month fiscal calendar) and week (e.g., the week starts on Sunday vs. Monday) through the sharedprefix.MICS.PARMS(CPLXDEF) or prefix.MICS.PARMS(SITE) parameter members.

SCHEDULE generates and submits a batch job to perform daily CA MICS database update processing and, if scheduled for today, database backup. If WEEKLY, MONTHLY, or YEARLY cycle database close-out processing is due, SCHEDULE includes the appropriate WEEKLY/MONTHLY/YEARLY processing in the generated batch job. SCHEDULE always includes a backup with WEEKLY/MONTHLY/YEARLY processing.

- You can specify the BACKUP operand with the SCHEDULE command to override the processing schedule and include a backup in the requested processing, even if no backup is currently scheduled.

- You can specify the NOBACKUP operand with the SCHEDULE command to omit backup processing even if BACKUP, WEEKLY, MONTHLY, or YEARLY processing is scheduled for today.

SCHEDULE also saves a copy of the generated database update processing in the prefix.MICS.RESTART.CNTL data set for the unit database.

4.3.4.4.14 STATUS Command

```
---------------------------  Unit Database Status  ----------------------------

 Command ===>


Database:  P  (PRIMARY) - CA MICS PRIMARY DATABASE

The status information was recorded at:  10OCT01 08:06
Status of this unit Database:            NON-UPDATABLE
Status of the cycle aging process:       Completed


 CA MICS  Last Completed
 Job      Step & Date     Status of Current Operation:   MONTHLY
--------  ---  -------    -----------------------------------------------------
DAILY     ALL  10OCT01    FAILED    DAY030                U310
MONTHLY   900  03SEP01    HELD      Prior job failed
BACKUP    900  09OCT01    HELD      Prior job failed
```

```
                          Status of Other Jobs
                          -----------------------------------------------------
WEEKLY    900  09OCT01    Completed
YEARLY    900  10JAN01    Completed
RESTORE   900  09OCT01    Completed



.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

The STATUS command displays status of the unit Database, the most recent scheduled processing, and unit Database update processes performed earlier.

Note: The status display does NOT reflect incremental update (INCRccc job) processing.

The top portion of the Unit Database Status panel summarizes current unit database status.

- Time the information was recorded.

- Database status:

  Completed   - Database updated successfully and available for ongoing processing.

  NON-UPDATABLE - Database update failed. RESTART required.

  SUSPENDED   - Database update processing has been blocked.

- Cycle aging status:

  Completed   - Cycle aging completed successfully.

  AGING   - Cycle aging is in progress. Call CA MICS Product Support for assistance before restarting.

The remainder of the Unit Database Status panel summarizes the status of the most recent execution for each of the CA MICS Database update jobs -- DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, and RESTORE. The name, last job step completed, processing date, and job status are displayed.

The STATUS display separates the most recent scheduled operations from those executed earlier. For example, if DAILY, WEEKLY, and BACKUP processing were scheduled for today, then DAILY, WEEKLY, and BACKUP would appear in the upper portion of the status display while MONTHLY, YEARLY and RESTORE would appear in the lower portion.

The Last Completed Step column should normally show a value of 900 -- job completed successfully. Otherwise, the last successful job step appears in this column. For example, if DAILY failed in the DAY500 (user reporting) step, then this panel would show 400 in the Last Completed Step.

The Status column lists current processing status.

- Completed - Completed successfully.

- DUE TODAY - Scheduled processing has not yet occurred.

- OVERDUE - Scheduled processing is late.

  Today's processing is OVERDUE if not complete by the user-specified deadline (hh:mm).

- Submitted - Submitted for execution, but not yet executing.

- INCOMPLETE - Execution started, but has not finished.

  If the processing is NOT still executing, then it may have been canceled by the operator, failed with a JCL error, or terminated by an MVS system crash. RESTART may be required.

- FAILED - Processing failed. RESTART required. The failing step is also listed.

- HELD - Scheduled to execute, but a prior process failed.

- SUSPENDED - Database update processing has been blocked.

4.3.4.4.15 SUSPEND Command

Use the SUSPEND command to block ALL unit database update processing.

SUSPEND marks a unit database as non-updatable.

- Operational Status and Tracking will not allow you to submit Database update processing for a unit database that is in suspended status.

- In addition, the Operational Status and Tracking SUSPEND command stops Database updates submitted by your production batch job scheduling facility or by the CA MICS batch SCHEDULE job.

Use the SUSPEND command to block Database updates while you are applying CA MICS PSP maintenance.

4.3.4.4.16 WEEKLY Command

Use the WEEKLY command to explicitly request weekly CA MICS Database update processing.

WEEKLY generates and submits a batch job to perform daily CA MICS Database update processing, weekly cycle Database close-out, and a Database backup.

Specify the NOBACKUP parameter to delete Database backup processing from the generated batch job.

WEEKLY also saves a copy of the generated Database update job in the prefix.MICS.RESTART.CNTL data set for the unit database.

4.3.4.4.17 YEARLY Command

Use the YEARLY command to explicitly request annual CA MICS Database update processing.

YEARLY generates and submits a batch job to perform daily CA MICS Database update processing, yearly cycle Database close-out, and a Database backup.

Specify the NOBACKUP parameter to delete Database backup processing from the generated batch job.

YEARLY also saves a copy of the generated Database update job in the prefix.MICS.RESTART.CNTL data set for the unit database.

4.3.4.4.18 INCRUPDT Command

Use the INCRUPDT command to access the incremental update job for the specified component and database.

The INCRUPDT command differs from the other supported commands but is similar to the SPLITSMF command in that it requires an additional parameter. The format of the primary command is:

```
INCRUPDT ccc dbid
```

where ccc is the 3-character component identifier and dbid is the 1-character database ID. INCRUPDT can be abbreviated to INC or INCR. Unlike other Operational Status and Tracking commands, the INCRUPDT command only supports one component (ccc) and dbid combination per execution of the command. That is, it can not be executed for multiple units within one execution of the command.

Execution as a line command requires use of the INCRUPDT command abbreviation in order to accommodate the required ccc parameter. The format of the line command is:

```
  INC ccc  -  or  - INCR ccc
```

No backup processing is performed with the incremental update (INCRccc) generated batch job.

The INCRUPDT command also saves a copy of the generated incremental database update job (INCRccc) in the prefix.MICS.RESTART.CNTL data set for the unit database.

4.3.4.4.19 SPLITSMF Command

Use the SPLITSMF command to access the splitsmf preprocessing job for the specified component and database.

The SPLITSMF command differs from the other supported commands but is similar to the INCRUPDT command in that it requires an additional parameter. The format of the primary command is:

```
SPLITSMF ccc dbid
```

where ccc is the 3-character component identifier and dbid is the 1-character database ID. SPLITSMF can be abbreviated to INC or INCR. Unlike other Operational Status and Tracking commands, the SPLITSMF command only supports one component (ccc) and dbid combination per execution of the command. That is, it can not be executed for multiple units within one execution of the command.

Execution as a line command requires use of the SPLITSMF command abbreviation in order to accommodate the required ccc parameter. The format of the line command is:

```
SPL ccc
```

No backup processing is performed with the splitsmf preprocessing (SPLITSMF) generated batch job.

The SPLITSMF command also saves a copy of the generated database update preprocessing job (SPLITSMF) in the prefix.MICS.RESTART.CNTL data set for the unit database.

4.3.4.4.20 AUDIT Command

Use the AUDIT command to explicitly request CA MICS audit archive processing.

AUDIT generates and submits a batch job to perform audit archival of the CA MICS DETAIL and/or DAYS files.

The AUDIT command is only available for a unit that has specified ARCHIVE AUDIT YES JOB in prefix.MICS.PARMS(JCLDEF). If weekly audit archive processing is executed in the WEEK300 step of the WEEKLY job rather than as a standalone job, a message is issued to that effect and no job is displayed.

No backup processing is included with the AUDIT generated batch job.

AUDIT also saves a copy of the generated audit archival job in the prefix.MICS.RESTART.CNTL data set for the unit database.

4.3.4.4.21 HISTM Command

Use the HISTM command to explicitly request CA MICS monthly history archive processing.

HISTM generates and submits a batch job to perform monthly history archive processing of the MONTHS timespan for the specified units.

The HISTM command is only available for a unit that has specified ARCHIVE HISTM YES JOB in prefix.MICS.PARMS(JCLDEF). If monthly history archive processing is executed in the MONTH300 step of the MONTHLY job rather than as a standalone job, a message is issued to that effect and no job is displayed.

No backup processing is performed with the HISTM generated batch job.

HISTM also saves a copy of the generated history archive job in the prefix.MICS.RESTART.CNTL data set for the unit database.

4.3.4.4.22 HISTW Command

Use the HISTW command to explicitly request CA MICS weekly history archive processing.

HISTW generates and submits a batch job to perform weekly history archive processing of the WEEKS timespan files for the specified units.

The HISTW command is only available for a unit that has specified ARCHIVE HISTW YES JOB in prefix.MICS.PARMS(JCLDEF). If weekly history archive processing is executed in the WEEK300 step of the WEEKLY job rather than as a standalone job, a message is issued to that effect and no job is displayed.

No backup processing is performed with the HISTW generated batch job.

HISTW also saves a copy of the generated weekly history archive job in prefix.MICS.RESTART.CNTL data set for the unit database.

4.3.5 Database Update Audit Trail (ADMIHL)

The Input History Log (IHL) file contains data identifying the raw data that has been input for CA MICS Database updating over the last n days, where n was defined using the INPUTLOGLIMIT keyword in prefix.MICS.PARMS(EXECDEF). The data is keyed by system identifier (SYSID), product, and update cycle.

The file is maintained in the DAYS timespan and is referenced by the label ADMIHL01. It is only maintained for one cycle, so it is always referenced as &pADMD..ADMIHL01 where p is the unit Database ID. For more information about the file, refer to the CA MICS Database Structure and Content Guide, section 2.2.

You can browse the Input History Log via the online Operational Status and Tracking facility (HISTORY command) or by using distributed MICF inquiries (e.g., BASLA1, CA MICS Daily Processing Summary).

The following diagram illustrates how the Input History Log File is updated. Each DAILY job step adds information to the IHL file for the CA MICS product or field developed application being processed.

```
+-----------------------------------------------------------+
|                                                           |
|  CA MICS Daily Job Stream                                 |
|                                                           |
|  +---------------------------+                            |
|  |                           |                            |
|  |   Step DAYSMF in DAILY    |   +---------+              |
|  |  +---------------------+  |   |SYSID/   |              |
|  |  | Write SYSID/        |----->|Product  |              |
|  |  | product checkpoint  |  |   |Data     |              |
|  |  | to a temporary file |  |   +---------+              |
|  |  +---------------------+  |  /                         |
|  |                           | /                          |
|  |   Update Steps in Daily   |/ Input History Log         |
|  |  +---------------------+  /   +---------+              |
|  |  | Update Input History| /|   |SYSID/   |              |
|  |  | Log with today's    |----->|Product  |              |
|  |  | update status.      |  |   |Data     |              |
|  |  +---------------------+  |   +---------+              |
|  +---------------------------+                            |
+-----------------------------------------------------------+
```

When incremental update is active, audit trail information is collected by each INCRccc job execution and maintained in the product's incremental update DAYS level database (IUDAYS).

At the end of the day, the product's DAILY job step (DAYnnn) "rolls-up" the incremental update audit trail information, including audit trail data from the DAYnnn step itself to the unit database ADMIHL01 file.

Note: Audit trail information for incremental update (INCRccc job) executions is NOT reflected in the ADMIHL file UNTIL the associated DAILY job (DAYnnn) step completes.

4.3.6 Run Status Report

The CA MICS Run Status Report shows critical database update status information from the unit checkpoint file (prefix.MICS.CHECKPT.DATA). To understand CA MICS checkpoint facilities, see section 4.3.7, CA MICS Checkpoint File, of this guide.

The CA MICS Run Status Report is produced several times:

- As the last report in any DAILY, WEEKLY, MONTHLY, or YEARLY operational job.
- After database BACKUP or RESTORE jobs.
- As the last report in any INCRccc, AUDIT, HISTW and HISTM operational job.
- After unsuccessful execution of the SCHEDULE job.
- On demand, by running the RSTATUS job.

If the Run Status Report finds the database in a non-updatable condition, the Run Status Report step will terminate with a U0999 abend. When you find a U0999 abend in a Run Status Report step (INCRRSR, DAYRSR, WEEKRSR, MONTHRSR, YEARRSR, BKUPRSR, RSTRRSR, SCDRSR, or RSTATUS), see the Run Status Report for an explanation of the database problem.

Note: The Run Status Report only processes the incremental update checkpoint data set during execution of INCRccc processing. During all other processing, INCRccc job status is not reflected in the Run Status Report. Other processing (DAILY, WEEKLY, and so on) will report on the unit checkpoint (prefix.MICS.CHECKPT.DATA). However, if the DAYALL step of the DAILY job aborts due to incremental update related issues, the unit checkpoint is updated with flags that cause the Run Status Report to document the incremental update-related failure.

Figure 4-3 depicts the report's format. It is divided into multiple "report blocks."

- The first three report blocks (as shown in Figure 4-3) will always appear.
  - The first block identifies the report.
  - The second block identifies the overall database update

status.

- The third block lists the detailed status of each of the operational database update jobs.

o Additional report blocks are displayed to document issues related to incremental update and/or standalone archive tape processing.

Information provided in the first three blocks of the Run Status Report is also available through the Operational Status and Tracking facility STATUS command.

```
+-----------------------------------------------------------+
|  CA MICS ssssssss RUN STATUS REPORT FOR DATABASE dddddddd |
|                                                           |
| PROCESS: jjjjjjj            DATE: dayname,month dd,yyyy  |
|-----------------------------------------------------------|
|                                                           |
| PROCESS | SUCCESSFULLY TERMINATED  | ON ddmmmyy.          |
|         | IN PROGRESS              |                      |
|         | SUCCESSFULLY INITIALIZED |                      |
|         | ABNORMALLY TERMINATED    |                      |
|                                                           |
||RESTART REQUIRED: | RESTART=(stepname.MICS) |          |  |
||                  | RERUN BACKUP JOB        |          |  |
||                  | RERUN RESTORE JOB       |          |  |
||                                                      |  |
||ABEND IN AGING.  CONTACT CA MICS PRODUCT SUPPORT.      |  |
||                                                      |  |
||MULTIPLE ABNORMAL ENDS.  CONTACT CA MICS PRODUCT SUPPORT. |
|                                                           |
|-----------------------------------------------------------|
|                                                           |
|             GENERAL RUN STATUS FOR ALL PERIODS            |
|                                                           |
| JOB       COMPLETION DATE        TERMINATION STATUS       |
|-----------------------------------------------------------|
|                                                           |
| DAILY         ddmmmyy     | SUCCESSFULLY TERMINATED     | |
|                           | ABNORMALLY ENDED IN STEP xxx| |
|     -  OR  -              | INVALID CHECKPOINT FILE     | |
|                                                           |
| INCRUPDT      ddmmmyy     | SUCCESSFULLY TERMINATED     | |
|                           | ABNORMALLY ENDED IN INCRxxx | |
|                           | INVALID CHECKPOINT FILE     | |
|                                                           |
| WEEKLY        ddmmmyy     | SUCCESSFULLY TERMINATED     | |
|                           | ABNORMALLY ENDED IN STEP xxx| |
|                           | INVALID CHECKPOINT FILE     | |
|                                                           |
| MONTHLY       ddmmmyy     | SUCCESSFULLY TERMINATED     | |
|                           | ABNORMALLY ENDED IN STEP xxx| |
|                           | INVALID CHECKPOINT FILE     | |
|                                                           |
| YEARLY        ddmmmyy     | SUCCESSFULLY TERMINATED     | |
|                           | ABNORMALLY ENDED IN STEP xxx| |
|                           | INVALID CHECKPOINT FILE     | |
|                                                           |
| BACKUP        ddmmmyy     | SUCCESSFULLY TERMINATED     | |
|                           | ABNORMALLY ENDED IN STEP xxx| |
|                           | INVALID CHECKPOINT FILE     | |
|                                                           |
| RESTORE       ddmmmyy     | SUCCESSFULLY TERMINATED     | |
|                           | ABNORMALLY ENDED IN STEP xxx| |
|                           | INVALID CHECKPOINT FILE     | |
|                                                           |
| AUDIT         ddmmmyy     | SUCCESSFULLY TERMINATED     | |
|                           | ABNORMALLY ENDED IN STEP xxx| |
|                           | INVALID CHECKPOINT FILE     | |
|                                                           |
| HISTW         ddmmmyy     | SUCCESSFULLY TERMINATED     | |
|                           | ABNORMALLY ENDED IN STEP xxx| |
|                           | INVALID CHECKPOINT FILE     | |
|                                                           |
| HISTM         ddmmmyy     | SUCCESSFULLY TERMINATED     | |
|                           | ABNORMALLY ENDED IN STEP xxx| |
|                           | INVALID CHECKPOINT FILE     | |
|                                                           |
+-----------------------------------------------------------+
```

Figure 4-3. Format of CA MICS Run Status Report

FIRST REPORT BLOCK

The first block of the report contains the title and timestamp of the report and identifies the last process of interest at the time the report is printed. In the first report content block:

ssssssss   Represents the name of the job step from which the report program was called.

dddddddd   Is the 8-character database unit identifier specified in the DATABASE statement of prefix.MICS.PARMS(JCLDEF).

jjjjjjjj   Is the name of the process of interest in the current report.

DATE: Is the date on which the report was produced.

The content of the report depends on the following:

- The content of the update status record in prefix.MICS.CHECKPT.DATA, or in the case of INCRccc, the update status record in prefix.ccc.IUCKPT.DATA.

- The operational job step from which the Run Status Report was executed.

- Any previous scheduling activity done by Operational Status and Tracking or the CA MICS SCHEDULE job.

SECOND REPORT BLOCK

In the second block of the report, the overall status of the database update is given.

If all processes have successfully terminated:

- If the checkpoint data set is not at its initial state, the general status comment in block 2 will contain one of these comments:

  PROCESS SUCCESSFULLY TERMINATED ON ddmmmyy.

  PROCESS IN PROGRESS ON ddmmmyy.

  The process identifier in the first block of the report will be INCRUPDT, DAILY, WEEKLY, MONTHLY, or YEARLY. The process is identified from the Run Status Report executing under the conditions noted in the following table:

IF the	AND the	THEN the	AND the
report was	last item	block 1	process
run in	scheduled	PROCESS name	status
this step	was	is set to be	printed is
DAYRSR	DAILY	DAILY	SUCCESSFUL
	WEEKLY	WEEKLY	INCOMPLETE
	MONTHLY	MONTHLY	INCOMPLETE
	YEARLY	YEARLY	INCOMPLETE
INCRRSR	any	INCRUPDT	SUCCESSFUL

WEEKRSR	any	WEEKLY	SUCCESSFUL
MONTHRSR	any	MONTHLY	SUCCESSFUL
YEARRSR	any	YEARLY	SUCCESSFUL
BKUPRSR	DAILY	DAILY	SUCCESSFUL
	WEEKLY	WEEKLY	SUCCESSFUL
	MONTHLY	MONTHLY	SUCCESSFUL
	YEARLY	YEARLY	SUCCESSFUL
RSTRRSR	DAILY	DAILY	SUCCESSFUL
	WEEKLY	WEEKLY	SUCCESSFUL
	MONTHLY	MONTHLY	SUCCESSFUL
	YEARLY	YEARLY	SUCCESSFUL
RSTATUS	DAILY	DAILY	SUCCESSFUL
	WEEKLY	WEEKLY	SUCCESSFUL
	MONTHLY	MONTHLY	SUCCESSFUL
	YEARLY	YEARLY	SUCCESSFUL

If a process has abnormally terminated:

o The process identifier in the first block of the report will be INCRUPDT, DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, or RESTORE.

   If the abnormal termination is in a WEEK, MONTH, or YEAR step, then the process name is WEEKLY, MONTHLY, or YEARLY, respectively.

   If the abnormal termination is in a DAY step, then the process name is the update process last scheduled (DAILY, WEEKLY, MONTHLY, or YEARLY).

   If the abnormal termination is in an INCR step, then the process name is INCRUPDT.

o If the abending process was DAILY, WEEKLY, MONTHLY, or YEARLY, the general status comment in block 2 will contain:

```
PROCESS ABNORMALLY TERMINATED ON ddmmmyy.
RESTART REQUIRED:  RESTART=(stepname.MICS)
```

   If the abending process was in aging at the time, an additional line will be printed:

ABEND IN AGING. CONTACT CA MICS PRODUCT SUPPORT.

- o If the abending process was BACKUP, the general status comment in block 2 will contain:

  PROCESS ABNORMALLY TERMINATED ON ddmmmyy.
  RESTART REQUIRED: RERUN BACKUP JOB

- o If the abending process was RESTORE, the general status comment in block 2 will contain:

  PROCESS ABNORMALLY TERMINATED ON ddmmmyy.
  RESTART REQUIRED: RERUN RESTORE JOB

- o If prefix.MICS.CHECKPT.DATA has been destroyed, the general status comment in block 2 will contain:

  PROCESS ABNORMALLY TERMINATED ON ddmmmyy. MULTIPLE
  ABNORMAL ENDS. CONTACT CA MICS PRODUCT SUPPORT.

The Run Status Report identifies the SCHEDULEd process as having failed if the following conditions are true:

- o All detailed processes have successfully completed with step identifiers of 900.

- o The last SCHEDULEd process was WEEKLY, MONTHLY, or YEARLY.

- o The DAILY status date is later than the WEEKLY, MONTHLY, or YEARLY status date.

It prints restart instructions to restart the last SCHEDULEd process from prefix.MICS.RESTART.CNTL in the first step of the WEEKLY, MONTHLY, or YEARLY process. This situation is much easier to see with an example:

- o Assume a WEEKLY process was scheduled using the Operational Status and Tracking SCHEDULE command. This fact is recorded in the update status record of prefix.MICS.CHECKPT.DATA by the scheduling flags noted in section 4.3.3.3. A successful execution of the Run Status Report from the WEEKRSR step of the WEEKLY JCL will clear these flags.

- o Assume the checkpoint file shows the DAILY update completed on 13AUG01 and the WEEKLY update completed on 06AUG01. If the scheduling flag has not been cleared, the RSTATUS program senses that the WEEKLY update never started after the last part of the DAILY update completed successfully. Therefore, the WEEKLY process scheduled had abnormally terminated and restart should be done in the first step of the WEEKLY JCL.

THIRD REPORT BLOCK

The third block of the CA MICS Run Status Report is the GENERAL RUN STATUS FOR ALL PERIODS. This block shows the DAILY, WEEKLY, MONTHLY, and YEARLY update processes, the BACKUP and RESTORE database maintenance processes, the date upon which they last completed execution, and their termination status.

Note: If the executing process was INCRccc, then the INCRUPDT process will replace the DAILY process line.

SPECIAL CASE: AFTER THE RUN OF A CKPTINIT JOB

When generating a new database unit, the CKPTINIT job puts the checkpoint data set in an initialized state. This state persists until data is processed by a DAILY run for the unit so that files are written in the database and data range records are written in the Checkpoint File. When in that initialized state,

- o The process identifier in the first block of the report will be CKPTINIT.

- o The general status comment in block 2 will contain:

  PROCESS SUCCESSFULLY INITIALIZED ON ddmmmyy.

- o The termination status on all lines of block 3 will contain:

  SUCCESSFULLY INITIALIZED

INCREMENTAL UPDATE AND ARCHIVE TAPE ISSUES BLOCK

This report block appears when processing is aborted due to problems with incremental update or standalone archive tape processing.

Incremental update related problems are reported in the INCRRSR step as well as when the DAILY job fails in the DAYALL step. The following is an example of the messages you might encounter in this situation. See the DAYALL step

MICSLOG listing for a detailed explanation of the error conditions.

```
+-----------------------------------------------------------+
|                                                           |
| INCREMENTAL UPDATE AND ARCHIVE TAPE RELATED ISSUES:      |
|                                                           |
|    ONE OR MORE INCREMENTAL UPDATE (INCRccc) JOBS IS IN   |
|      PROGRESS OR REQUIRES RECOVERY/RESTART               |
|                                                           |
|    ONE OR MORE INCREMENTAL UPDATE CHECKPOINT FILES       |
|      WERE NOT FOUND OR COULD NOT BE ALLOCATED DISP=SHR   |
|                                                           |
|    THE UNIT CHECKPOINT IS NOT LARGE ENOUGH TO HOLD THE NEW|
|      ENTRIES FROM THE INCR. UPDATE CHECKPOINT DATA SETS  |
|                                                           |
+-----------------------------------------------------------+
```

If the DAILY job DAYALL step or the first step of the WEEKLY or MONTHLY job aborts because one of the standalone archive tape processing jobs (AUDIT, HISTW, or HISTM) has not yet executed, messages describing the problem will be found both in the Run Status Report and in the failing step's MICSLOG listing. The following is an example of the messages you might encounter in one of these situations.

```
+-----------------------------------------------------------+
|                                                           |
| INCREMENTAL UPDATE AND ARCHIVE TAPE RELATED ISSUES:      |
|                                                           |
|    AUDIT JOB FOR THE 19MAR2001 WEEKLY HAS NOT YET EXECUTED|
|      EXECUTE THE AUDIT JOB AND THEN RESTART WEEKLY       |
|                                                           |
|    HISTW JOB FOR THE 19MAR2001 WEEKLY HAS NOT YET EXECUTED|
|      EXECUTE THE HISTW JOB AND THEN RESTART WEEKLY       |
|                                                           |
|    HISTM JOB FOR THE 03MAR2001 MONTHLY HAS NOT YET EXECUTE|
|      EXECUTE THE HISTM JOB AND THEN RESTART MONTHLY      |
|                                                           |
+-----------------------------------------------------------+
```

STANDALONE ARCHIVE PROCESSING WARNINGS BLOCK

This report block may appear when you enable standalone archive tape processing. It includes warnings about pending archive tape jobs that may be required before the next DAILY, WEEKLY, or MONTHLY job will execute.

This information is provided as an early warning of potential problems you may encounter. When you see these messages, you may want to verify that AUDIT, HISTW, and/or HISTM jobs are scheduled for execution at the proper times.

The following is an example of the archive tape warning messages you might see.

```
+-----------------------------------------------------------+
|                                                           |
| WARNING:  ONE OR MORE ARCHIVE TAPE JOBS NEED TO BE RUN    |
|                                                           |
|    AUDIT JOB FOR THE 29FEB2000 WEEKLY HAS NOT YET EXECUTED|
|      EXECUTE THE AUDIT JOB BEFORE THE NEXT DAILY JOB      |
|                                                           |
|    HISTW JOB FOR THE 29FEB2000 WEEKLY HAS NOT YET EXECUTED|
|      EXECUTE THE HISTW JOB BEFORE THE NEXT WEEKLY JOB     |
|                                                           |
|    HISTM JOB FOR THE 29FEB2000 MONTHLY HAS NOT YET EXECUTE|
|      EXECUTE THE HISTM JOB BEFORE THE NEXT MONTHLY JOB    |
|                                                           |
+-----------------------------------------------------------+
```

4.3.7 CA MICS Checkpoint File

The CA MICS Checkpoint File is a permanent, physical sequential data set where CA MICS maintains a running status of the system's operation and database contents for an entire unit database. The Checkpoint File is frequently referred to as the unit checkpoint. The unit checkpoint contains a fixed number of records. It is updated by each database update job step in the DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, and RESTORE operational jobs. The update takes place as the last action of the step and only after all previous functions within the step have successfully executed.

The Checkpoint File is concerned with ensuring database integrity across database update job steps and update cycles.

o The unit checkpoint should not be confused with database update restart checkpoints which are product specific, temporary files used to save an image of the operational environment for restart within a database update job step.

o The unit checkpoint is also separate and distinct from the optional incremental update checkpoint files which are product specific, permanent data sets used to ensure the integrity of incremental update processing for a single CA MICS data integration product. The INCRccc jobs do NOT modify the unit checkpoint.

The checkpoint file is also updated when optional stand-alone Archive Audit (AUDIT job), Archive Weekly History (HISTW job), Archive Monthly History (HISTM job) processing completes. These updates take place while the checkpoint file is allocated DISP=SHR and use internal CA MICS ENQ/DEQ facilities to prevent concurrent updates to the checkpoint file.

In addition, Incremental Update INCRccc jobs may update the checkpoint file to clear FORCE/SELECT specifications that were established for the INCRccc job execution. This update also allocates the checkpoint file DISP=SHR and depends on the CA MICS ENQ/DEQ facilities to ensure data set integrity.

The following diagram illustrates the relationship of the Checkpoint File to a database update job step in the DAILY, WEEKLY, MONTHLY, or YEARLY operational job.

NOTE: The INCRccc jobs process the product-level incremental update checkpoint files in a very similar manner, but do NOT update the unit checkpoint except to clear

FORCE/SELECT specifications.

```
+-----------------------------------------------------------+
|                                                           |
|  CA MICS Job (for example, DAILY)                         |
|                                                           |
|  +---------------------------+                            |
|  |                           |               +-------+    |
|  |   Step N in Job           |               |   C   |    |
|  |  +---------------------+  |  Read Only    |   H   |    |
|  |  | Step's First Action |<-----------------|   E   |    |
|  |  +---------------------+  |               |   C   |    |
|  |  | Step's Processing   |  |               |   K   |    |
|  |  +---------------------+  |Update In-place|   P   |    |
|  |  | Start of Aging      |<---------------->|   O   |    |
|  |  +---------------------+  |Update In-place|   I   |    |
|  |  | SYSID/COMPONENT     |<---------------->|   N   |    |
|  |  | Update, If Required |  |               |   T   |    |
|  |  +---------------------+  |Update In-place|       |    |
|  |  | Aging Completed     |<---------------->|   F   |    |
|  |  +---------------------+  |Update In-place|   I   |    |
|  |  | Step's Last Action  |<-+-------------->|   L   |    |
|  |  +---------------------+  |               |   E   |    |
|  |                           |               +-------+    |
|  +---------------------------+                            |
|                                                           |
|                                                           |
+-----------------------------------------------------------+
```

The Checkpoint File is updated by each step in each of the DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, and RESTORE jobs. The Checkpoint File is referenced at least twice by each step:

o The first reference is a read only access enabling the step to determine if it is permitted to execute. If the prerequisite step has not executed, the job stream is aborted with a user abend 100. If the prerequisite step has successfully executed, then the step's normal processing functions are performed.

o The second reference occurs after all functions operate successfully. At that point, control is passed to the last action of the step, which again accesses the Checkpoint File to update its status, indicating that this step has successfully completed.

Prerequisite steps are identified in the Job Steps section of this guide. For example, before executing DAY030, DAY020 must complete successfully if both the Batch and Operations Analyzer and the Hardware and SCP Analyzer are installed in the unit database.

For steps that update the database, the aging flag in the Checkpoint File indicates that the Database is being updated. The flag is set prior to cycle aging and cleared when cycle aging has completed. If an abnormal termination occurs during cycle aging, this flag notifies you that normal restart is not allowed.

Database update steps also update the time range status section of the Checkpoint File by recording the date range of the data used to update the Database.

The RSR (Run Status Report) step of each operational job updates the current process flag (D, W, M, or Y), but does NOT alter the aging flag or the job step number fields. Failures in the RSR step do NOT impact database update. The database remains in updatable condition and restart is not required.

The optional incremental update checkpoint files have the same format as the unit checkpoint file. Each incremental update checkpoint file is initialized as a copy of the unit checkpoint file by the first INCRnnn job for the corresponding product executed each day. Subsequent INCRccc job executions update the ccc product incremental update checkpoint in the same way that the DAYnnn steps execute the unit checkpoint file. Then, at the end of the day, the corresponding DAYnnn step updates the unit checkpoint file from the incremental update checkpoint file.

Additional fields in the checkpoint are used in monitoring and controlling stand-alone archive tape processing and to identify DAYALL step abends due to problems in Incremental Update processing.

o When Incremental Update is active for one or more products in the unit, the DAILY job DAYALL step sets the IUS_FLAG field to '...' and then verifies that all pending INCRccc jobs are complete and that there is room in the checkpoint file for rolling-up checkpoint information from the incremental update checkpoint files. Error conditions are marked with an 'I', 'U', and/or 'S' in the appropriate column, and these flags are used in formatting the Run Status Report output. DAYALL clears the IUS_FLAG to blanks if no problems are encountered.

- When optional stand-alone archive tape processing is enabled, the WEEK300 and MONTH300 steps set flags indicating that AUDIT, HISTW, and/or HISTM operational job execution is required. The DAILY job DAYALL step and the first steps of the WEEKLY and/or MONTHLY jobs check these flags to determine if prerequisite processing has completed. Archive tape processing status is also included in the Run Status Report.

The unit checkpoint file and incremental update checkpoint files are documented further in the following sections.

1 - Checkpoint Restart and Termination Status Record
2 - Checkpoint Database Update Time Range Records
3 - Duplicate Data Check Process
4 - Using the Select Option for Input Data
5 - Using the Force Option for Input Data

4.3.7.1 Checkpoint Restart and Termination Status Record

There are a fixed number of records in the unit checkpoint file and the optional incremental update checkpoint files. The first record in the each checkpoint file is the Restart and Termination Status Record. The remaining records are the Database Update Time Range Records.

Figure 4-4 summarizes the Restart and Termination Status Record format.

The Checkpoint Restart and Termination Status Record contains a flag, ending step name, and ending date for each of the DAILY, WEEKLY, MONTHLY, YEARLY, BACKUP, and RESTORE operational jobs. Additionally, the database prefix, assigned by the prefix.MICS.PARMS(JCLDEF) PREFIX statement, is also maintained to ensure that the CA MICS procedures (MICSSHR, MICSDB, etc.) are being used with their corresponding database. Other flag fields (described later) are used in monitoring and controlling stand-alone archive tape processing and in diagnosing failures in the DAILY job's DAYALL step (the first step of every DAILY job).

The flags and ending step names are essential and must be maintained intact according to the specifications for their use. The flags must be D, W, M, Y, B, or R, in their respective columns, and the ending step names must have one of the values listed in Step Name column of the charts in the Job Steps section of this guide (e.g., D030 for step DAY030).

NOTE: The Restart and Termination Status Record in the optional incremental update checkpoint files is formatted exactly the same as in the unit checkpoint file. However, incremental update processing uses only the D= flag and processing date.

The six ending dates are used by Operational Status and Tracking and by the CA MICS SCHEDULE job to determine when DAILY, WEEKLY, MONTHLY, YEARLY, and BACKUP processing are due. The dates are also printed on the Run Status Report and saved in the Operational Status and Tracking Control tables.

CA MICS provides two mechanisms to bypass the data integrity checks built into the system. The Select and Force facilities are implemented through the Checkpoint File. These mechanisms should be used with care to ensure that the integrity of the Database is not compromised.

o The Select facility enables the Administrator to select a portion of the data on a given input file. It is especially valuable for processing cumulative input files (e.g., SMF history files).

- The Force facility enables the System Administrator to override standard Database time-range checks. This facility is useful in situations where a portion of data was unavailable for the normal operational processing. Because the data was within the timespan of the data used to update the Database, it would no longer be allowed to pass the standard time-range tests on a succeeding update. Force enables the System Administrator to process data already encompassed by the Database's current time-ranges.

```
+--------------+----------+--------+------------------------+
| Data Element | Position | Format | Purpose                |
+--------------+----------+--------+------------------------+
| D_FLAG       |   01-01  |  X     | The value 'D'.         |
| Delimiter    |   02-02  |  X     | The value '='.         |
| D_STEP       |   03-05  |  XXX   | Last step successfully |
|              |          |        | completed in DAILY or  |
|              |          |        | INCRccc.               |
|              |   06-06  |        | RESERVED               |
| D_DATE       |   07-13  | DDMMMYY| Date step terminated.  |
|              |   14-14  |        | RESERVED               |
| W_FLAG       |   15-15  |  X     | The value 'W'.         |
| Delimiter    |   16-16  |  X     | The value '='.         |
| W_STEP       |   17-19  |  XXX   | Last step successfully |
|              |          |        | completed in WEEKLY.   |
|              |   20-20  |        | RESERVED               |
| W_DATE       |   21-27  | DDMMMYY| Date step terminated.  |
|              |   28-28  |        | RESERVED               |
| M_FLAG       |   29-29  |  X     | The value 'M'.         |
| Delimiter    |   30-30  |  X     | The value '='.         |
| M_STEP       |   31-33  |  XXX   | Last step successfully |
|              |          |        | completed in MONTHLY.  |
|              |   34-34  |        | RESERVED               |
| M_DATE       |   35-41  | DDMMMYY| Date step terminated.  |
|              |   42-42  |        | RESERVED               |
| Y_FLAG       |   43-43  |  X     | The value 'Y'.         |
| Delimiter    |   44-44  |  X     | The value '='.         |
| Y_STEP       |   45-47  |  XXX   | Last step successfully |
|              |          |        | completed in YEARLY.   |
|              |   48-48  |        | RESERVED               |
| Y_DATE       |   49-56  | DDMMMYY| Date step terminated.  |
.              .          .        .                        .
.              .          .        .                        .
.              .          .        .                        .

                 Continued in next column
```

```
             Continued from prior column
+--------------+----------+--------+------------------------+
| Data Element | Position | Format | Purpose                |
+--------------+----------+--------+------------------------+
.              .          .        .                        .
.              .          .        .                        .
.              .          .        .                        .
| B_FLAG       |   57-57  |  X     | The value 'B'.         |
| Delimiter    |   58-58  |  X     | The value '='.         |
| B_STEP       |   59-61  |  XXX   | Last step successfully |
|              |          |        | completed in BACKUP.   |
|              |   62-62  |        | RESERVED               |
| B_DATE       |   63-69  | DDMMMYY| Date step terminated.  |
|              |   70-70  |        | RESERVED               |
| R_FLAG       |   71-71  |  X     | The value 'R'.         |
| Delimiter    |   72-72  |  X     | The value '='.         |
| R_STEP       |   73-75  |  XXX   | Last step successfully |
|              |          |        | completed in RESTORE.  |
|              |   76-76  |        | RESERVED               |
| R_DATE       |   77-83  | DDMMMYY| Date step terminated.  |
|              |   84-84  |        | RESERVED               |
| IUS_FLAG     |   85-87  |  IUS   | Flags set for abends   |
|              |          |        | in DAYALL related to   |
|              |          |        | Incr. Update processing|
|              |          |        |   Blank - DAYALL ok    |
|              |          |        |   'I..' - one or more  |
|              |          |        |           INCRccc      |
|              |          |        |           failures     |
|              |          |        |   '.U.' - Unable to    |
|              |          |        |           allocate one |
|              |          |        |           or more Incr.|
|              |          |        |           Update Ckpt. |
|              |          |        |           data sets    |
|              |          |        |   '..S' - No room in   |
|              |          |        |           unit ckpt for|
|              |          |        |           new entries  |
|              |          |        |           from Incr.   |
|              |          |        |           Update Ckpt. |
|              |          |        |           data sets    |
|              |   88-124 |        | RESERVED               |
.              .          .        .                        .
.              .          .        .                        .
.              .          .        .                        .

               Continued in next column
             Continued from prior column
+--------------+----------+--------+------------------------+
| Data Element | Position | Format | Purpose                |
+--------------+----------+--------+------------------------+
.              .          .        .                        .
.              .          .        .                        .
.              .          .        .                        .
| AAUD_FLG     |  125-126 |   xx   | Archive Audit status   |
|              |          |        | flag                   |
|              |          |        |   Blank - Archive Audit|
|              |          |        |           is in WEEK300|
|              |          |        |   AR    - AUDIT job    |
|              |          |        |           required     |
|              |          |        |   AC    - AUDIT job    |
|              |          |        |           completed    |
|              |  127-127 |        | RESERVED               |
| AAUD_DTE     |  128-134 | DDMMMYY| Date 'AR'/'AC' flag set|
|              |  134-135 |        | RESERVED               |
| AWKH_FLG     |  136-137 |   xx   | Archive Weekly History |
|              |          |        | status flag            |
|              |          |        |   Blank - Weekly       |
|              |          |        |           History is   |
|              |          |        |           in WEEK300   |
|              |          |        |   WR    - HISTW job    |
|              |          |        |           required     |
|              |          |        |   WC    - HISTW job    |
|              |          |        |           completed    |
|              |  138-138 |        | RESERVED               |
| AWKH_DTE     |  139-145 | DDMMMYY| Date 'WR'/'WC' flag set|
|              |  146-146 |        | RESERVED               |
| AMNH_FLG     |  147-148 |   xx   | Archive Monthly History|
|              |          |        | status flag            |
|              |          |        |   Blank - Monthly      |
|              |          |        |           History is   |
|              |          |        |           in WEEK300   |
|              |          |        |   MR    - HISTM job    |
|              |          |        |           required     |
|              |          |        |   MC    - HISTM job    |
|              |          |        |           completed    |
|              |  149-149 |        | RESERVED               |
| AMNH_DTE     |  150-156 | DDMMMYY| Date 'MR'/'MC' flag set|
|              |  157-157 |        | RESERVED               |
.              .          .        .                        .
.              .          .        .                        .
.              .          .        .                        .

               Continued in next column
```

```
            Continued from prior column
+--------------+----------+--------+------------------------+
| Data Element | Position | Format | Purpose                |
+--------------+----------+--------+------------------------+
.              .          .        .                        .
.              .          .        .                        .
.              .          .        .                        .
| AGING        |  158-162 |  XXXXX | Aging note.  If blank, |
|              |          |        | database aging was     |
|              |          |        | successful. If AGING   |
|              |          |        | appears in columns,    |
|              |          |        | abend occurred within  |
|              |          |        | the aging process.     |
|              |          |        | If IURLP appears in    |
|              |          |        | columns, abend occurred|
|              |          |        | after incremental      |
|              |          |        | update roll-up to the  |
|              |          |        | unit database.         |
|              |  163-163 |        | RESERVED               |
| DB PREFIX    |  164-185 |  X(19) | The database prefix,   |
|              |          |        | used by the procedure  |
|              |          |        | and database, defined  |
|              |          |        | in MICS.PARMS(JCLDEF). |
|              |  186-192 |        | RESERVED               |
| AS_FLAGS     |  193-197 |  DWMYB | Flags used by the      |
|              |          |        | Automatic Scheduler.   |
| PROCESS FLAG |  198     |  X     | Current CA MICS Process|
|              |  199-200 |        | RESERVED               |
+--------------+----------+--------+------------------------+

 Figure 4-4. Checkpoint Restart and Termination Status Record
```

4.3.7.2 Checkpoint Database Update Time Range Records

There are a fixed number of records in the unit checkpoint file and the incremental update checkpoint files. The first record is the Restart and Termination Status Record. The remaining records are the Database Update Time Range Records.

There are normally 100 Database Update Time Range Records in a checkpoint file. The number of Update Time Range Records is specified by the prefix.MICS.PARMS(SITE) CKPTCNT parameter (which defaults to 100).

Figure 4-5 summarizes the database update time range record format.

The Database update time range records document the types of data, systems, and time ranges processed by CA MICS. This information is used in protecting the Database from duplicate input data.

CA MICS provides two levels of protection against using the same data more than once (duplicate data) in the Database:

o Duplicate data within the same input stream is automatically dropped by each of the respective CA MICS update processors on a daily basis. This process does not involve the Checkpoint File.

o Duplicate input data occurs when data that has previously been input into the database update is again inadvertently input to the update process. If the input data falls within the time-range kept by ORGSYSID and product in the Checkpoint File, it is automatically dropped, unless specific administrator action is signaled to force the data through.

CA MICS uses the unit checkpoint file to maintain an update time range bounding the lowest and highest date/time of data processed in the last or any previous update for each CA MICS product and system. The incremental update checkpoint files maintain the update time range for INCRccc executions during the day. Time range records are maintained by original system identifier (ORGSYSID) and CA MICS product. For systems such as IMS and CICS, the product check is extended to include the subsystem identifier (e.g., IMSID or CICSID). By matching each individual input record read against the table of time ranges, CA MICS avoids processing the same input data twice.

You can browse unit checkpoint file Database Update Time Range records with the Operational Status and Tracking CHECKPT command. You can also browse the unit checkpoint or any of the optional incremental update checkpoint files with PDF Browse.

```
+--------------+----------+--------+------------------------+
| Data Element | Position | Format | Purpose                |
+--------------+----------+--------+------------------------|
| ORGSYSID     |   01-04  |  X(4)  | Original System ID     |
|              |   05-05  |  X     | Reserved               |
| COMPONENT    |   06-09  |  X(4)  | CA MICS Component Name |
|              |          |        | (TSO, RMF, SMF, etc.)  |
|              |   10-13  |  X(4)  | Subsystem ID           |
|              |          |        | (IMSID, CICSID, etc.)  |
|              |   14-14  |  X     | Reserved               |
| LOWEST DATA  |   15-33  |  Time  | Lowest date and time   |
|  TIME-STAMP  |          |  Stamp | EVER input and used    |
|              |          |        | in the update process  |
|              |          |        | for this SYSID and     |
|              |          |        | COMPONENT combination  |
|              |          |        |  DDMMMYY:HH:MM:SS.SS   |
|              |   34-34  |  X     | Reserved.              |
| HIGHEST DATA |   35-53  |  Time  | Highest date and time  |
|  TIME-STAMP  |          |  Stamp | EVER input and used    |
|              |          |        | in the update process  |
|              |          |        | for this SYSID and     |
|              |          |        | COMPONENT combination  |
|              |          |        |  DDMMMYY:HH:MM:SS.SS   |
|              |   54-54  |  X     | Reserved               |
|                                                           |
|  The following fields apply to Select and Force.  Enter   |
|  these options in any order or combination.               |
| FORCE/SELECT |   55-56  |  XX    | 'S(' for SELECT option |
|              |          |        | 'F(' for FORCE option  |
| LOW  DATETIME|   57-75  |  Time  |  DDMMMYY:HH:MM:SS.SS   |
|              |          |  Stamp |                        |
|              |   76-76  |  X     | ' ' separating blank   |
| HIGH DATETIME|   77-95  |  Time  |  DDMMMYY:HH:MM:SS.SS   |
|              |          |  Stamp |                        |
|              |   96-96  |  X     | ')' closing parenthesis|
|              |   97-97  |  X     | ' ' separating blank   |
| FORCE/SELECT |   98-99  |  XX    | 'S(' for SELECT option |
|              |          |        | 'F(' for FORCE option  |
| LOW  DATETIME| 100-118  |  Time  |  DDMMMYY:HH:MM:SS.SS   |
|              |          |  Stamp |                        |
|              | 119-119  |  X     | ' ' separating blank   |
| HIGH DATETIME| 120-138  |  Time  |  DDMMMYY:HH:MM:SS.SS   |
|              |          |  Stamp |                        |
|              | 139-139  |  X     | ')' closing parenthesis|
|  The following fields are reserved.                       |
|              | 140-184  |  X     | Reserved               |
|              | 185-200  |  HEX   | Highest Timestamp in   |
|              |          |        | floating point notation|
+--------------+----------+--------+------------------------+
```

Figure 4-5. Checkpoint Database Update Time Range Record

The following example should help you read the checkpoint file's database update time range records.

An installation runs CA MICS on one processor identified as P033 in prefix.MICS.PARMS(SYSID). The site has been using CA MICS from January 1, 2001 to January 27, 2001 and is licensed for the following products:

- Batch and Operations Analyzer (component ID = SMF)

- Hardware and SCP Analyzer (component ID = RMF)

- Analyzer for TSO (component ID = TSO)

This site's unit checkpoint Database update time range records are:

```
    P033 RMF      01JAN01:00:00:10.03 27JAN01:23:59:46.00
    P033 SMF      01JAN01:00:00:02.05 27JAN01:23:30:23.00
    P033 TSO      01JAN01:00:00:01.11 27JAN01:23:58:12.00
```

The oldest data ever input into CA MICS was from January 1, 2001. The most recent data input into CA MICS was from January 27, 2001.

4.3.7.3 Duplicate Data Check Process

CA MICS provides two forms of duplicate data protection checks:

- Duplicate data within the same input stream is automatically dropped by each of the respective product's update processors. This process does not involve the checkpoint file.

- Duplicate input data occurs when data that has previously been input into the database update is again inadvertently input to the update process. If the input data falls within the time-range kept by ORGSYSID and product in a checkpoint file, it is automatically dropped, unless specific administrator action is signalled to force the data through.

While the examples and discussion in this section refer directly to the unit checkpoint file and the DAILY update job, the concepts and processing description apply equally to the individual, product specific incremental update checkpoint data sets.

The date/time ranges maintained in checkpoint file Database update time range records are matched by ORGSYSID and component combination. For systems such as IMS and CICS, the component check is extended to include a subsystem ID (e.g., IMSID or CICSID). Data is processed or dropped based on the following conditions:

- If an input record time stamp is greater than the checkpoint entry high time stamp and the date in the input record time stamp is not greater than today's date, the record is PROCESSED.

- If an input record time stamp is less than the checkpoint entry high time stamp, the record is DROPPED because it is considered duplicate data.

  Data dropped in this way may be input to the update process using the Force option.
- If the input record time stamp is less than the checkpoint entry low time stamp, the record is DROPPED because it was deemed to be generated before the product was installed.

  Data dropped in this way may be input to the update process using the Force option.
- If the input record time stamp date is greater than today's date, the record is DROPPED because it is considered to have been generated with an invalid IPL date.

  A warning is written to the log when "future" data is found. Very special care is required to bring this data into the database. Call CA MICS Product Support for assistance.

The following figure illustrates unit checkpoint file processing in the DAILY Database update job steps.

```
+-----------------------------------------------------------+
|                                                           |
|  CA MICS Daily Job Stream                                 |
|                                                           |
|  +---------------------------+                            |
|  |   Step DAYSMF in DAILY    |               +-------+    |
|  |  +---------------------+  |               |       |    |
|  |  | INPUTRDR's First    |<-+---------------|       |    |
|  |  | Action: Read in the |  |  Read Only    |       |    |
|  |  | unit checkpoint,    |  |               |       |    |
|  |  | and build SYSID-    |  |               |       |    |
|  |  | Component Time Range|  |               |       |    |
|  |  | Table.              |  |               |       |    |
|  |  +---------------------+  |               |   C   |    |
|  |  | Match time-stamps   |  |               |   H   |    |
|  |  | to SYSID-Component. |  |               |   E   |    |
|  |  | Drop data not       |  |               |   C   |    |
|  |  | passing range tests.|  |               |   K   |    |
|  |  +---------------------+  |               |   P   |    |
|  |                           |               |   O   |    |
|  |                           |               |   I   |    |
|  |                           |               |   N   |    |
|  |   Update Steps in Daily   |               |   T   |    |
|  |  +---------------------+  |               |       |    |
|  |  | Build SYSID/        |<-----------------|   F   |    |
|  |  | Component Time Range|  |  Read Only    |   I   |    |
|  |  | Table.              |  |               |   L   |    |
|  |  +---------------------+  |               |   E   |    |
|  |  | Update entries in   |  |               |       |    |
|  |  | the Time Range      |  |               |       |    |
|  |  | Table.              |  |               |       |    |
|  |  +---------------------+  |               |       |    |
|  |  | Write entire SYSID  |  |               |       |    |
|  |  | Component Table     |  |               |       |    |
|  |  | to unit checkpoint  |  |               |       |    |
|  |  | with new update     |<---------------->|       |    |
|  |  | time ranges.        |  |Update In-place|       |    |
|  |  +---------------------+  |               +-------+    |
|  +---------------------------+                            |
+-----------------------------------------------------------+
```

The following example illustrates the duplicate data check process.

Sample unit checkpoint Database update time range records for a system running one processor with the CA MICS SMF, RMF, TSO, and CICS products:

```
    1...5...10...15...20...25...30...35...40...45...50...55
    P033 RMF      01JAN01:00:00:10.03 27JAN01:23:59:46.00
    P033 SMF      01JAN01:00:00:02.05 27JAN01:23:30:23.00
    P033 TSO      01JAN01:00:00:01.11 27JAN01:23:58:12.00
    P033 CIC PROD 01JAN01:00:07:00.28 27JAN01:19:38:10.05
```

Using the above update time ranges, the CA MICS routine that first reads the data source matches each record read against its respective ORGSYSID and product time range to determine if the record should be selected or dropped. This process is performed by the INPUTRDR step (DAYSMF) and/or each of the update processor steps (001-190).

The selection process adheres to the following rules:

1. Each record is matched with its corresponding ORGSYSID and product in the table. If a new ORGSYSID-product is encountered, it is automatically added to the table.

   In this example, the ORGSYSID/product match is extended to include CICSID for the CICS product (PROD in this example).
2. If the input record's time stamp is greater than the checkpoint high time stamp and NOT GREATER THAN TODAY'S DATE, the record is selected for further processing.

   In this example (assuming today is January 28, 2001), SMF product input data record generated on January 27 would be accepted. In addition, data generated at 12:45 p.m. on January 27 would be processed because 27JAN01:23:45:00.00 (the input record time stamp) is greater than the checkpoint entry high time stamp (27JAN01:23:30:23.00).
3. If the SELECT option was specified and the input record's time stamp is outside of the SELECT range, the record is NOT selected for further processing.

4. Records that are not selected by the tests above are dropped. Note that the Force option can be used to allow processing of records dropped by the duplicate data check.

4.3.7.4 Using the Select Option for Input Data

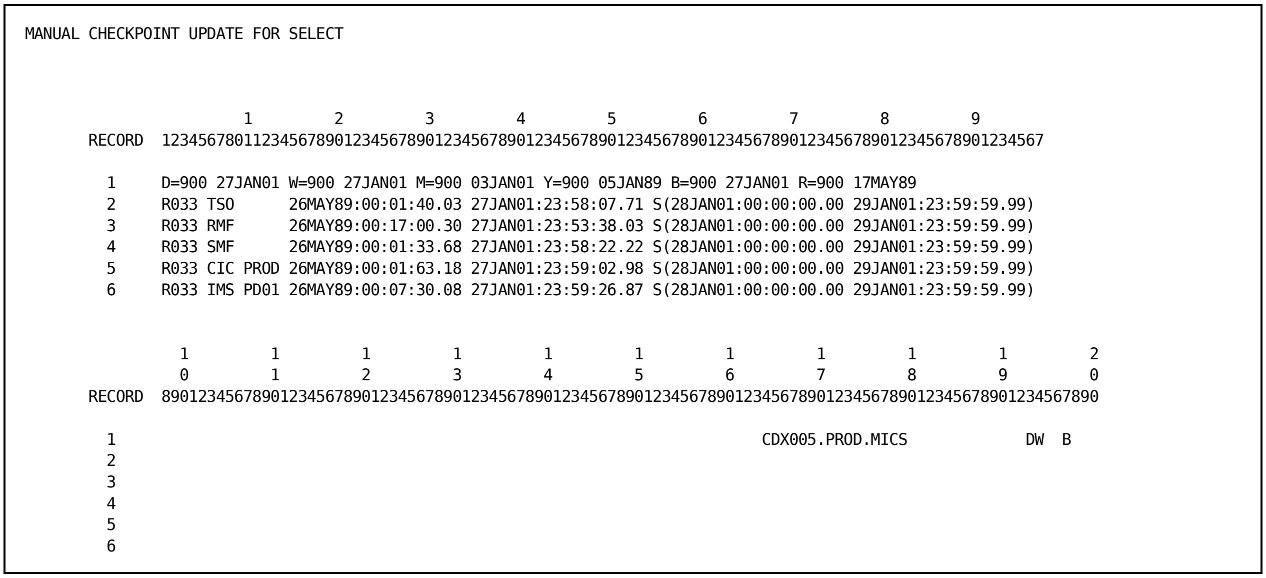
The Select option provides a means of selecting only a portion of the data on a given input file. The Select option is especially valuable for processing cumulative input files (e.g., SMF history files). For example, you can use the Select option to input into CA MICS just a single day from a week-to-date input tape. Select is often used in combination with the Force option when you need to load input data that was missing from the original Database update input stream.

Use the Operational Status and Tracking FORCE command to access the Select option. See the online tutorials and the Operational Status and Tracking section (4.3.4) of this guide for more information.

The unit checkpoint file is used for all Force and Select specifications. Select and Force parameters are automatically copied to the incremental update checkpoint file at the beginning of incremental update processing (the INCRccc operational job).

CA MICS automatically clears Select criteria from the unit checkpoint file when updating a checkpoint record. This technique avoids the problem of subsequent Database updates reusing the Select specifications.

NOTE: The incremental update INCRccc operational jobs access the unit checkpoint in read-only mode (DISP=SHR). At successful completion, INCRccc jobs will clear the unit checkpoint Select criteria for the corresponding product. This checkpoint update operation uses CA MICS ENQ/DEQ facilities to prevent concurrent unit checkpoint update by two or more processes, and thus it is possible that the unit checkpoint will not be available when the INCRccc job attempts the update. The INCRccc job will normally wait until the unit checkpoint data set is available for DISP=SHR update, but if the maximum wait time is exceeded, the INCRccc job will abort without removing the unit checkpoint Select criteria. If this occurs, you may need to manually clear the unit checkpoint file Select specification for this product. You can use the Operational Status and Tracking FORCE command, or you can edit prefix.MICS.CHECKPT.DATA to manually erase your Select specifications.



```
MANUAL CHECKPOINT UPDATE FOR SELECT



                        1         2         3         4         5         6         7         8         9
       RECORD  1234567801123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

         1     D=900 27JAN01 W=900 27JAN01 M=900 03JAN01 Y=900 05JAN89 B=900 27JAN01 R=900 17MAY89
         2     R033 TSO      26MAY89:00:01:40.03 27JAN01:23:58:07.71 S(28JAN01:00:00:00.00 29JAN01:23:59:59.99)
         3     R033 RMF      26MAY89:00:17:00.30 27JAN01:23:53:38.03 S(28JAN01:00:00:00.00 29JAN01:23:59:59.99)
         4     R033 SMF      26MAY89:00:01:33.68 27JAN01:23:58:22.22 S(28JAN01:00:00:00.00 29JAN01:23:59:59.99)
         5     R033 CIC PROD 26MAY89:00:01:63.18 27JAN01:23:59:02.98 S(28JAN01:00:00:00.00 29JAN01:23:59:59.99)
         6     R033 IMS PD01 26MAY89:00:07:30.08 27JAN01:23:59:26.87 S(28JAN01:00:00:00.00 29JAN01:23:59:59.99)


                 1         1         1         1         1         1         1         1         1         1         2
                 0         1         2         3         4         5         6         7         8         9         0
       RECORD  890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456790

         1                                                                 CDX005.PROD.MICS             DW  B
         2
         3
         4
         5
         6
```

You can also manually update the unit checkpoint file to add Select specifications.

Edit prefix.MICS.CHECKPT.DATA.

Activate the Select option by placing the time stamp range of the data to be selected, enclosed in parentheses and preceded by an 'S', in the checkpoint file's Database update time range record for the required ORGSYSID and product combination. When this is done, only records in the Select range are considered for input into the Database for the specified ORGSYSID and product. When the Checkpoint File is updated, any Select time ranges are automatically cleared to avoid their accidental misuse on succeeding executions.

The figure above shows a sample unit checkpoint as it would appear if the data for January 28, 2001 and January 29, 2001 were to be selected from all input data. Since the Database has been updated with data from January 27, any data after the highest date time stamp would normally be accepted.

4.3.7.5 Using the Force Option for Input Data

The Force option enables the System Administrator to override standard Database time-range checks. This option is useful in situations where a portion of data was unavailable for normal operational processing. Because the data was within the timespan of the data used to update the Database, it would no longer be allowed to pass the standard time-range tests on a succeeding update. Force enables the System Administrator to process data already encompassed by the Database's current time-ranges.

When you use Force, all records in the force date/time range you specify are accepted by the CA MICS duplicate data checks, even if the input data falls within the range of data contained in the Database.

When using Force, normal CA MICS duplicate data checks are bypassed. It is crucial that you take precautions avoid processing the same data multiple times. Use the Operational Status and Tracking HISTORY command or access the Input History Log File directly to determine the data ranges that have been used for previous updates.

You can use Force in combination with Select. If you specify the same date/time ranges for Select and Force, then Select will ensure that you process only the data you plan to Force into the CA MICS Database.

To simplify recovery should an error occur, ensure that an up-to-date Database backup is available when using Force.

Use the Operational Status and Tracking FORCE command to access the Force option. See the online tutorials and the Operational Status and Tracking section (4.3.4) of this guide for more information.

The unit checkpoint file is used for all Force and Select specifications. Select and Force parameters are automatically copied to the incremental update checkpoint file at the beginning of incremental update processing (the INCRccc operational job).

CA MICS automatically clears Force (and Select) criteria from the checkpoint file when updating a checkpoint record. This technique avoids the problem of subsequent Database updates reusing the Force specifications.

NOTE: The incremental update INCRccc operational jobs access the unit checkpoint in read-only mode (DISP=SHR). At successful completion, INCRccc jobs will clear the unit checkpoint Force (and Select) criteria for the corresponding product. This checkpoint update operation uses CA MICS ENQ/DEQ facilities to prevent concurrent unit checkpoint update by two or more processes, and thus it is possible that the unit checkpoint will not be available when the INCRccc job attempts the update. The INCRccc job will normally wait until the unit checkpoint data set is available for DISP=SHR update, but if the maximum wait time is exceeded, the INCRccc job will abort without removing the unit checkpoint Force (and Select) criteria. If this occurs, you may need to manually clear the unit checkpoint file Force and Select specifications for this product. You can use the Operational Status and Tracking FORCE command, or you can edit prefix.MICS.CHECKPT.DATA to manually erase your Force (and Select) specifications.

```
MANUAL CHECKPOINT UPDATE FOR FORCE


                        1         2         3         4         5         6         7         8         9
       RECORD  1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

         1     D=900 27JAN01 W=900 27JAN01 M=900 03JAN01 Y=900 05JAN89 B=900 27JAN01 R=900 17MAY89
         2     R033 TSO      26MAY89:00:01:40.03 27JAN01:23:58:07.71 F(25JAN01:00:00:00.00 25JAN01:23:59:59.99)
         3     R033 RMF      26MAY89:00:17:00.30 27JAN01:23:53:38.03 F(25JAN01:00:00:00.00 25JAN01:23:59:59.99)
         4     R033 SMF      26MAY89:00:01:33.68 27JAN01:23:58:22.22 F(25JAN01:00:00:00.00 25JAN01:23:59:59.99)
         5     R033 CIC PROD 26MAY89:00:01:63.18 27JAN01:23:59:02.98 F(25JAN01:00:00:00.00 25JAN01:23:59:59.99)
         6     R033 IMS PD01 26MAY89:00:07:30.08 27JAN01:23:59:26.87 F(25JAN01:00:00:00.00 25JAN01:23:59:59.99)


                 1         1         1         1         1         1         1         1         1         1         2
                 0         1         2         3         4         5         6         7         8         9         0
       RECORD  8901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

         1                                                                     CDX005.PROD.MICS             DW  B
         2     S(25JAN01:00:00:00.00 25JAN01:23:59:59.99)
         3     S(25JAN01:00:00:00.00 25JAN01:23:59:59.99)
         4     S(25JAN01:00:00:00.00 25JAN01:23:59:59.99)
         5     S(25JAN01:00:00:00.00 25JAN01:23:59:59.99)
         6     S(25JAN01:00:00:00.00 25JAN01:23:59:59.99)
```

You can also manually update the unit checkpoint file to add Force specifications.

Edit prefix.MICS.CHECKPT.DATA.

Activate the Force option by entering the range of the data to be forced, enclosed in parentheses and preceded by an 'F', in the checkpoint file Database update time range record for the required ORGSYSID and product combination. When Force is active, all records in the force range pass the date/time range check even if they fall within the range of data contained in the Database. When the unit checkpoint file is updated, any force time stamps are automatically cleared to avoid their accidental misuse on succeeding executions.

The figure above shows a sample unit checkpoint as it would appear if the data for January 25, 2001 were to be Forced into the Database. Since the Database has been updated with data from January 27, the data from January 25 would normally be rejected as duplicate data.

In the example, both Force and Select criteria were specified to ensure that only data for January 25 would be processed. We highly recommended that you use Select whenever you use Force.

4.3.8 System Restart and Recovery

This section discusses specific topics related to restarting CA MICS operational jobs and recovering the CA MICS database.

Refer to the Operational Processes, Jobs, and Steps section (4.3.3) of this guide for documentation on the operational jobs mentioned in this section.

RESTARTING CA MICS DATABASE UPDATE JOBS

The Operational Status and Tracking RESTART command handles restart for the DAILY, WEEKLY, MONTHLY, and YEARLY jobs. See the Operational Status and Tracking section (4.3.4).

You can also restart the DAILY, WEEKLY, MONTHLY, and YEARLY jobs manually by specifying the job statement RESTART= parameter. The CA MICS operational jobs are generated with RESTART=* on the job statement to indicate that processing should begin with the first step of the job.

o Edit prefix.MICS.RESTART.CNTL if processing was submitted by Operational Status and Tracking or the SCHEDULE job.

  or

  Edit prefix.MICS.CNTL(jjjjjjjj), where jjjjjjjj is DAILY/WEEKLY/MONTHLY/YEARLY, if processing was submitted manually or via your installation's batch scheduling facility.

o Change RESTART=* to RESTART=(stepname.MICS) on the job statement, where stepname is the operational step (e.g., DAY030) where processing should begin. The CA MICS Run Status Report lists the correct RESTART= parameter.

o If internal step restart is enabled for the batch job step where processing will be restarted, then processing will automatically resume at the last completed processing phase in this job step.

o If you need to override automatic internal step restart and force the step to start from the beginning, specify SYSPARM=NORESTART on the JCL EXEC statement for this batch job step.

o Submit the batch job and CANCEL the edit session. DO NOT SAVE THE MODIFIED JCL.

OVERRIDING WORK FILE DYNAMIC DATA SET ALLOCATION PARAMETERS

Internal step restart uses OS/390 dynamic allocation services to create new data sets and to access existing data sets. Data set allocation parameters are specified by product in prefix.MICS.PARMS(cccOPS) and permanent changes to data set allocation parameters (e.g., to increase the space allocation for the WORK data set) require both changing the cccOPS parameter and executing the corresponding cccPGEN job. However, in restart situations, such as when recovering from a production job abend, you may temporarily override data set allocation parameters for one or more dynamically allocated data sets by using the //PARMOVRD facility.

For example,

o If the SAS log indicates that the job failed due to a shortage of disk space on one of the WORKnn data sets (where nn is 01 - 99) or the cccXWORK data set (where ccc is the product associated with this database update step),

   - Edit the operational job JCL for the step that failed and add a //PARMOVRD DD stream containing the WORK and/or RESTARTWORK parameters to temporarily override the data set allocation parameters for the failing data sets to increase the space allocation. For example,

```
     //PARMOVRD DD *
      WORK   SPACE=(CYL,(50,50)) STORCLAS=MICSTEMP
      RESTARTWORK SPACE=(CYL,(50,50))
      RESTARTWORK STORCLAS=MICSTEMP
```

   - Restart the database update job step from the beginning by specifying,

```
    SYSPARM=NORESTART
```

      on the JCL EXEC statement.

   - After the job step completes successfully, remove the //PARMOVRD DD stream to resume using the data set allocation parameters you specified in prefix.MICS.PARMS(cccOPS). If you believe that the temporary change to the data set allocation parameters should be made permanent, then increase the amount of space requested on the cccOPS WORK (for WORKnn data sets) or RESTARTWORK (for the cccXWORK data set) parameter and run cccPGEN.

See section 2.3.6 of this guide for more information on the

//PARMOVRD facility.

RESTARTING CA MICS INCREMENTAL UPDATE JOBS

Operational Status and Tracking does NOT support the incremental update INCRccc jobs. You must restart INCRccc processing manually.

The incremental update INCRccc jobs always begin with the INCRnnn processing step, which is where you want to restart processing. Thus, you restart an INCRccc job by simply resubmitting it for execution. You can do this by editing prefix.MICS.CNTL(INCRccc) and entering the SUBMIT command, or you can use your installation's production batch job scheduling facilities to resubmit the INCRccc job.

The internal step restart considerations discussed previously apply equally to incremental update INCRccc jobs. For example, you can use the //PARMOVRD facility to temporarily increase WORKnn space allocations for an INCRccc job just as you can for a DAILY job DAYnnn step.

If you specified INCRDB TAPE or INCRDB DYNAM in prefix.MICS.PARMS(cccOPS), then OS/390 dynamic allocation services are used to create new IUDETAIL and IUDAYS data sets and to access existing data sets. Data set allocation parameters are specified by product in prefix.MICS.PARMS(cccOPS) and permanent changes to data set allocation parameters (e.g., to increase the space allocation for the IUDETAIL data set) require both changing the cccOPS parameter and executing the corresponding cccPGEN job. However, in restart situations, such as when recovering from a production job abend, you may temporarily override data set allocation parameters for one or more dynamically allocated data sets by using the //PARMOVRD facility.

For example, you might edit the INCRccc JCL and add a //PARMOVRD DD stream containing the INCRDETAIL and/or INCRDAYS parameters to temporarily override the data set allocation parameters for the failing data sets to increase the space allocation as follows:

```
     //PARMOVRD DD *
      INCRDETAIL  SPACE=(CYL,(50,50)) STORCLAS=MICSDASD
      INCRDAYS    SPACE=(CYL,(10,10)) STORCLAS=MICSDASD
```

After the job step completes successfully, remember to remove the //PARMOVRD DD stream to resume using the data set allocation parameters specified in prefix.MICS.PARMS(cccOPS).

If you believe that the temporary change to the data set allocation parameters should be made permanent, then increase the amount of space requested on the cccOPS INCRDETAIL or INCRDAYS parameter and run cccPGEN.

See section 2.3.6 of this guide for more information on the //PARMOVRD facility.

NOTE: If an INCRccc job fails with a U300 abend (less than 10 input records), internal step restart is active for this product, and you plan to resolve the error condition through use of the checkpoint FORCE/SELECT facility (to force processing of data older than the current checkpoint high-ENDTS), then you must also specify the EXEC statement SYSPARM=NORESTART parameter in the INCRccc job JCL to re-initialize the incremental update checkpoint file with the unit checkpoint FORCE/SELECT specifications.

DAYSMF WORK FILE RECOVERY AND RESTART

The DAILY job normally uses work data sets for passing data from the DAYSMF step to each following update processor step which takes input from SMF. If fewer than two CA MICS products take input from SMF, DAYSMF work files will not exist and DAILY will NOT have a DAYSMF step. Refer to section 2.3.3.2.1.1 for more information on the COMPONENTS and SMFRECORDING keywords of prefix.MICS.PARMS(JCLDEF).

If DAYSMF work files are used, these data sets are allocated either permanently or temporarily based on the parameters specified in prefix.MICS.PARMS(JCLDEF) using the keyword DAYSMF. The DAYSMF keyword is discussed in section 2.3.3.2.1.3.

Prior to restarting the DAILY job, verify that the DAYSMF work files are still available. The default data set names are

```
    prefix.MICS.ccc.DATA
```

where ccc is the component identifier (e.g., RMF). If the DAYSMF work files are missing, you must recreate them before restarting the DAILY job.

The Operational Status and Tracking RESTART command gives you the option to include DAYSMF temporary work file recovery at the front of the generated restart job. Simply reply YES to the prompt on the RESTART Unit Database panel. If the DAYSMF work files are defined as permanent (and have been scratched), refer to the manual restart instructions below. See the Operational Status and Tracking section (4.3.4) of this guide for more information.

For a manual restart, you must recover the DAYSMF work files manually before restarting DAILY processing. For example, if a DAILY restart is required and the DAYSMF work files have been scratched, use the DAYSMFR job to rebuild the DAYSMF work files; then restart the job. To submit the DAYSMFR job, enter the following command:

```
    SUB 'prefix.MICS.CNTL(DAYSMFR)'
```

NOTE: If the DAYSMF work files have been defined as permanent, then you must first allocate the DAYSMF work files as explained in Section 3.5.6.2, "CA MICS Database and File Allocation", before executing DAYSMFR. If the DAYSMF work files have been defined as temporary, then the allocation will be done by DAYSMFR.

NOTE: The DAYSMFR job will abort if one or more incremental update checkpoint files indicate that incremental or daily update processing is in progress or failed. In this situation, you may specify,

```
    SYSPARM=FORCE
```

on the DAYSMF step EXEC statement (in the DAYSMFR job) to override the incremental update checkpoint status check and "force" DAYSMFR job execution.

SPLITSMF OUTPUT FILE RECOVERY AND RESTART

If the INCRSPLIT USE option is specified for one or more products in the unit database (see cccOPS), then the SPLITSMF job is used to create tailored input data sets for the corresponding INCRccc jobs. The SPLITSMF job dynamically allocates, catalogs, and populates a prefix.MICS.ccc.IUSPLTDS data set for each product marked as INCRSPLIT USE. The corresponding INCRccc jobs dynamically allocate and read the corresponding SPLITSMF output data set. At successful end of job, the INCRccc job dynamically deletes its input data set.

If one or more prefix.MICS.ccc.IUSPLTDS data sets are damaged or deleted prior to INCRccc execution, the INCRccc job will fail and you will need to rerun the SPLITSMF job to recreate the data sets before restarting INCRccc processing. SPLITSMF will recreate ALL of the prefix.MICS.ccc.IUSPLTDS data sets as follows:

o If the data set is not found, then a new data set is allocated and populated.

o If the data set already exists, it will be allocated DISP=MOD and the "new" input data will be added to the end of the existing data.

The DISP=MOD processing of an existing data set is required to ensure that no input data is lost in cases where SPLITSMF is executed a second time prior to INCRccc execution. For example, if the availability of another incremental update input data set triggered SPLITSMF job submission before all INCRccc jobs completed processing the prior input data set, data would be lost if the new data simply replaced the existing data sets.

Standard CA MICS duplicate data elimination facilities shield the database from any duplicate data that might be introduced by SPLITSMF appending "new" data to an existing data set as follows:

o If SPLITSMF is run a second time before INCRccc processing completes, the input file will contain duplicate data which will be dropped by standard SORT NODUPS processing which has always been used to shield the database from accidental inclusion of duplicate input data in a single DAILY database update.

o If SPLITSMF is rerun using the same input SMF data stream after INCRccc processing for a product completes, then the next INCRccc job execution for this product will re-input data that has already been processed. Standard CA MICS checkpoint processing will drop this duplicate data because it is "older" than the current checkpoint high end timestamp (ENDTS).

NOTE: The SPLITSMF job will abort if one or more of the associated incremental update checkpoint files indicate that the INCRccc job is in progress or failed. If an INCRccc job failed due to a missing input data set and you are rerunning SPLITSMF to recreate this missing data set, specify,

```
SYSPARM=FORCE
```

on the SPLITSMF job step EXEC statement to override the incremental update checkpoint status check and "force" SPLITSMF job execution.

ACCOUNTING AUDIT FILE RECOVERY AND RESTART

The DAY199 step of the DAILY operational job updates Accounting and Chargeback month-to-date audit files. You must restore the Accounting and Chargeback ACTAUDIT DAY1 file to a status synchronized with the online CA MICS database prior to restarting the DAY199 step or after running the RESTORE job.

If you restart DAY199 or restore the database without restoring the ACTAUDIT DAY1 file, the ACTAUDIT files (DAY1 and DAY2) will contain duplicate data. See the Accounting and Chargeback User Guide for more information on the ACTAUDIT files.

The Operational Status and Tracking RESTART command gives you the option to include ACTAUDIT DAY1 file restore at the front of the generated restart job. Simply reply YES to the prompt on the RESTART Unit Database panel. See the Operational Status and Tracking section (4.3.4) of this guide for more information.

For a manual restart, you must restore the ACTAUDIT DAY1 file manually before restarting DAY199 processing. For example, if a restart is required in step DAY199, use the ACTDAY1R job to restore the ACTAUDIT DAY1 file; then restart the job. To submit the ACTDAY1R job, enter the following command:

```
    SUB 'prefix.MICS.CNTL(ACTDAY1R)'
```

You must submit the ACTDAY1R job manually after Database RESTORE processing. Refer to the Accounting and Chargeback User Guide for more information.

RESTARTING OTHER JOBS

Restart the SPLITSMF job from the beginning. Simply resubmit prefix.MICS.CNTL(SPLITSMF) with the same JCL overrides (or temporary changes). If you are restarting SPLITSMF to recreate a missing prefix.MICS.ccc.IUSPLTDS data set and INCRccc failed due to the missing prefix.MICS.ccc.IUSPLTDS, then specify SYSPARM=FORCE on the SPLITSMF step EXEC statement to override the standard abort if an INCRccc job failure is detected.

Restart the AUDIT job from the beginning. Simply resubmit prefix.MICS.CNTL(AUDIT) with the same JCL overrides (or temporary changes).

Restart the HISTW job from the beginning. Simply resubmit prefix.MICS.CNTL(HISTW) with the same JCL overrides (or temporary changes).

Restart the HISTM job from the beginning. Simply resubmit prefix.MICS.CNTL(HISTM) with the same JCL overrides (or temporary changes).

Restart the RSTATUS job from the beginning. Issue the Operational Status and Tracking RSTATUS command again or submit prefix.MICS.CNTL(RSTATUS).

Restart the BACKUP job from the beginning. Issue the Operational Status and Tracking BACKUP command again or submit prefix.MICS.CNTL(BACKUP).

Restart the SCHEDULE job from the beginning. Simply resubmit prefix.MICS.CNTL(SCHEDULE) with the same JCL overrides (or temporary changes).

Restart the RESTORE job from the beginning. Issue the Operational Status and Tracking RESTORE command again or submit prefix.MICS.CNTL(RESTORE).

If RESTORE fails in the pre-restore BACKUP step, you can bypass pre-restore backup processing.

o Use the Operational Status and Tracking RESTORE command with the NOBACKUP parameter. For example, to restore the P (PRIMARY) unit database without the pre-restore backup, enter:

```
RESTORE P NOBACKUP
```

o For a manual restore, edit prefix.MICS.CNTL(RESTORE) and specify RESTART=RSTR900.MICS on the job statement. DO NOT SAVE THE MODIFIED JCL.

Refer to the Operational Processes, Jobs, and Steps section (4.3.3) of this guide for more information on the RESTORE operational job.

Restart the DAYSMFR job from the beginning. Simply resubmit prefix.MICS.CNTL(DAYSMFR) with the same JCL overrides (or temporary changes). If the DAYSMFR step fails with messages indicating that one or more incremental or daily update steps failed, then specify SYSPARM=FORCE on the DAYSMF step EXEC statement (in the DAYSMFR job) to override the standard abort if an incremental update failure is detected.

Restart the ACTDAY1R job from the beginning. Simply resubmit prefix.MICS.CNTL(ACTDAY1R) with the same JCL overrides (or temporary changes).

Restart the RSTRTBLS job from the beginning. Simply resubmit prefix.MICS.CNTL(RSTRTBLS) with the same JCL overrides (or temporary changes).

Restart the RSTRTLIB job from the beginning. Simply resubmit prefix.MICS.CNTL(RSTRTLIB) with the same JCL overrides (or temporary changes).

Restart the IUDBINIT job from the beginning. Simply resubmit prefix.MICS.CNTL(IUDBINIT) with the same JCL overrides (or temporary changes).

MANUAL RESTART EXAMPLE

The following is an example of manually restarting CA MICS database update processing.

The SCHEDULE job determined that the WEEKLY process is due and submitted a generated job for daily database update, weekly cycle closeout, and database backup. Processing failed.

The CA MICS Run Status Report shows that processing failed in step WEEK300 with an S001 abend. Database aging is NOT in progress, so restart is possible. Further analysis shows an I/O error on an archive tape. The operations staff determines that the problem is due to a hardware failure and the failing tape drive is varied off the system.

You can now restart CA MICS processing.

o Edit the SCHEDULE Restart File, which is generally named prefix.MICS.RESTART.CNTL.

o Find the RESTART parameter on the job statement.

o Change the RESTART statement to read RESTART=(WEEK300.MICS).

o Submit the job from within ISPF Edit. DO NOT SAVE THE MODIFIED JCL.

4.3.9 CA MICS DDnames

The figures in this section summarize the z/OS ddnames used in CA MICS cataloged procedures, in the generated CA MICS operational jobs, and in dynamic allocation within the CA MICS programs. See the product guides for ddnames used by individual CA MICS products.

Note: DDnames used by the DAILYRPT, WEEKRPT, and MONTHRPT jobs are shown under the DAILY, WEEKLY, and MONTHLY columns.

DDnames used by the INCRccc jobs are shown under the DAILY column.

DDnames used only in generation jobs are shown under the DAILY column.

```
+------------------------------------------------------------+
|        |  JOB XREF   |                                      |
|        | | | | | | |S|                                      |
|        | | |M| | |R|C|                                      |
|        | |W|O|Y|B|E|H|                                      |
|        |D|E|N|E|A|S|E|                                      |
|        |A|E|T|A|C|T|D|                                      |
|        |I|K|H|R|K|O|U|                                      |
|        |L|L|L|L|U|R|L|                                      |
| DDNAME |Y|Y|Y|Y|P|E|E| File Description                     |
+--------+-+-+-+-+-+-+-+--------------------------------------+
|STEPLIB |Y|Y|Y|Y|Y|Y|Y| Execution Library                    |
|MICSLOG |Y|Y|Y|Y|Y|Y|Y| CA MICS Log Output File              |
|SYSLIST |Y|Y|Y|Y|Y|Y|Y| SYSOUT File for Standard Reports     |
|SYSLOG  |Y|Y|Y|Y|Y|Y|Y| SYSOUT File for the SAS Log Report   |
|SASAUTOS|Y|Y|Y|Y|Y|Y|Y| SAS Autocall Macro Libraries         |
|SASSNAP |Y|Y|Y|Y|Y|Y|Y| SAS Snap Dump Data Set               |
|SASHELP |Y|Y|Y|Y|Y|Y|Y| SAS Help Library                     |
|SASMSG  |Y|Y|Y|Y|Y|Y|Y| SAS Messages Library                 |
|CONFIG  |Y|Y|Y|Y|Y|Y|Y| SAS Configuration Parameters         |
|WORK    |Y|Y|Y|Y|Y|Y|Y| Work File for Interim SAS Files      |
|WORKnn  |Y| | | | | | | CA MICS Product Interim SAS Work     |
|        | | | | | | | |   Files                              |
|DETAIL  |Y|Y|Y|Y|Y|Y| | Database DETAIL Timespan             |
|DAYS    |Y|Y|Y|Y|Y|Y| | Database DAYS Timespan               |
|WEEKS   |Y|Y|Y|Y|Y|Y| | Database WEEKS Timespan              |
|MONTHS  |Y|Y|Y|Y|Y|Y| | Database MONTHS Timespan             |
|YEARS   |Y|Y|Y|Y|Y|Y| | Database YEARS Timespan              |
|CHECKPT |Y|Y|Y|Y|Y|Y| | CA MICS Unit Checkpoint File         |
+------------------------------------------------------------+
```

Figure 4-6. CA MICS DDnames (Part 1 of 3)

```
+-----------------------------------------------------------+
|        |  JOB XREF   |                                    |
|        | | | | | | |S|                                    |
|        | | |M| | |R|C|                                    |
|        | |W|O|Y|B|E|H|                                    |
|        |D|E|N|E|A|S|E|                                    |
|        |A|E|T|A|C|T|D|                                    |
|        |I|K|H|R|K|O|U|                                    |
|        |L|L|L|L|U|R|L|                                    |
| DDname |Y|Y|Y|Y|P|E|E| File Description                   |
+--------+-+-+-+-+-+-+-+------------------------------------+
|IUDETAIL|Y| | | | | | | Incremental Update DETAIL Timespan |
|IUDAYS  |Y| | | | | | | Incremental Update DAYS Timespan   |
|IUCKPT  |Y| | | | | | | Incremental Update Checkpoint File |
|FT15F00n|Y|Y|Y|Y|Y|Y|Y| CA MICS OS Work Files (n=1-6)      |
|SYSOUT  |Y|Y|Y|Y|Y|Y|Y| SYSOUT File for SORT Messages      |
|SORTLIB |Y|Y|Y|Y|Y|Y|Y| Sort Execution Program Library     |
|SORTCNTL|Y|Y|Y|Y|Y|Y|Y| DFSORT or CASORT Parameters        |
|$ORTPARM|Y|Y|Y|Y|Y|Y|Y| SYNCSORT Parameters                |
|SORTWKnn|Y|Y|Y|Y|Y|Y|Y| Sort Work Files                    |
|SASIN   |Y|Y|Y|Y|Y|Y|Y| SAS Compiler Initialization File   |
|SYSIN   |Y|Y|Y|Y|Y|Y|Y| SAS Compiler Input File            |
|MCOLIB  |Y|Y|Y|Y|Y|Y|Y| CA MICS Complex Object Library     |
|MUOLIB  |Y|Y|Y|Y|Y|Y|Y| CA MICS Unit Object Library        |
|INCLLIB |Y|Y|Y|Y|Y|Y|Y| sharedprefix.MICS.INCLLIB          |
|LOAD    |Y|Y|Y|Y|Y|Y|Y| prefix.MICS.LOAD                   |
|PARMS   |Y|Y|Y|Y|Y|Y|Y| prefix.MICS.PARMS                  |
|SHRPARMS|Y|Y|Y|Y|Y|Y|Y| sharedprefix.MICS.PARMS            |
|SOURCE  |Y|Y|Y|Y|Y|Y|Y| sharedprefix.MICS.SOURCE           |
|USOURCE |Y|Y|Y|Y|Y|Y|Y| prefix.MICS.USER.SOURCE            |
|PREFIX  |Y|Y|Y|Y|Y|Y|Y| prefix.MICS.USER.SOURCE(PREFIX)    |
|GENLIB  |Y|Y|Y|Y|Y|Y| | sharedprefix.MICS.SOURCE           |
|HPARMS  |Y|Y|Y|Y|Y|Y| | sharedprefix.MICS.HOLD.PARMS       |
|PROTOLIB|Y|Y|Y|Y|Y|Y| | sharedprefix.MICS.PROTOLIB         |
|CLIST   |Y|Y|Y|Y|Y|Y| | sharedprefix.MICS.CLIST            |
|MWFTABL |Y|Y|Y|Y|Y|Y| | sharedprefix.MICS.ISPTLIB (write)  |
|MAFISTL |Y|Y|Y|Y|Y|Y| | sharedprefix.MICS.ISPTLIB          |
|MWFTLIBR|Y|Y|Y|Y|Y|Y| | sharedprefix.MICS.ISPTLIB (read)   |
|ISPTLIB |Y|Y|Y|Y|Y|Y| | sharedprefix.MICS.ISPTLIB + ISPF   |
|ISPPLIB |Y|Y|Y|Y|Y|Y| | sharedprefix.MICS.ISPPLIB/ISPHLIB  |
|ISPMLIB |Y|Y|Y|Y|Y|Y| | sharedprefix.MICS.ISPMLIB + ISPF   |
|ISPSLIB |Y|Y|Y|Y|Y|Y| | sharedprefix.MICS.ISPSLIB          |
|ISPLOG  |Y|Y|Y|Y|Y|Y| | ISPF Log                           |
|ISPCTL1 |Y|Y|Y|Y|Y|Y| | ISPF Output Control File           |
|ISPFILE |Y|Y|Y|Y|Y|Y| | ISPF File Tailoring Output File    |
|ISPPROF |Y|Y|Y|Y|Y|Y| | ISPF Profile                       |
|SYSTSPRT|Y|Y|Y|Y|Y|Y| | Batch TMP Print File               |
|SYSTSIN |Y|Y|Y|Y|Y|Y| | Batch TMP Input File               |
+-----------------------------------------------------------+
```

Figure 4-6. CA MICS DDnames (Part 2 of 3)

```
+-------------------------------------------------------------+
|        |  JOB XREF   |                                    |
|        | | | | | | |S|                                    |
|        | | |M| | |R|C|                                    |
|        | |W|O|Y|B|E|H|                                    |
|        |D|E|N|E|A|S|E|                                    |
|        |A|E|T|A|C|T|D|                                    |
|        |I|K|H|R|K|O|U|                                    |
|        |L|L|L|L|U|R|L|                                    |
| DDname |Y|Y|Y|Y|P|E|E| File Description                   |
+--------+-+-+-+-+-+-+-+------------------------------------+
|LOCKIT  |Y|Y|Y|Y|Y|Y| | Internal Processing Control        |
|ULOCKIT |Y| | | | | | | Internal Unit-Level Control        |
|ccc     |Y| | | | | | | Interim Data Set for ccc Component |
|INPUTSMF|Y| | | | | | | Input SMF Data Set                 |
|INPUTccc|Y| | | | | | | Input Data Set for ccc Component   |
|SMFDRCTR|Y| | | | | | | Input CA SMF Director duplicate    |
|        | | | | | | | |   split Index                      |
|cccXCKPT|Y| | | | | | | Internal Step Restart Control File |
|        | | | | | | | |   for ccc Component                |
|cccXWORK|Y| | | | | | | Internal Step Restart Interim Work |
|        | | | | | | | |   Files for ccc Component          |
|IUKccc  |Y| | | | | | | ccc Component Incremental Update   |
|        | | | | | | | |   Checkpoint Allocated Dynamically |
|IUSccc  |Y| | | | | | | ccc Component SPLITSMF job output  |
|        | | | | | | | |   data set Allocated Dynamically   |
|IU*TAPE*|Y| | | | | | | Incremental Update INCRDB TAPE     |
|        | | | | | | | | Files                              |
|IUTEMPCK|Y| | | | | | | DAYSMF temporary checkpoint file   |
|        | | | | | | | |   used with Incremental Update     |
|PARMOVRD|Y| | | | | | | Dynamic Allocation Parm. Overrides |
|AUDIT*  | |Y| | | | | | Archive Audit GDGs                 |
|        | | | | | | | |   In WEEK300 step or the AUDIT job |
|HISTW*  | |Y| | | | | | Archive Weekly History GDGs        |
|        | | | | | | | |   In WEEK300 step or the HISTW job |
|HISTM*  | | |Y| | | | | Archive Monthly History GDGs       |
|        | | | | | | | |   In MONTH300 step or the HISTM job|
|BACKUPDT|Y| | | |Y|Y| | Online DETAIL Timespan Backup GDG  |
|BACKUPDY|Y| | | |Y|Y| | Online DAYS Timespan Backup GDG    |
|BACKUPWK|Y| | | |Y|Y| | Online WEEKS Timespan Backup GDG   |
|BACKUPMN|Y| | | |Y|Y| | Online MONTHS Timespan Backup GDG  |
|BACKUPYR|Y| | | |Y|Y| | Online YEARS Timespan Backup GDG   |
|JOBSTRM | | | | | | |Y| OS Data Set for Storing Job Stream |
+-------------------------------------------------------------+
```

Figure 4-6. CA MICS DDnames (Part 3 of 3)

IU*TAPE* refers to the optional Incremental Update tape data sets for the INCRDB TAPE option. Tape DDs are allocated for DETAIL and DAYS timespan input (0) generation and output (+1) generation tape data sets. The ddnames are:

IUXTAPEI, IUXTAPEO, IUDTAPEI, IUDTAPEO.

AUDIT*, HISTW*, and HISTM* refer to the series of ddnames used for the archive audit, weekly history, and monthly history generation data groups, respectively. These names follow the naming convention:

DDname	File Description
AUiiifff	Archive audit output
HWiiifff	Weekly archive history output
HMiiifff	Monthly archive history output
HIiiifff	Archive history master file input

4.3.10 Data Set Names

This section summarizes the data sets and data set names used by the CA MICS system. You can change the names of the data sets as described in the CA MICS User Names Table (JCLNAMES) section (2.3.3.3.2.3) of this guide.

CA MICS uses four types of data sets:

o shared data sets used by all unit databases

o data sets specific to one unit database

o tape-resident (archival) data sets specific to one unit database

o MVS system data sets

Shared data sets used by all CA MICS database units are:

```
          sharedprefix.MICS.ASM
          sharedprefix.MICS.BIN
          sharedprefix.MICS.CLIST
          sharedprefix.MICS.CNTL
          sharedprefix.MICS.DIC.TEXT
          sharedprefix.MICS.DOC.TEXT
          sharedprefix.MICS.DTF.INDEX
          sharedprefix.MICS.GENLIB
          sharedprefix.MICS.HOLD.PARMS
          sharedprefix.MICS.HOLD.USER.SOURCE
          sharedprefix.MICS.INC.TEXT
          sharedprefix.MICS.INCLLIB
          sharedprefix.MICS.ISPHLIB
          sharedprefix.MICS.ISPMLIB
          sharedprefix.MICS.ISPPLIB
          sharedprefix.MICS.ISPSLIB
          sharedprefix.MICS.ISPTLIB
          sharedprefix.MICS.LIBLOAD.CNTL
          sharedprefix.MICS.LOAD
          sharedprefix.MICS.LOCALMOD.CNTL
          sharedprefix.MICS.MACAUTOS
          sharedprefix.MICS.MACLIB
          sharedprefix.MICS.MCOLIB
          sharedprefix.MICS.MWFPDS.DATA
          sharedprefix.MICS.MWFSAS.DATA
          sharedprefix.MICS.OBJ
          sharedprefix.MICS.PARMS
          sharedprefix.MICS.PROTOLIB
          sharedprefix.MICS.SASFLS
```

```
sharedprefix.MICS.SCREENS
sharedprefix.MICS.SOURCE
sharedprefix.MICS.TABLES
sharedprefix.MICS.TEST.INCLLIB
sharedprefix.MICS.TEST.SOURCE
sharedprefix.MICS.USER.SOURCE
```

DASD data sets for use with a specific database are:

```
prefix.MICS.CHECKPT.DATA
prefix.MICS.CNTL
prefix.MICS.IMSSUS1
prefix.MICS.IMSSUS2
prefix.MICS.MODEL
prefix.MICS.MUOLIB
prefix.MICS.PARMS
prefix.MICS.RESTART.CNTL
prefix.MICS.USER.SOURCE
prefix.MICS.SPECIAL.SOURCE

prefix.MICS.DETAIL
prefix.MICS.DAYS
prefix.MICS.WEEKS
prefix.MICS.MONTHS
prefix.MICS.YEARS
```

The DAYSMF step of the DAILY job splits the input data into work data sets for the CA MICS products in the unit database. The work data sets are named as follows, where ccc is the corresponding component ID:

```
prefix.MICS.ccc.DATA
```

See the prefix.MICS.PARMS(JCLDEF) SMFRECORDING parameter for a list of the ccc values for a unit database. Your prefix.MICS.PARMS(JCLDEF) DAYSMF parameter specification controls whether the DAYSMF work data sets are permanent data sets or whether they are scratched after the DAILY job completes.

When incremental update is enabled for a Data Integration Application, the corresponding INCRccc and DAILY database update job steps allocate the following data sets:

```
prefix.MICS.ccc.IUCKPT
prefix.MICS.ccc.IUDETAIL
prefix.MICS.ccc.IUTAPE
tapeprefix.MICS.ccc.IUXTAPE.GggggV00
tapeprefix.MICS.ccc.IUDTAPE.GggggV00
```

Note: In the list above, ccc is the component ID.

The IUXTAPE and IUDTAPE Generation Data Group data sets apply to the optional INCRDB TAPE option.

The SPLITSMF job splits the input data into work data sets for processing in the INCRccc jobs of those CA MICS products in the unit database that have both INCRUPDATE YES and INCRSPLIT USE parameters specified in the cccOPS parameter member. The work data sets are named as follows, where ccc is the corresponding component ID:

```
prefix.MICS.ccc.IUSPLTDS
```

See the prefix.MICS.PARMS(JCLDEF) SMFRECORDING parameter for a list of the potential ccc values for a unit database. Then check prefix.MICS.PARMS(cccOPS) for the INCRUPDATE YES and INCRSPLIT USE options.

When internal step restart is enabled for a Data Integration Application, the corresponding DAILY database update job step dynamically allocates the following data sets:

```
prefix.MICS.cccXCKPT
prefix.MICS.cccXWORK
prefix.MICS.cccWRKnn
```

Note: In the list above, ccc is the component ID and nn the work file number (01 - 99).

If default archiving definitions are used, the archive tape data sets used by a specific unit are:

```
tapeprefix.MICS.AUDIT.iiifff.GggggV00
tapeprefix.MICS.BACKUP.tttttt.GggggV00
tapeprefix.MICS.BACKUP.CHECKPT.GggggV00
tapeprefix.MICS.HISTW.iiifff.GggggV00
tapeprefix.MICS.HISTM.iiifff.GggggV00
tapeprefix.MICS.MBACKUP.tttttt.GggggV00
tapeprefix.MICS.MBACKUP.CHECKPT.GggggV00
```

Note: In the list above, iiifff represents a CA MICS file name and tttttt is a time-span or can be SCREENS, TABLES, CAPACITY, CIMANAGE, or ISPTLIB.

MVS system data sets that are updated for CA MICS are:

- MVS system or user catalog

- SYS1.PROCLIB or other cataloged procedure data set

Another way to look at the data sets listed above is to consider their functions in a running CA MICS system. CA MICS data sets can be subdivided by what they contain:

- Programs in both source and load module form. With SAS, programs are run from their source version. The load libraries are used for CA MICS routines coded in IBM assembly language.

- Data, both online and offline.

- Documentation in both machine-readable and printable form.

- Control Information, whether maintained in CA MICS files or MVS system files (for example, prefix.MICS.PARMS or SYS1.PROCLIB).

- Test Libraries for testing modifications to CA MICS.

Each of these functions is described in detail in the sections below.

PROGRAMS

sharedprefix.MICS.ASM

A source library containing the source for some modules in the CA MICS system written in assembly language.

sharedprefix.MICS.BIN

A binary object distribution library for CA MICS distributed system, UNIX Systems Services (USS), and facilities related to CA's U.S. Patented Neugents(R) technology. The members in this library are generally "downloaded" to distributed systems and/or UNIX Systems Services HFS sub-directories.

sharedprefix.MICS.MACLIB

A source library containing the assembly language macros for the CA MICS system.

```
sharedprefix.MICS.OBJ

  An OS/390 object module distribution library for CA MICS
  distributed system, UNIX Systems Services (USS), and
  Neugents related facilities.  The members in this library
  are frequently processed by the OS/390 linkage editor on
  your system to install CA MICS facilities that operate
  under UNIX Systems Services.

sharedprefix.MICS.LOAD

  A load module library containing routines that perform
  CA MICS functions not implementable in SAS (such as those
  that update the Checkpoint data set in place), CA MICS
  extensions to the SAS language (such as new FORMATs and
  INFORMATs), and CA MICS extensions to IBM's ISPF (such as
  MADA).

sharedprefix.MICS.ISPxLIB

  A set of libraries containing the ISPF panels, messages,
  tutorials, and tables used by the CA MICS online
  facilities.

  These libraries include:

    sharedprefix.MICS.ISPHLIB - ISPF tutorial and help panels
    sharedprefix.MICS.ISPMLIB - ISPF messages
    sharedprefix.MICS.ISPPLIB - ISPF panels
    sharedprefix.MICS.ISPSLIB - ISPF skeletons
    sharedprefix.MICS.ISPTLIB - ISPF tables

sharedprefix.MICS.INCLLIB

  A library containing SAS source members that consist
  primarily of SAS %INCLUDE statements.  These members are
  used in place of concatenated SYSIN DD statements to allow
  conditional selection of SAS modules.

sharedprefix.MICS.MACAUTOS

  A library containing the source for SAS macros in the
  CA MICS system.  This is a SAS autocall macro library.

sharedprefix.MICS.SOURCE

  A library containing the source for modules in the CA MICS
  system written in SAS.
```

sharedprefix.MICS.USER.SOURCE

A library containing complex-level source code modules intended to be modified by the user (e.g., those that generate exception reports) and SAS macros produced as part of the CA MICS installation and generation processes.

sharedprefix.MICS.HOLD.USER.SOURCE

A source library containing the source modules copied by the COPYLIBS and COPYLIBU jobs when a new unit database is created.

prefix.MICS.USER.SOURCE

A library containing unit-level source code modules intended to be modified by the user (e.g., those that generate exception reports) and SAS macros produced as part of the CA MICS installation and generation processes.

sharedprefix.MICS.CLIST

A library containing source modules other than assembler, SAS, or ISPF dialog manager code.

DATA (Online)

The following data sets comprise the CA MICS database. If you must break up any of the following data sets in order to provide enough space to hold all of your data, refer to the CA MICS Database Split Table (DBSPLIT) section (2.3.3.3.2.4) of this guide. For products that have their own guides, refer to the information contained in those guides for information specific to the product.

sharedprefix.MICS.ACTAUDIT

A data set used by CA MICS Accounting and Chargeback. Refer to the product guide for more information about this data set.

sharedprefix.MICS.CAPACITY

A SAS-managed data set containing files used by the CA MICS Capacity Planner. See the product guide for more information about this data set.

sharedprefix.MICS.DTF.INDEX

A sequential data set used as a directory of all Comma Separated Value (CSV) extract files available to other applications. See the CA MICS Q&R Workstation User Guide for more information about this data set.

sharedprefix.MICS.TABLES

A SAS-managed data set containing data for CA MICS files in the Accounting and Operations information areas. These files are not organized by timespan; data is entered into the files manually or by monthly Accounting and Chargeback processing.

sharedprefix.MICS.MWFPDS.DATA

A library used by the CA MICS Information Center Facility (MICF) Shared Inquiry Output Catalog feature to save printed reports and the SAS log from CA MICS inquiry execution for later review.

sharedprefix.MICS.MWFSAS.DATA

A SAS-managed data set used by the CA MICS Information Center Facility (MICF) Shared Inquiry Output Catalog feature to save color graphics from MICF inquiry execution for later review.

prefix.MICS.DETAIL

A SAS-managed data set holding all of the SAS files that make up the DETAIL timespan of the CA MICS database.

prefix.MICS.DAYS

A SAS-managed data set holding all of the SAS files that make up the DAYS timespan of the CA MICS database.

prefix.MICS.WEEKS

A SAS-managed data set holding all of the SAS files that make up the WEEKS timespan of the CA MICS database.

prefix.MICS.MONTHS

A SAS-managed data set holding all of the SAS files that make up the MONTHS timespan of the CA MICS database.

prefix.MICS.YEARS

A SAS-managed data set holding all of the SAS files that make up the YEARS timespan of the CA MICS database.

Summarizing data at the YEARS level is optional. A small YEARS file is kept in the database if you set the YEARS timespan to inactive in prefix.MICS.PARMS(SITE) or in sharedprefix.MICS.PARMS(CPLXDEF). The YEARS file contains empty cycles (or zero observations) of CA MICS files supported at the YEARS timespan.

If you set the YEARS timespan to inactive by setting the YEARS timespan flag to No for each file in sharedprefix.MICS.GENLIB(cccGENIN) (where ccc is the three-letter component identifier), no YEARS files are kept in the database.

Incremental Update Data Sets

When incremental update is enabled for a Data Integration Application, the corresponding INCRccc and DAILY database update job steps allocate the data sets listed below. Depending on the options selected, these data sets can be one of the following:

- Permanent.
- Allocated at the beginning of the INCRccc or DAILY job step and then deleted after being copied to tape.
- Allocated at the beginning of the day and then deleted when the DAILY job completes successfully.

Note: ccc is the Data Integration Application Component ID.

prefix.MICS.ccc.IUDETAIL

A SAS-managed data set holding the current day's incremental update DETAIL timespan SAS files for the ccc product.

prefix.MICS.ccc.IUDAYS

A SAS-managed data set holding the current day's incremental update DAYS timespan SAS files for the ccc product.

prefix.MICS.ccc.IUCKPT

The ccc product's incremental update checkpoint file is used to keep track of the status of incremental update processing for this product. Both the status of the last incremental update step to complete and the low and high date/time stamps associated with input data processed to-date are kept here. This data set allows CA MICS to maintain database integrity during incremental update processing.

Internal Step Restart Data Sets

When internal step restart is enabled for a Data Integration Application, the corresponding DAILY database update job step dynamically allocates the following data sets.

These data sets are allocated and cataloged at the beginning of the database update job step, and then are deleted at normal end of job. However, if the job step abends, the data sets are retained for restart.

Note: ccc is the Data Integration Application Component ID.

prefix.MICS.cccXCKPT

A SAS-managed data set holding the SAS files that comprise the internal step restart checkpoint for this database update step. These include the internal step restart control file, the internal step restart data set status file, and last completed database update processing phase checkpoint files.

prefix.MICS.cccXWORK

The SAS work file for this database update step.

prefix.MICS.cccWRKnn

The multiple work file support data sets for this database update step, where nn is the work file number (01 - 99).

DATA (Offline)

All of the following data sets are named using the tapeprefix to associate them with other data sets specific to one CA MICS unit database.

The value of tapeprefix is set by the TAPEPREFIX option of prefix.MICS.PARMS(JCLDEF). You can set tapeprefix equal to prefix or it can be different if required to meet your site's JCL standards for the naming of data sets stored on tape.

CA MICS packs multiple, related tape data sets onto a single tape in order to conserve tape volumes.

In the following data set names:

```
o  timespan is either DETAIL, DAYS, WEEKS, MONTHS, YEARS,
             CAPACITY, CIMANAGE, ISPTLIB, SCREENS, or TABLES

o  gggg      is the generation number

o  nn        is the volume number

o  iiifff    is the name of a CA MICS file (e.g., HARCPU)

o  ccc       is the CA MICS product identifier (e.g., SMF)
```

tapeprefix.MICS.BACKUP.timespan.GggggVnn

CA MICS backs up the online database as specified in prefix.MICS.PARMS(JCLDEF). Each time a backup is made, a new Generation Data Group (GDG) catalog entry is created for each of the timespans (DETAIL, DAYS, WEEKS, MONTHS, YEARS). The Checkpoint data set associated with the online database is also backed up since it must be kept synchronized with the database. For the BACKUP job in the PRIMARY unit, complex data sets are also backed up. These include the sharedprefix.MICS CAPACITY, CIMANAGE, ISPTLIB, SCREENS, and TABLES data sets.

tapeprefix.MICS.MBACKUP.timespan.GggggVnn

CA MICS takes an additional backup of the online database and its checkpoint file each month as an added precaution against catastrophic data loss. These data sets are patterned after the normal backup data sets discussed above.

tapeprefix.MICS.AUDIT.iiifff.GggggVnn

CA MICS supports the creation of archive audit files. This support allows you to keep information about individual units available for far longer periods of time than can be managed in the online database.

If ARCHIVE AUDIT YES is specified in prefix.MICS.PARMS(JCLDEF), the AUDIT flag is set to Yes in sharedprefix.MICS.GENLIB(cccGENIN), and audit archive is active for this file in prefix.MICS.PARMS(DBMODEL); then each week thereafter, a data set with a name in the form shown above is created. This data set contains a copy of the entire week's DETAIL-level data (or DAYS-level data if the DETAIL timespan is not supported) for user-related (as opposed to system-related) files.

tapeprefix.MICS.HISTW.iiifff.GggggVnn

CA MICS supports the creation of archive history files. This support allows you to keep information in the WEEKS timespan available for far longer periods of time than can be managed in the online database.

If ARCHIVE HISTW YES is specified in prefix.MICS.PARMS(JCLDEF), the HISTW flag is set to Yes in sharedprefix.MICS.GENLIB(cccGENIN) and weekly history archive is active for this file in prefix.MICS.PARMS(DBMODEL). Then, each week thereafter, a data set with a name in the form shown above is created. This data set contains all of the WEEKS data previously gathered for each file plus data from the week just ended (subject to the cutoff limit specified in prefix.MICS.PARMS(DBMODEL)).

tapeprefix.MICS.HISTM.iiifff.GggggVnn

CA MICS supports the creation of archive history files. This support allows you to keep information in the MONTHS timespan available for far longer periods of time than can be managed in the online database.

If ARCHIVE HISTM YES is specified in prefix.MICS.PARMS(JCLDEF), the HISTM flag is set to Yes in sharedprefix.MICS.GENLIB(cccGENIN), and monthly history archive is active for this file in prefix.MICS.PARMS(DBMODEL); then each month thereafter, a data set with a name in the form shown above is created. This data set contains all of the MONTHS data previously gathered for each file plus that from the month just ended (subject to the cutoff limit specified in prefix.MICS.PARMS(DBMODEL)).

tapeprefix.MICS.ccc.IUXTAPE.GggggVnn

The CA MICS incremental update facility supports the option to offload incremental update data to tape between INCRccc and DAILY job executions. With this option, data processed since the beginning of the day is online only when needed. This tape data set stores DETAIL timespan data from incremental update processing.

tapeprefix.MICS.ccc.IUDTAPE.GggggVnn

The CA MICS incremental update facility supports the option to offload incremental update data to tape between INCRccc and DAILY job executions. With this option, data processed since the beginning of the day is online only when needed. This tape data set stores DAYS timespan data from incremental update processing.

DOCUMENTATION

The text for all CA MICS documentation is available in the following libraries. JCL and programs are supplied with CA MICS to format this text into printable guides or it can be browsed online using the Documentation Access option from the CA MICS Workstation Facility.

sharedprefix.MICS.DOC.TEXT

This data set contains all CA MICS documentation that is not generated or is not a data element definition. It is a variable blocked, LRECL=137 library containing single column text (columns 2-62) and figures (to column 133).

The members of this library have stylized names:

o the first one or two characters are the CA MICS manual identifier

o the remaining characters denote the chapter and section in the guide.

For example, the text you are reading is found in member L43A, where

o L is the (arbitrary) code for the CA MICS Planning, Installation, Operation, and Maintenance Guide

o 43A denotes chapter 4, section 3.10. Prefaces are denoted as X0, appendices as XA and so forth. Section numbers above 9 are coded as letters, where A=10, B=11, and so forth.

Members used in multiple guides or sections have names different from the structure noted above. For example, sample ISPF panel members use the same name as in the ISPPLIB library.

CA MICS manual identifiers are:

AH	- CA MICS Tandem Option Guide
AI	- CA MICS CA ASTEX Option Guide
BE	- CA MICS How to Use the PSP
BS	- CA MICS System Administrator Guide
C	- CA MICS Network Analyzer Option Guide
CK	- CA MICS Space Collector Option Guide
CL	- CA MICS IMS Log Data Extractor Option Guide
CM	- CA MICS Data Transfer Option for VM/CMS Guide
FB	- CA MICS Analyzer for TSO Guide
FC	- CA MICS Hardware and SCP Analyzer Guide
FD	- CA MICS Batch and Operations Analyzer Guide
FE	- CA MICS Analyzer Option for CICS Guide
FG	- CA MICS Analyzer Option for IMS Guide
FH	- CA MICS System Reliability Analyzer Option Guide
FI	- CA MICS Analyzer Option for VM/CMS Guide
FJ	- CA MICS Space Analyzer Option Guide
FM	- CA MICS Analyzer Option for MeasureWare Guide
FN	- CA MICS Analyzer Option for DB2 Guide
FO	- CA MICS Analyzer Option for MQSeries Guide
FQ	- CA MICS Analyzer Option for VSE/Power Guide
FT	- CA MICS Tape Analyzer Option Guide
FV	- CA MICS Analyzer Option for VAX/VMS Guide
FW	- CA MICS Web Analyzer Option Guide
H	- CA MICS Standard Reports Guide
I	- CA MICS Database Structure and Content Guide
JD	- CA MICS Document Access Guide
JR	- CA MICS MICF Reference Guide
JU	- CA MICS MICF User Guide
L	- CA MICS Planning, Installation, Operation, and Maintenance Guide
M	- CA MICS System Modification Guide
PA	- CA MICS Capacity Planner Option Guide
PD	- CA MICS Performance Manager Option Guide
PG	- CA MICS StorageMate Option Guide
PK	- CA MICS Accounting and Chargeback Option Concepts and Overview Guide
PL	- CA MICS Accounting and Chargeback Option User Guide
W	- CA MICS Analyzer Option for CA IDMS Guide

sharedprefix.MICS.DIC.TEXT

The text for the Data Dictionary resides in its own data set. The member names in this library are the names of the data elements themselves. Data elements are defined in the cccGENIN members of sharedprefix.MICS.GENLIB. For example, the element EXCCODE identified in BASGENIN is documented in this data set as member EXCCODE.

sharedprefix.MICS.INC.TEXT

This library holds documentation members that are copied by one or more documentation members during printing. Most members in this library are generated by the CA MICS Component Generator.

CONTROL INFORMATION

Every application system has a large amount of control information associated with it. Some of this information serves to guide the operation of the system's own programming (such as parameter cards), while a good deal of it exists to guide the operation of various operating system facilities (such as entries in catalogs, VTOCs, the JCL PROCLIBs, etc.) needed by the system.

CA MICS employs some very common control techniques (e.g., use of GDGs for related data sets) and some innovative ones (e.g., generation of tailored SAS code as a means of implementing CA MICS generation parameters). The following data sets are used to control the CA MICS system.

MVS System Catalog

CA MICS employs the MVS catalog mechanism for the standard purpose of locating its data sets. All CA MICS tape data sets (see above) are cataloged using the Generation Data Group facility.

SYS1.PROCLIB or Other Cataloged Procedure Data Set

CA MICS creates JCL procedures that are used by its operational jobs (e.g., DAILY) and to allow user inquiry into the Database. These PROCs are placed in the members PROCJOBx of prefix.MICS.CNTL. The PROCJOBx jobs (also in prefix.MICS.CNTL) move the JCL procedures to one of your cataloged procedure (PROCLIB) data sets (it need not actually be SYS1.PROCLIB).

sharedprefix.MICS.CLIST

A library containing CA MICS installation CLISTs and user-defined CLISTs for CA MICS ISPF applications startup.

prefix.MICS.CHECKPT.DATA

The Checkpoint data set is used by CA MICS operational jobs (e.g., the DAILY update job) to keep track of the operating status of the system. Information such as the last job step to complete and the low and high date/time stamps associated with input data processed by each installed component is kept here. This data set allows CA MICS to maintain database integrity by refusing to restart an operational job in any spot but the correct one.

prefix.MICS.CNTL

A JCL library containing generated batch jobs for a unit database.

sharedprefix.MICS.CNTL

A JCL library containing distributed and generated batch jobs that apply to the entire CA MICS complex.

sharedprefix.MICS.GENLIB

GENLIB contains control information used principally by the CA MICS Component Generator. The generator input statements for each product are located in this library. It also contains one member, DBTABLE, that is used by CA MICS for cross-checking parameters between unit databases.

sharedprefix.MICS.HOLD.PARMS

A library containing a standard set of parameter members that are copied by the COPYLIBS and COPYLIBU jobs when a creating a new unit database.

sharedprefix.MICS.LOCALMOD.CNTL

Store your local modifications (e.g., IEBUPDTE modification jobs) to CA MICS in this library. Refer to the System Modification Guide before modifying CA MICS programs.

prefix.MICS.MODEL

When the SYSTEMMODEL JCLGEN option cannot be used at your site, the DCB attributes of the prefix.MICS.MODEL data set are used to create new archive and backup file generation data groups. Since this data set never contains data, it frequently becomes the target of DASD cleanup programs and can be inadvertently scratched from time to time.

sharedprefix.MICS.PROTOLIB

A library of prototype JCL jobs and JCL procedures (PROCs). Users with extraordinary JCL tailoring needs and a good knowledge of the workings of the CA MICS JCLGEN process can modify members in this library to make changes in generated jobs.

Although sharedprefix libraries are shared by all CA MICS units maintained at your site, unit-specific changes are possible through use of Generation Control Language (GCL). Refer to the JCLGEN Parameters for Special Requirements section (2.3.3.3) of this guide.

prefix.MICS.PARMS

This library's members contain the numerous parameter specifications you must make to tailor CA MICS to your site's requirements. Some of the members contain parameter statements and others are exits you must code in SAS. Parameter statements do not become effective until a generation takes place (e.g., JCLGEN or ALLPGEN). Exits become effective immediately.

sharedprefix.MICS.PARMS

This library is similar to prefix.MICS.PARMS in function, but is used to perform JCL generation at the CA MICS complex level rather than the unit database level.

sharedprefix.MICS.SASFLS

A SAS-managed data set used to ship the contents of the TABLES, SCREENS, and MWFSAS.DATA data sets. During the installation of CA MICS, the SASFLS data set is used to create and populate these data sets. This data set is not used after the other data sets are created.

sharedprefix.MICS.SCREENS

A SAS-managed data set containing the screen formats for the SAS Full Screen Product (FSP). SAS/FSP is used to enter data for CA MICS files that allow manual data entry.

TEST LIBRARIES

sharedprefix.MICS.TEST.INCLLIB
sharedprefix.MICS.TEST.LOAD
sharedprefix.MICS.TEST.SOURCE

These libraries are provided to facilitate testing user modifications with a TEST unit database.

4.3.11 Abend Codes and Corrective Action

This section documents abend codes you may encounter with CA MICS processing. Refer to other sections in this guide for more information on restarting CA MICS operational jobs.

SUMMARY

User Abend Code	Termination Condition/Reason
100	Database marked non-updatable.
101	Database marked non-updatable; aging in progress when failure occurred.
110	Internal step restart failed; required files are missing or corrupted.
120	Internal step restart failed; saved work files missing or corrupted.
130	Stand-alone AUDIT, HISTW, and/or HISTM archive tape processing job not yet executed.
300	Input data file was empty.
310	The original system identifier in the input data was not defined in the SYSID parameter when the last run of ALLPGEN or BASPGEN occurred.
320	Zone was not defined in ZONE parameter when the last ALLPGEN or BASPGEN job was run.
400	Invalid checkpoint record encountered.
410	Limit of ORGSYSID/COMPONENT entries is exceeded in the Checkpoint File.
420	Invalid FORCE or SELECT range was specified.
425	Invalid SYSPARM value was specified.
430	INPUTRDR end-of-file routine was not entered after the last input record.
440	SAS version is not compatible with CA MICS

```
             version.

   User
Abend Code   Termination Condition/Reason
----------   ------------------------------------------------

     450     PREFIX in the Checkpoint File does not match the
             PREFIX used in the update procedure.

     460     Checkpoint File initialization failed or the
             checkpoint routine initial update processing
             failed.

     470     Exit did not return control properly.

     500     Execution date less than previous date.

     510     YEAR is not in the range of years defined to
             CA MICS.

     600     CA MICS JCL delimiter missing.

 800-899     Codes reserved for user.

     998     Runtime error with explanatory text.

     999     RSTATUS found non-updatable database
             or
             SAS System abend.

    1310     Insufficient memory.


  System
Abend Code   Termination Condition/Reason
----------   ------------------------------------------------

    S337     Less than expected number of records in
             Checkpoint File.

 SA13-10     Tape files cannot be written because prior file
             does not exist.

    S0C1     Batch ISPF dialog processing failure.

    S122     CA Common Services for z/OS and OS/390 not
             installed or operational.

ABEND CODES AND CORRECTIVE ACTION
```

```
   User
Abend Code   Termination Condition and Corrective Action
----------   -----------------------------------------------

     100     Database marked non-updatable.

             Probable causes:

             First, if a job is executed and its previous
             execution did not complete successfully, the job
             will not be allowed to continue.

             Second, if any other standard job's last run was
             unsuccessful, this job will not be allowed to
             continue.

             Third, if a RESTART specification was not
             removed after a previous run, the job will not
             be allowed to start at the specified restart
             step.

             Corrective Action:

             Identify the job that unsuccessfully terminated,
             correct or eliminate the problem causing the
             abend, and restart the job, ensuring its
             successful completion.


     101     Database marked non-updatable; aging in
             progress when failure occurred.

             Contact CA Technical Support for assistance.
             Have available the SAS log for the abended run
             and PROC DATASETS reports for the DETAIL, DAYS,
             WEEKS, MONTHS, and YEARS files.

     110     Internal step restart failed; required files
             are missing or corrupted.

             One or more SAS files required for internal step
             restart are missing or corrupted.

             Corrective Action:

             Specify SYSPARM=NORESTART on the JCL EXEC for
             the failing step and restart the job.
```

120 Internal step restart failed; saved work files missing or corrupted.

One or more intermediate work files on the cccXWORK and WORKnn data sets are missing or corrupted. These files are required to restart this processing phase.

Corrective Action:

Specify SYSPARM=NORESTART on the JCL EXEC for the failing step and restart the job.

130 Stand-alone AUDIT, HISTW, and/or HISTM archive tape processing job not yet executed.

You enabled stand-alone archive audit and/or history processing, but the archive processing from the previous WEEKLY or MONTHLY job has not yet executed.

Corrective Action:

Review the MICSLOG for the abending step to determine which archive tape processing job (AUDIT, HISTW, and/or HISTM) must be executed before you can resume today's scheduled processing. Determine why that job has not yet executed and complete that processing before restarting the abending operational job step.

300 Input data file was empty.

Probable causes:

A DAILY step's input file was read and the input file was empty or all input records were rejected for some reason. The MICSLOG output provides additional information as to the reason why the data was rejected.

The duplicate data checks used in input processing excluded all but 10 of the records due to Checkpoint File date range checking.

A _USRSEL (User Select) Exit caused all input records to be rejected.

Corrective Action:

Determine whether or not there should be input data. It is possible that no data is available, for example in the case of a CICS update step run for a day when CICS was not up. If a U300 abend occurs and no data is expected, specify SYSPARM=NODATA on the JCL EXEC statement for the failing step and restart the job.

If data is expected, determine what happened to the input data. When it is located, restart the DAILY job.

310 The original system identifier in the input data was not defined in the SYSID parameter when the last run of ALLPGEN or BASPGEN occurred.

Corrective Action:

Add the undefined system identifier to the SYSID and any other necessary parameter members, run BASPGEN, and restart the abended job.

or

Specify the UNDEFINEDSYSID EXCLUDE computing system control parameter at either the complex level or unit level, run the corresponding parameter generation job, and restart the abended job.

or

Temporarily specify a EXCLUDEORGSYSIDS UNDEFINEDSYSID statement in the prefix.MICS.PARMS(EXECDEF) member and restart the abended job.

320 Zone was not defined in ZONE parameter when the last CPLXGEN, ALLPGEN, or BASPGEN job was run.

Corrective Action:

If this unit is using the complex-level zone definition, add the undefined zone to the complex-level CPLXZONE PARMS member and run CPLXGEN. Otherwise, add the undefined zone to the unit ZONE PARMS member and run BASPGEN. Then restart the abended job.

400 Invalid checkpoint record encountered.

The Checkpoint File's first record contains an invalid step number or run completion date for one of the jobs tracked by this record, or an embedded blank line was found in the ORGSYSID/COMPONENT section of the database update time range records of the checkpoint data set.

Corrective Action:

Correct the checkpoint file and restart the abended job.

410 Limit of ORGSYSID/COMPONENT entries is exceeded in the Checkpoint File.

Either there are the maximum number of database update time range records in the Checkpoint File data set or the checkpoint data set was inadvertently line numbered. Checkpoint size is specified by the CKPTCNT parameter in prefix.MICS.PARMS(SITE).

Corrective Action:

If the checkpoint data set was line numbered and the sequence numbers are in columns 193-200 of each record, remove the sequence numbers and restart the job.

If any other reason caused this abend, contact CA Technical Support.

420 Invalid Force or Select range was specified.

Force or Select time ranges in other than SAS DATETIME 19.2 format.

Corrective Action:

Refer to the MICSLOG for more details. Correct the time range and restart the job.

425 Invalid SYSPARM value was specified.

A value specified for the SYSPARM symbolic parameter on the JCL EXEC statement is not recognized by CA MICS or is invalid for the job step being executed.

Corrective Action:

Refer to the MICSLOG for more details. Correct the SYSPARM value and restart the job. Note that a comma is not a valid delimiter within the SYSPARM value.

430 INPUTRDR end-of-file routine was not entered after the last input record.

This condition can be symptomatic of a bug in an exit routine or CA MICS parameter library member.

Corrective Action:

Contact CA Technical Support.

440 SAS version is not compatible with CA MICS version.

Have the SAS version, printed at the top of the SAS log (ddname FT11F001) at hand and contact CA Technical Support.

450 PREFIX in the Checkpoint File does not match the PREFIX used in the update procedure.

This abend is usually caused by changing CA MICS library data set names.

Corrective Action:

Contact CA Technical Support.

460 Checkpoint File initialization failed or the checkpoint routine initial update processing failed.

Checkpoint File initialization fails if the Checkpoint File has already been used (data range records exist in the Checkpoint File). Job CKPTINIT should not be forced to run. Contact CA Technical Support.

Checkpoint routine initial update processing fails if two or more database update time range records have identical ORGSYSID/COMPONENT identifiers. This can happen if you edit the Checkpoint File incorrectly. Contact CA MICS Product Support.

470 Exit did not return control properly

The Exit Check Routine issues this abend when the code of the user exit named in the abend message prevents program control from returning to the statement that follows the invocation of the exit.

Corrective Action:

Review the code in the user exit. See the Notes on Coding CA MICS Parameters section in this guide.

500 Execution date less than previous date

This abend is issued by the SCHEDULE job. If the SCHEDULE job detects that today's date of execution is less than any of the step completion dates maintained in the Checkpoint File, this abend is generated and prohibits the execution of SCHEDULE. Either today's date is wrong in MVS or it was incorrectly carried by MVS on a previous day.

Corrective Action:

If today's date is incorrect, reset the MVS date. If previous dates are incorrect, change them in the Checkpoint File and restart the SCHEDULE job.

510 YEAR is not in the range of years defined to CA MICS.

This abend only occurs when the 13MONTHYEAR option is specified in prefix.MICS.PARMS(SITE) or sharedprefix.MICS.PARMS(CPLXDEF). It occurs if a year is encountered which was not defined to CA MICS for the YEAR derivation routine.

Corrective Action:

Update the 13MONTHYEAR statement in sharedprefix.MICS.PARMS(CPLXDEF), run the CPLXGEN job, and then run the BASPGEN job.

or

Update the 13MONTHYEAR statement in prefix.MICS.PARMS(SITE) and run the BASPGEN job.

or

Update the sharedprefix.MICS.SOURCE members $DWMYnn and #DWMYnn for special fiscal calendars. See the prefix.MICS.PARMS(SITE) 13MONTHYEAR statement for member names.

600 CA MICS JCL delimiter missing

The standard CA MICS JCL procedures (DAILY, WEEKLY, MONTHLY, YEARLY, and BACKUP) have a special delimiter following their job cards, which is used by Operational Status and Tracking and the SCHEDULE batch job. The delimiter is:

```
//* START OF JCL
```

Corrective Action:

Check each of the five procedures in the prefix.MICS.CNTL library to ensure that the

//* START OF JCL statement follows the job card and related JES parameter (e.g., JOBPARM), then rerun the SCHEDULE job.

800-899 Codes reserved for user

998 Runtime error with explanatory text

A CA MICS runtime routine found an error condition which causes processing to stop. Further processing might damage the database if this abend occurs during an operational job. A description of the condition is printed in either the CA MICS log or SAS log (check the CA MICS log first).

Corrective Action:

Correct the problem according to the explanatory text and restart the job.

999 RSTATUS found non-updatable database
or
SAS System abend.

The Run Status Report (RSTATUS) has found the database in non-updatable condition. See the printed Run Status Report for details or use the online Operational Status and Tracking facility to research the problem.

or

SAS abended this job due to a processing problem that is identified in the SAS log.

Corrective Action:

For a non-updatable database, correct the problem described in the Run Status Report and restart the job.

For a SAS system abend, correct the problem reported in the SAS log and restart the job.

1310 Insufficient memory.

```
             The job that is being executed requires more
             memory than is available.

             Corrective Action:

             Increase the REGION size on the job step.  If
             the REGION is at its maximum value, break the
             job into multiple steps and include a REGION
             parameter on each step.

  System
Abend Code   Termination Condition and Corrective Action
----------   ------------------------------------------------

    S337     Less than expected number of records in
             Checkpoint File

             The Checkpoint File has been edited and one or
             more lines deleted.  The Checkpoint File must
             have exactly 1 more record than the value
             specified for the CKPTCNT parameter in
             prefix.MICS.PARMS(SITE).  For example, the
             default CKPTCNT specification is 100, meaning
             that the checkpoint data set should contain
             exactly 101 records.

             Corrective Action:

             Contact CA Technical Support.


 SA13-10     Tape files cannot be written because prior file
             does not exist

             Corrective Action:

             Determine why the previous file was not written,
             then call CA Technical Support for help in
             correcting the problem.


    S0C1     Batch ISPF dialog processing failure.

             MICF inquiry production reporting and other
             batch ISPF dialog jobs must generate a S0C1
             abend to cause the batch job step to end with a
             non-zero return code.  The S0C1 abend and the
             associated batch job step RC=12 indicate that a
             batch ISPF dialog process has failed.
```

Corrective Action:

Review the CA MICS log and ISPLOG outputs to determine the reason for batch ISPF dialog termination. Be sure to have the MICSLOG, ISPLOG, and other batch job outputs available if you need to call CA Technical Support for help in correcting the problem.

S122 CA Common Services for z/OS not installed or operational.

Either the CA Common Services for z/OS LMP service is not installed and operational, or the console operator cancelled the job requesting a dump.

CA LMP is one of the CA Common Services for z/OS. CA LMP provides a standardized and automated approach to the tracking of licensed software. It must be installed and operational before you can install or run CA MICS.

For more information on how LMP works with CA MICS, see Section 3.2.2 in the PIOM. For a full description of the procedure for defining the CA LMP execution key to the CAIRIM parameters, see the section on CA LMP in the CA Common Services for z/OS and OS/390 Administrator Guide.

4.3.12 Special Processing Considerations

The following sections list special considerations for CA MICS processing.

1 - Data Unavailable
2 - Minimal Input Data
3 - Impact of Seasonal Time Changes on CA MICS
4 - CA MICS Year-End Processing Considerations
5 - Internal Step Restart Considerations
6 - Incremental Update Considerations

4.3.12.1 Data Unavailable

CA MICS normally requires that data from all data sources be processed at the same time and will shut down with a user 300 ABEND if data is not available when expected.

To prevent an ABEND when you know that data from a particular data source will not be available, add the SYSPARM=NODATA parameter to the EXEC card of the step that normally receives the data. Doing so means that the system accepts the no-data condition rather than abending and facilitates keeping CA MICS files synchronized to the same processing intervals.

Restrict use of this parameter to those times when it is absolutely essential. If left for an indefinite period, processing problems could go undetected.

If you are using the CA MICS incremental update feature to facilitate parallel processing of the data integration product daily database update within a single unit database, then you may need to specify SYSPARM=NODATA for the DAILY job steps. In this mode of operation, the INCRccc jobs process all of the day's input data and the DAILY job steps operate with empty input files.

When SYSPARM=NODATA is used, empty 01 DETAIL and DAYS cycles are created for all files whose update step is using the parameter. (When a user 300 ABEND occurs, where no new cycles are created and existing cycles are not aged.)

Note: NODATA does NOT exclude data; it merely prevents a user 300 ABEND.

If NODATA is specified and there is data, the data will be read and added to the Database. You cannot use SYSPARM=NODATA to cause the format routine to discard valid data. If the format routine sees valid data, it will process it.

Only code SYSPARM=NODATA in the DAYSMF step if data for ALL products is unavailable.

To add SYSPARM=NODATA in DAILY job steps,

o Edit prefix.MICS.CNTL(DAILY)

o Enter

```
FIND DAYnnn
```

in the Command area (where nnn is the step number to be changed).

o Replace

```
SYSPARM=
```

with

```
SYSPARM=NODATA
```

o END and save the updated DAILY job JCL.

o When you expect data, edit prefix.MICS.CNTL(DAILY) and remove SYSPARM=NODATA (back to SYSPARM=).

To add SYSPARM=NODATA in INCRccc jobs (incremental update),

o Edit prefix.MICS.CNTL(INCRccc)

o Enter

```
FIND INCRnnn
```

in the Command area (where nnn is the step number to be changed).

o Replace

```
SYSPARM=
```

with

```
SYSPARM=NODATA
```

o END and save the updated INCRccc job JCL.

o When you expect data, edit prefix.MICS.CNTL(INCRccc) and remove SYSPARM=NODATA (back to SYSPARM=).

Remember that your change will be lost if you regenerate the DAILY or INCRccc jobs. For example, if you run the JCLGEND job, your SYSPARM=NODATA specification will be cleared.

4.3.12.2 Minimal Input Data

CA MICS assumes that a problem exists if less than 10 input records are available to process. CA MICS treats this condition as the equivalent of a no-data condition. SPECIAL unit databases are not subject to the 10 input record validation.

Follow the instructions in the Data Unavailable section (4.3.12.1) in this guide to handle situations where you expect less than 10 input records.

4.3.12.3 Impact of Seasonal Time Changes on CA MICS

All data sources that are processed by CA MICS are sensitive to a time of day adjustment that usually occurs in the spring and fall of the year. The United States calls this "Daylight Saving" versus "Standard" time. Other countries have different names, such as "Summer Time" in parts of Europe.

A time adjustment occurs twice a year: in the spring, clocks are set ahead one hour, and in the fall they are set back one hour. That is, one early morning in the spring, 2:00 a.m. becomes 3:00 a.m., and hour 2 of that day is totally missing from any system measurement. One early morning in the fall, 2:00 a.m. becomes 1:00 a.m., and hour 1 of that day occurs two times. The change of time is usually accomplished in the operating system by an IPL, or by dynamically changing the system clock or SYSPLEX timer.

This can be graphically illustrated:

In the spring:

```
    time changes here----------+
                               |
                               V
    |--------|--------|--------|--------|------->
     hour 23 hour 0 hour 1 hour 3 hour 4
```

In the fall:

```
    time changes here----------+
                               |
                               V
    |--------|--------|--------|--------|------->
     hour 23 hour 0 hour 1 hour 1 hour 2
```

The way your installation handles these time changes can have an effect on the input data supplied to CA MICS. The problems associated with these time changes and the consequences of actions that your installation might take are described below.

NOTE: Neither the spring nor fall time change will have an effect on your installation's input data if no jobs are executing during the time change (for example, in an installation that has no midnight shift). The system is simply IPLed on Monday morning with the correct time.

SPRING

The time change in the spring has a much smaller impact on the CA MICS Database than the one that takes place in the fall. The degree of impact is also dependent on the way your installation handles the time change.

There are several points to keep in mind if the system is in operation when the clock is advanced one hour. If batch jobs are running, or if online sessions are active, advancing the clock will result in elongated run times and response times. If priority charging is in effect, the service objectives of the installation might not be met. These events can affect the chargeback system. CA MICS files, plots, graphs, and reports will contain very little data for the 02:00 to 02:59 hour (that is, HOUR = 2) because very few ENDTS time stamps will be recorded for that period. This may affect your exception reporting.

Elongated job run times, response times, and their resultant effects upon the chargeback system can be avoided by using one of the following methods.

o Schedule preventive maintenance during the time change period. This method requires the least amount of manual intervention on the part of the installation.

o Shut down or quiesce the system at 01:59 standard time, and then IPL the system at 03:00 daylight saving time (elapsed down time: one minute). If the system is shut down, one hour of input data will be missing from CA MICS. If it remains active, then batch jobs and online sessions will have their elapsed times extended by one hour. If this method is used, the operations staff must be careful to IPL the system at 03:00 daylight saving time. Otherwise data may be produced for a nonexistent hour.

The above methods result in very little data for the 02:00 to 02:59 hour and may inconvenience your operations staff and active online users; but installations that use SMF data for accounting and chargeback purposes often choose one of these two methods in order to preserve the validity of the chargeback system.

Other alternatives include:

o Issue the SETCLOCK command from the OS operator console to temporarily increase the Greenwich mean time (GMT) offset by one hour; but be careful. Some CA MICS products process data in which event time stamps are offset by the GMT offset. These products are the CICS and SRL products, but it is expected that more products will eventually be affected. CA MICS only supports one definition of a GMT offset per system or subsystem processed. This creates a problem IF the GMT offset changes when the local time of day changes. CICS and LOGREC data collected with the old GMT offset must be processed by CA MICS, then the GMT offset in the CICS or SRL OPTION statement changed, then an ALLPGEN run--all before the new GMT offset data is processed by CA MICS.

o Recognize that a problem may exist with the data and be prepared to ignore the problem.

FALL

There are several difficulties that arise when the system is in operation when the clock is turned back one hour.

If the system is allowed to run in the "second" hour 1 in the fall, some unusual time stamp events occur. For example, batch jobs may complete or print before they start or online sessions may end before they begin. CA MICS treats such obviously impossible input queue times as bad data and sets the incalculable interval to a value of zero or a missing value. Jobs, transactions, and system measurement intervals that span these time changes may have incalculable values for many data elements. These data elements may include (but are not limited to) queue times, elapsed times, print times, reader times, and DURATION. CPU times are rarely affected.

The CA MICS DETAIL file will contain twice as much data for the 01:00 to 01:59 hour. Service objectives will be invalid and certain times will be missing because they were negative. In the summary files (such as DAYS and WEEKS), the hourly summaries will contain twice the volume (for example, DURATION = 2 hours). In addition, the zone summaries in the MONTHS and YEARS files will contain an extra hour. Plots and graphs will report inflated values for the 01:00 to 01:59 hour and lose their significance.

As mentioned earlier, the CICS and SRL products process data in which event time stamps are offset by the GMT offset. In addition, it is expected that other products will eventually be affected. CA MICS supports only one definition of a GMT offset per system or subsystem processed. If the local time of day changes at the same time as the GMT offset, this causes a problem. Before CA MICS processes the new GMT offset data, you must: 1) have CA MICS process the CICS and LOGREC data collected with the old GMT offset; 2) change the GMT offset in the CICS or SRL OPTION statement; and 3) run an ALLPGEN.

Some solutions include:

- Ignore the data for that time period. Reports would have to be ignored or deleted. All users would have to be warned that their reports were erroneous or notified that the daily reports had been eliminated.

- Shut the system down at 01:59 daylight saving time, turn the clock back and wait one hour; then IPL the system at 02:00 standard time. This method is rather drastic as it loses an hour of processing time, but it may be justified in those shops that cannot afford to have invalid data for that period. Installations that use SMF data for accounting and chargeback purposes often choose this method in order to preserve the validity of the chargeback system.

  When using this method unless the operations staff is EXTREMELY careful, the change of time can occur at an odd point. For example, if the system is allowed to run until 02:15 and is then reset to 15 minutes after the new hour, this can produce data with unusual overlaps.

- Apply GMT offset specifications to CA MICS coupled with monitor stoppage at the time change point. This allows a clean break at the time change point. If the SMF logs are dumped at the time change, CA MICS can process data from before the time change in one run and can process data from after the time change in the next run. This will cause data to be dropped, because data up to 02:00 would have been processed by the first run. Data for the period 01:00 to 02:00 would also occur in the second run, and be discarded due to the overlap of hour 1.

- Schedule hardware preventive maintenance over the time change period. This is usually practical because the time change is normally on a Sunday morning.

4.3.12.4 CA MICS Year-End Processing Considerations

This section highlights the CA MICS complex-level and unit-level specifications that should be checked prior to the start of the new calendar year. It also discusses how the execution of CA MICS operational jobs at the end of the year can affect the content of CA MICS database files.

CA MICS COMPLEX-LEVEL CONSIDERATIONS

1. Validate the holiday specifications in the CPLXZONE member of the sharedprefix.MICS.PARMS library. If any changes are made, execute sharedprefix.MICS.CNTL(CPLXGEN) to make them effective. See Section 2.3.1.10 for more information.

2. Review any user calculations in the $DWMYxxx and #DWMYxxx members of sharedprefix.MICS.SOURCE, where xxx is 12, 13, WK1, or a user-defined value. These members contain the macros that determine the values of the DAY, WEEK, MONTH, and YEAR elements. Hard-coded dates may need to be changed, but be careful; if your site has the CA MICS Capacity Planner management application installed, do not delete the start date of any year that could be used in long term capacity trend or forecast studies. See Section 4.7 in the System Modification Guide for information about coding these members.

3. If the YEARS TIMESPAN ACTIVE statement is coded in sharedprefix.MICS.PARMS(CPLXDEF) or if the YEARS TIMESPAN parameter is NOT specified in this member, then the YEARS timespan will be active by default in each unit database. Verify that enough space is available in each unit database YEARS timespan for additional data.

   If YEARS TIMESPAN INACTIVE statement is coded in sharedprefix.MICS.PARMS(CPLXDEF), then the YEARS timespan will be inactive by default in each unit database.

   Sites that operate on a non-standard (that is, greater than twelve months) calendar should also ensure that the sharedprefix.MICS.PARMS(CPLXDEF) 13MONTHYEAR parameter contains the start date of both the new year and the previous year. Leave enough specifications in the member to support the Capacity Planner management application if it is installed. See Section 2.3.1.8 for more information about the CPLXDEF member.

4. Review MICF inquiries, production inquiry jobstreams and user-written batch reports for hard coded selections that are based upon date, timespan, or file cycle values and adjust accordingly.

5. If the CA MICS Accounting and Chargeback management application is installed:

   o If your site utilizes a special accounting calendar that is different from the CA MICS calendar, review Sections 1.3.6 and 4.6 in the Accounting and Chargeback User Guide and ensure that monthly cutoff dates for the upcoming year have been specified.

   o Determine if any of the following files need to be updated for the coming year:

   ```
   ACTBGT00 - Budget File
   ACTCCA01 - Cost Center Attribute File
   ACTRCT00 - Recurring Credit File
   ACTRDT00 - Recurring Debit File
   ACTSEQ01 - Invoice Sequence Number File
   ```

   See Chapter 7 and Appendix A in the Accounting and Chargeback User Guide for more information about these files.

   o Examine complex-level utility jobs, prorating specifications and external file logic for any required changes. See Sections 4.15 and 1.3.8, and Chapter 4 in the CA MICS Accounting and Chargeback User Guide, respectively, for information about changing these specifications.

CA MICS UNIT-LEVEL CONSIDERATIONS

Review the following items for each unit in the complex.

1. For units that override the complex-level ZONE definition, validate the holiday specifications in the ZONE member of prefix.MICS.PARMS. If any changes are made, execute prefix.MICS.CNTL(BASPGEN) to make them effective. See Section 2.3.2.3 for further information.

2. We recommend that you do a year-end review of all other prefix.MICS.PARMS members to ensure that they contain adequate information for the new year. Review step 8 in Section 3.8.2.2 for a list of standard parameter members

to check.

3. If the years timespan is active, ensure that enough space is available in the YEARS timespan for additional data. The years timespan will be active if YEARS TIMESPAN ACTIVE is coded in prefix.MICS.PARMS(SITE), or if the YEARS TIMESPAN parameter is NOT specified in this member and defaults to active due to sharedprefix.MICS.PARMS(CPLXDEF) specifications,

   If the YEARS timespan is inactive, do not execute the YEARLY job.

   If the 13MONTHYEAR parameter is specified at the unit-level in prefix.MICS.PARMS(SITE), ensure that this parameter contains the start date of both the new year and the previous year. Leave enough specifications in the parameter to support the Capacity Planner management application if it is installed. See Section 2.3.2.4 for more information about the SITE member.

4. If the CA MICS Accounting and Chargeback management application is installed in this unit:

   - Determine if the rate specifications for any journal file or external file need to be changed for the coming year. The effective date facility allows you to enter these rates in advance. See Sections 1.3.3 in the CA MICS Accounting and Chargeback User Guide for more information.

   - Determine if any changes are required to unit-level prorating specifications. See Section 1.3.8 in the CA MICS Accounting and Chargeback User Guide for more information.

   - Review Section 5.3 in the CA MICS Accounting and Chargeback User Guide for yearly processing steps.

SCHEDULING CA MICS UPDATE JOBS FOR YEAR-END PROCESSING

The end of the calendar year can necessitate a change in the normal execution of CA MICS operational jobs. The CA MICS DAILY job should be submitted in each unit as usual. The submission of the WEEKLY job, however, can be affected by the day of the week on which the new year begins or by the way weekend job scheduling is handled at your site. User reports and inquiries that select data based upon the WEEK variable or the cycle number can be affected as a consequence.

It may be necessary for your site to submit an additional WEEKLY job at the beginning of the new year to ensure that data for the last week of the previous year and data for the first week of the new year reside in different cycles.

Two CA MICS facilities can be used to predict the requirement for an additional WEEKLY job.

Use the SCHEDULE command in the MWF Operational Status and Tracking facility or manually submit the SCHEDULE job in each unit to find out how the end of the calendar year will affect CA MICS processing. Section 4.3 discusses each of these facilities in detail.

Issue the SCHEDULE command in the Operational Status and Tracking facility for each unit to produce a report of jobs that will be submitted over the next seven day period. (Note that this command will consolidate all CA MICS jobs for the day into one JCL jobstream and write the data to prefix.MICS.RESTART.CNTL.)

Submit the prefix.MICS.CNTL(SCHEDULE) job in "simulation" or "test" mode in order to print a list of CA MICS update jobs that will execute over a specified time period. You can submit this job even if your site does not use the CA MICS scheduling facilities. To execute the SCHEDULE job in test mode, modify the SYSPARM parameter on the //SCHEDULE EXEC card to read as follows:

```
    SYSPARM='BKFQ=DAILY TEST(TODAY=ddmmmyy TESTEND=ddmmmyy)'
```

where TODAY equals the starting date to be analyzed by the SCHEDULE job in day-month-year format, and TESTEND equals the ending date.

When TEST is specified, the SCHEDULE job will receive a return code. No jobs will be scheduled or submitted. You will find the simulated schedule report in the SASLOG DD.

Look at a sample report produced at a site that has the following operational configuration:

- o The CA MICS year is defined as a normal calendar year.

- o The CA MICS week begins on Sunday.

- o The BACKUP job is executed on a daily basis.

o The CA MICS SCHEDULE facility is used to submit jobs.

o The YEARS timespan is active.

o SCHEDULE WEEKEND NO is specified in the EXECDEF member of prefix.MICS.PARMS to prevent submission of the WEEKLY, MONTHLY, or YEARLY jobs on a weekend. This parameter has a direct impact on the amount of data transferred to the weekly file cycles at year-end.

The last column of the sample report lists the jobs that will be submitted on a given date, where DWMYB is equal to the DAILY, WEEKLY, MONTHLY, YEARLY, or BACKUP job, respectively. The YEARLY job will not be listed if the YEARS timespan is specified as INACTIVE in prefix.MICS.PARMS(SITE).

```
SUN 27DEC06 DAY=27 MONTH=12 YEAR=06 WEEK=53 DWMYB=D B
MON 28DEC06 DAY=28 MONTH=12 YEAR=06 WEEK=53 DWMYB=DW B
TUE 29DEC06 DAY=29 MONTH=12 YEAR=06 WEEK=53 DWMYB=D B
WED 30DEC06 DAY=30 MONTH=12 YEAR=06 WEEK=53 DWMYB=D B
THU 31DEC06 DAY=31 MONTH=12 YEAR=06 WEEK=53 DWMYB=D B
FRI 01JAN07 DAY=01 MONTH=01 YEAR=07 WEEK=01 DWMYB=D B
SAT 02JAN07 DAY=02 MONTH=01 YEAR=07 WEEK=01 DWMYB=D-  B
SUN 03JAN07 DAY=03 MONTH=01 YEAR=07 WEEK=02 DWMYB=D-  B
MON 04JAN07 DAY=04 MONTH=01 YEAR=07 WEEK=02 DWMYB=DW- B
TUE 05JAN07 DAY=05 MONTH=01 YEAR=07 WEEK=02 DWMYB=D M B
WED 06JAN07 DAY=06 MONTH=01 YEAR=07 WEEK=02 DWMYB=D YB
THU 07JAN07 DAY=07 MONTH=01 YEAR=07 WEEK=02 DWMYB=D B
FRI 08JAN07 DAY=08 MONTH=01 YEAR=07 WEEK=02 DWMYB=D B
SAT 09JAN07 DAY=09 MONTH=01 YEAR=07 WEEK=02 DWMYB=D B
SUN 10JAN07 DAY=10 MONTH=01 YEAR=07 WEEK=03 DWMYB=D B
```

The CA MICS scheduling facility has a built-in submission delay of one day for the WEEKLY job, three days for the MONTHLY job, and five days for the YEARLY job to allow time for adjustments to the database or recovery from abend situations. By default the CA MICS scheduling facility will not submit a WEEKLY and MONTHLY job or a MONTHLY and YEARLY job on the same day. A dash in the DWMYB column represents a CA MICS job that has a scheduling conflict. Note that the WEEKLY job is in conflict and will not run until Monday, January 4th. How does this affect the content of CA MICS database files?

CA MICS translates the system date at the time the WEEKLY job is executed to the current WEEK value. CA MICS cycles data from each database file's week-to-date 00 cycle to the 01 cycle based upon the value in the WEEK variable of each observation. All observations with a WEEK value that is less than the current week's value moves to the 01 cycle; observations with a WEEK value that is equal to the current week's value remains in the 00 cycle. How does this affect CA MICS reporting?

In the sample report above, a WEEKLY job will be submitted on Monday, December 28th, as usual. Note that the MONTH, YEAR and WEEK values change on Friday, January 1st, because of the change in the calendar year. (See how WEEK is determined in the System Modification Guide Section 4.7.2.4.) Another WEEKLY job would normally be scheduled for Saturday, January 2nd, but no weekend job submission is allowed in this unit. Therefore, the next WEEKLY job is scheduled for Monday, January 4th, which is CA MICS WEEK 2.

As a result of this delay in the normal execution of the WEEKLY job, the 01 cycle of all weekly files in this unit will contain data from WEEK 53 and WEEK 01. The source code for reports, inquiries, and graphs that use the WEEKS timespan at this site would have to filter out data for WEEK 01. The site would also have to change source code to include an additional cycle of weekly data in order to construct weekly and monthly reports for January's data. Based on this simulation report, the system administrator should schedule an additional WEEKLY job to execute on Saturday, January 2nd to avoid making these code changes.

The MONTHLY and YEARLY jobs at this site are scheduled for January 5th and 6th in the simulation report. The system administrator has the option of submitting the MONTHLY job on Sunday, or allowing it to execute as scheduled on Monday. The monthly files will be cycled based on the value in the MONTH variable and the yearly files will be cycled based on the value in the YEAR variable. No adjustment to report code is necessary.

CA MICS sites that use either one of the CA MICS scheduling facilities and have SCHEDULE WEEKEND YES specified in the prefix.MICS.PARMS(EXECDEF) each unit do not have to make these scheduling adjustments; a WEEKLY job will be scheduled automatically for Saturday, January 2nd. Thus, CA MICS will schedule two WEEKLY jobs in a seven-day period; the cycles created by each WEEKLY job will have less than seven day's worth of data in them.

Sites that do not use the CA MICS scheduling or Operational Status and Tracking facilities can create a schedule report that is based upon their own year-end date and submit their year-end processing jobs in a similar manner or adjust automated scheduling packages to adhere to a slightly modified year-end process.

If your site normally specifies SCHEDULE WEEKEND NO in prefix.MICS.PARMS(EXECDEF) or does not use the CA MICS Scheduling facility, then you may want to manually schedule an additional WEEKLY job at the end of the year so that data from the last week of the year will be separated from data in the first week of the new year.

4.3.12.5 Internal Step Restart Considerations

Internal step restart can significantly reduce time and resource usage to recover from daily update processing failures. CA MICS uses this checkpoint/restart technique:

- When activated for an enabled product, the database update job step "checkpoints" (or saves) intermediate results (work file contents) and the operational environmental at the end of each processing phase.

- Then, if required, the database update step can resume execution at the beginning of the processing phase in which the failure occurred.

- 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.

Processing Phases

Internal step restart implementation is unique to each CA MICS Data Integration Application. The number of processing phases and location of restart points depends on product specific processing requirements, data architecture, and data volume distribution.

The general processing phases are as follows:

- Read raw input data, convert to SAS format, and output to intermediate work files. This first processing phase is common across products. The first checkpoint is generally taken immediately after reading all input data.

- Sort, analyze, summarize, and enhance intermediate work file contents to create a new DETAIL cycle. This processing is comprised of one, two, or more processing phases.

- Summarize DETAIL data to create new DAYS cycles and to update current week-to-date and month-to-date cycles. This phase, DYSUM, is common to all products.

- Cutover new database cycles to production and "age" existing cycles. This phase, DYAGE, is common to all products.

See the product guides for more information on each product's internal step restart implementation.

Overhead

Enabling internal step restart adds some overhead to a product's database update job step, including 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, for some products and unit database environments, costs outweigh potential savings. For example, internal step restart may be of little value for a CICS unit supporting test regions, while proving very valuable in unit database updates for your production CICS regions. In general you will want to enable internal step restart for those products in a unit with large input data volume and long database update job step elapsed time. See the individual product guides for product specific instructions and considerations.

Cataloged Work Files

Each database update job step allocates one or more OS data sets to hold intermediate SAS work files, including the SAS WORK data set and the multiple work file data sets (WORK01 - WORKnn). These are normally allocated on system "scratch" space with a temporary, system assigned data set name. The work data sets are deleted when the job step ends, either successfully or with an abend.

When internal step restart is enabled, the work data sets are allocated and cataloged with permanent data set names (for example, prefix.MICS.cccXWORK, where ccc is the product ID) so they will be kept for use in restart should the database update job step abend. The work data sets are deleted when the database update job step completes successfully.

Some site DASD allocation standards do not allow "permanent" data sets on DASD volumes used for temporary work space. If this is the case at your site, you may need to direct the internal step restart data sets to a generic unit or storage class that allows cataloged data sets. See the product guides for required parameter syntax.

Data set allocation parameters for the WORKnn and WORK data sets are specified by product in prefix.MICS.PARMS(cccOPS) and permanent changes to data set allocation parameters (e.g., to increase the space allocation for the WORK data set) require both changing the cccOPS parameter and executing the corresponding cccPGEN job. However, in restart situations you can temporarily override data set allocation parameters for one or more dynamically allocated data sets by using the //PARMOVRD facility. See Section 2.3.6 in this guide for more information.

Dynamic Allocation

When internal step restart is active, dynamic allocation is employed for the work data sets. If your site restricts dynamic allocation of large, cataloged data sets, you may need to direct work data set allocation to a generic unit or storage class where dynamic allocation is allowed.

If your site automatically assigns batch job classes or schedules work based on the amount of DASD space a job requires, you may need to direct the CA MICS DAILY and/or INCRccc jobs to a specific job class and processing system. Once you enable internal step restart, your scheduling facilities will no longer be able to determine DASD space requirements by examining the DAILY or INCRccc job JCL.

4.3.12.6 Incremental Update Considerations

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.

- Standard CA MICS database update processing involves the following steps:

  1. Reading and processing raw input data to generate DETAIL and DAYS level CA MICS database files.

  2. Summarization of DETAIL/DAYS level data to update week-to-date and month-to-date database files.

- When you activate incremental update, the following happens:

  - 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.

  - 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.

- 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.

- Incremental update is activated and operates independently by product. The incremental update job for one product, INCRccc (where ccc is the product ID) can execute concurrently with the incremental update job for another product in the same unit database.

- 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            *
*            site, 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.              *
*                                                           *
*************************************************************
```

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 OS/390 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.

After activating incremental update, you will use four incremental update facility jobs found in prefix.MICS.CNTL, where 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.

o SPLITSMF

  You optionally execute this job to split the input data into separate data sets for one or more INCRccc jobs. The SPLITSMF job is a standalone version of the DAILY job's DAYSMF step designed specifically for INCRccc processing. Each execution of the SPLITSMF job is normally followed by one or more INCRccc jobs.

  The SPLITSMF job is activated by specifying both the INCRUPDATE YES and INCRSPLIT USE parameters in the cccOPS parameter member for one or more products. You would then integrate this job into your database update procedures for execution one or more times per day to preprocess portions of the total day's measurement data. You would schedule INCRccc jobs to execute immediately after SPLITSMF completes.

See the product guides for more information on each product's incremental update implementation.

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 product's 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 run-time of the product's DAYnnn job step.

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.

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 in the product guides). 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.

Operational Complexity

Incremental update expands your measurement data management and job scheduling issues. You must ensure that each incremental update (or optionally the associated SPLITSMF job) and the DAILY job process 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.

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.

o 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.

o 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.

Dynamic Allocation

Incremental update facilities optionally employ dynamic allocation for the incremental update DETAIL and DAYS timespan data sets. If your site restricts dynamic allocation of large, cataloged data sets, you may need to direct incremental update data set allocation to a generic unit or storage class where dynamic allocation is allowed.

If your site automatically assigns batch job classes and/or schedules work based on the amount of DASD space a job requires, you may need to direct the CA MICS INCRccc and/or DAILY jobs to a specific job class and/or processing system.

If you choose the incremental update dynamic allocation options, your scheduling facilities will no longer be able to determine DASD space requirements by examining the INCRccc or DAILY job JCL.

Data set allocation parameters for IUDETAIL and IUDAYS are specified by product in prefix.MICS.PARMS(cccOPS) and permanent changes to data set allocation parameters (e.g., to increase the space allocation for the IUDETAIL data set) require both changing the cccOPS parameter and executing the corresponding cccPGEN job. However, in restart situations, you can temporarily override data set allocation parameters for one or more dynamically allocated data sets by using the //PARMOVRD facility. See Section 2.3.6 in this guide for more information.

4.3.13 CA MICS Audit and History Archive Tapes

The CA MICS archive facility allows you to keep the direct access storage device (DASD) space requirements of CA MICS to a minimum by off-loading database files to tape. It also enables you to retain information for future analysis.

There are two types of CA MICS archive tapes, audit and history. The audit archive tape contains critical DETAIL-level files that can be used for accounting, performance, or security analysis. The history archive tapes contain data from user-designated files at the WEEKS or MONTHS timespan that can be used for long-term analysis.

The following sections describe the processes of establishing and creating the audit and history archive tapes and how to retrieve information from them.

1 - The CA MICS Audit Archive Tape
2 - The CA MICS History Archive Tapes
3 - Using CA MICS Archive Tapes With MICF

4.3.13.1 The CA MICS Audit Archive Tape

The audit archive tape contains data from selected database unit files that contain a user identification field in the file key. CA MICS automatically selects these files during the WEEKLY job or the stand-alone AUDIT job based on your JCLDEF specifications.

File selection is based upon the components that you have installed in each database unit and the status of the file (i.e., active or inactive). Section 6.2.3, Tailoring Archive Files, in the System Modification Guide explains how to add and delete audit archive files.

ESTABLISHING AUDIT ARCHIVING

Audit archiving is controlled by several CA MICS statements and parameters. The COMPJOB and FILE statements in sharedprefix.MICS.GENLIB(cccGENIN), where "ccc" is the three-character product identifier, determine if audit archives can be created. The COMPJOB statement has an indicator that shows whether the product is eligible to create audit archives. And the FILE statement determines whether a product file is eligible for audit archiving. In addition, the FILE statement in prefix.MICS.PARMS(DBMODEL) determines whether a file is created. The ADM information area files are not supported for audit archiving. You must specify the following parameters in prefix.MICS.PARMS(JCLDEF) to complete the establishment of audit archiving:

- The ARCHIVE AUDIT YES parameter specifies that audit archiving is to take place

  - If you specify ARCHIVE AUDIT YES JOB, audit archiving occurs in the stand-alone AUDIT operational job.

    - If ARCHIVE AUDIT YES JOB is specified and the AUDIT job is run more than once a week, the following must be added to prefix.MICS.PARMS(EXECDEF)

      ```
      USERDEF AUDITCWK YES
      ```

      This parameter overrides the default audit archive processing so that data for the current week is retained and copied to the new audit archive tape cycle.

  - If you specify ARCHIVE AUDIT YES JOB AUTOSUBMIT, the

WEEK300 step of the WEEKLY job automatically submits the AUDIT operational job to perform audit archiving.

- If you specify ARCHIVE AUDIT YES STEP or take the default (i.e., ARCHIVE AUDIT YES), audit archiving takes place in the WEEK300 step of the WEEKLY job.

o The TAPEPREFIX parameter specifies the high-level qualifier of the data set names on the audit archive tape.

o The AUDITGDG parameter specifies the maximum number of generation data group (GDG) entries that CA MICS should establish for the audit tape. Since audit archiving is done on a weekly basis, this number is equal to the number of weeks that each audit tape is retained.

Section 6.2.3 of the CA MICS System Modification Guide explains how to add and delete audit archive files.

HOW THE AUDIT ARCHIVE TAPE IS CREATED

During the stand-alone AUDIT job or the WEEK300 step of the WEEKLY job, CA MICS obtains your setting for AUDITCYC which is specified in prefix.MICS.PARMS and defines the number of CA MICS file cycles you want written to the audit archive file. If not specified, CA MICS searches the database unit for the last ten (10) DETAIL cycles of each of the files in the audit archive selection list. If the DETAIL time-span has no active cycles, then cycles from the DAYS level are used. If both time-spans are active, the DETAIL level is used. The FILE statement in prefix.MICS.PARMS(DBMODEL) can restrict audit archive to either the DETAIL or DAYS time-span. If fewer than the number of requested (or the default value of 10) cycles exist, all available DETAIL or DAYS cycles are used. The observations from the available cycles, excluding the current week's observations, are then concatenated into one SAS file on tape. Usually, all of the SAS files can be accommodated on one physical tape reel. Note that the files are not summarized during this process and that duplicate observations may be present in the different generation files (see AUDIT ARCHIVE RETRIEVAL CONSIDERATIONS below).

Each file on the audit tape is entered in the OS catalog as:

```
    tapeprefix.MICS.AUDIT.iiifff.GggggV00
```

where "tapeprefix" is the high-level qualifier specified in the TAPEPREFIX parameter of prefix.MICS.PARMS(JCLDEF), "iiifff" is the information area and filename, and "gggg" is the absolute number of a generation data group (GDG).

RETRIEVING DATA FROM THE AUDIT ARCHIVE TAPE

The audit tapes may be accessed through a MICF inquiry or a batch job. Refer to Section 4.3.13.3, Using CA MICS Archive Tapes With MICF, for more details about accessing the archive tapes in MICF inquiries.

The JCL specifications that are used to access the files on the audit archive tape are different from those used to access the files on the database unit. Database unit files are referenced as:

```
//S01 EXEC MICSSHRx
//SYSIN DD *
DATA;
  SET &iiit..BATJOBnn ;
```

where x is the database unit ID, "&iiit" is a CA MICS/SAS macro variable representing the information area (iii), time-span (t) of the file to be accessed, and nn is cycle. Coding the macro variable as compared to a time-span ddname eliminates the need to know if a particular file has been defined to DETAIL, DETAIL2 or another split database file. Refer to Section 2.3.3.3.2.4 for information about the CA MICS Database Split Table.

Files on the audit archive tape are referenced as:

```
//S01 EXEC MICSNDBx
//ddname DD DSN=tapeprefix.MICS.AUDIT.iiifff<gdgnumber>,
// DISP=OLD
//SYSIN  DD *
DATA;
  SET ddname.iiifff;
```

where x is the database unit ID, "ddname" is a name that you assign, "tapeprefix" is the high-level qualifier specified in the TAPEPREFIX parameter, "iiifff" is the information area and filename, and "<gdgnumber>" is either the absolute GDG number or the relative GDG number. Note that the time-span and cycle number are not used to access a file on the audit archive tape. The tape contains DETAIL (and/or DAYS) files by default, and observations from those files have been concatenated into one SAS data set on tape.

Let's look at an example. If you specified twelve (12) generations for the AUDITGDG parameter and "PRIMARY" for the TAPEPREFIX parameter, you could retrieve the BATJOB file on the most current audit archive tape by its absolute generation number:

```
    //RETRIEVE DD DSN=PRIMARY.MICS.AUDIT.BATJOB.G0012V00,
    // DISP=OLD
```

or by its relative generation number:

```
    //RETRIEVE DD DSN=PRIMARY.MICS.AUDIT.BATJOB(0),DISP=OLD
```

The relative generation number generally corresponds to the age of the data on the audit archive tape. This correspondence is in relation to the file currently on the CA MICS database unit. For example, the BATJOB data for the current week (i.e., the +1 generation) resides on the CA MICS database unit. PRIMARY.MICS.AUDIT.BATJOB(0) contains the data from last week, and PRIMARY.MICS.AUDIT.BATJOB(-12) contains data from thirteen weeks ago.

AUDIT ARCHIVE RETRIEVAL CONSIDERATIONS

1. All of the CA MICS database files selected during the AUDIT job or WEEK300 step reside on a single audit archive tape. You can reference only one file on a tape volume at a time in a SAS DATA step or procedure. If you need more than one file from a physical tape reel, use the SAS COPY procedure to copy the desired files to disk; then use the disk files for your requirements.

2. Remember that observations from up to ten online cycles were used to create each file on the audit archive tape. These observations often represent more than one week's worth of data; some of them overlap (duplicate) data contained on the audit archive tape from the previous week. The technique of overlapping data ensures that no audit data will be lost if the WEEKLY job (and AUDIT job if being used) is scheduled up to the second or third day in the following week. These duplicate observations must be deleted when more than one audit tape is used.

   The duplicates can be deleted in the following manner:

   o Use the SAS SET statement to concatenate the tape files in a DATA Step.

- Use the CA MICS common data elements (e.g., ENDTS, DAY, MONTH and YEAR) along with other data elements in the selected file as data selection criteria to reduce the data for your requirements.

- Use PROC SORT with the NODUPS option to sort the output work file. Use the appropriate CA MICS sort/sequence macro (% fffSEQ(TS=ssssssss), where "fff" is the file identifier and "ssssssss" is the time-span name (DETAIL, DAYS, etc.)

3. We mentioned above that the files on the audit archive tape are not summarized. You should use the CA MICS Summarization Facility to resummarize the data before using it for your requirements. See the MICF Reference Guide.

4. With the ARCHIVE AUDIT JOB option, you have the additional capability to execute the AUDIT job more frequently than the normal weekly. For example, if you find that the DASD space requirements for keeping 10 cycles of DETAIL (or DAYS) level data in one of your unit databases is too great, then you can reduce the number of cycles retained online by running the AUDIT job more frequently, say 3 times per week (which lets you reduce the number of online cycles to 4 or 5 without risk of losing data). If you implement this approach, be aware that each audit archive tape will contain less data than one might expect (e.g., 5 days of data vs. the default 10 days). Alternatively, you could adjust the value of AUDITCYC in prefix.MICS.PARMS(SITE) in accordance with the frequency at which you execute the AUDIT job, without reducing the number of online cycles. This would avoid writing redundant data. For example, if you execute the AUDIT job every 3 days, you could specify AUDITCYC 3 to only select 3 cycles of each file, but still retain 10 cycles online.

4.3.13.2 The CA MICS History Archive Tapes

The history archive tapes are created during both weekly and monthly processing. Based on your specifications in prefix.MICS.PARMS(JCLDEF), Archive Weekly History processing will occur during the WEEKLY job or in the stand-alone HISTW operational job, and Archive Monthly History processing will occur during the MONTHLY job or in the stand-alone HISTM operational job. The history archive tapes contain observations from the most current cycles of user-designated WEEKS and MONTHS database unit files, respectively.

ESTABLISHING HISTORY ARCHIVING

History archiving is controlled by several CA MICS statements and parameters. The COMPJOB and FILE statements in sharedprefix.MICS.GENLIB (cccGENIN), where "ccc" is the three-character product identifier, determine if weekly and/or monthly archiving files can be created. The COMPJOB statement has indicators that show whether the product is eligible to create weekly and/or monthly archiving files. And the FILE statement determines whether a product file is eligible for weekly and/or monthly archiving. The ADM files are not supported for archiving. You must specify parameters in the JCLDEF and DBMODEL members of prefix.MICS.PARMS to complete the establishment of history archiving. These parameters determine the retention limits of the physical tape reel, the specific weekly and/or monthly files to be archived, and the maximum number of file cycles to be stored on tape.

The JCLDEF parameters are as follows:

o The ARCHIVE HISTW YES parameter specifies that weekly history archiving is to take place.

  - If you specify ARCHIVE HISTW YES JOB, weekly history archiving occurs in the stand-alone HISTW operational job.

  - If you specify ARCHIVE HISTW YES JOB AUTOSUBMIT, the WEEK300 step of the WEEKLY job automatically submits the HISTW operational job to perform weekly history archiving.

  - If you specify ARCHIVE HISTW YES STEP or take the default (i.e., ARCHIVE HISTW YES), weekly history archiving takes place in the WEEK300 step of the WEEKLY job.

- The ARCHIVE HISTM YES parameter specifies that monthly history archiving is to take place.

  - If you specify ARCHIVE HISTM YES JOB, monthly history archiving occurs in the stand-alone HISTW operational job.

  - If you specify ARCHIVE HISTM YES JOB AUTOSUBMIT, the MONTH300 step of the MONTHLY job automatically submits the HISTW operational job to perform monthly history archiving.

  - If you specify ARCHIVE HISTM YES STEP or take the default (i.e., ARCHIVE HISTM YES), monthly history archiving takes place in the MONTH300 step of the MONTHLY job.

- The TAPEPREFIX parameter specifies the high-level qualifier of the data set names on the history archive tape.

- The HISTWGDG parameter specifies the maximum number of generation data group (GDG) entries that CA MICS should build for the weekly history archive tape and establishes a default retention period. Unless there is an overriding specification in the EXPDT parameter of JCLDEF, CA MICS will automatically multiply this number by 7 days and determine the retention period of the tape. For example, if HISTWGDG 7 is specified in JCLDEF, CA MICS will generate a retention period of 49 days. This retention period will apply to each file that is archived during the WEEK300 step.

- The HISTMGDG parameter specifies the maximum number of generation data group (GDG) entries that CA MICS should build for the monthly history archive tape and establishes a default retention period. Unless there is an overriding specification in the EXPDT parameter of JCLDEF, CA MICS will automatically multiply this number by 31 days. For example, if HISTMGDG 5 is specified in JCLDEF, CA MICS will generate a retention period of 155 days. This retention period will apply to each file that is archived during the MONTH300 step.

HISTWGDG and HISTMGDG do NOT equal the maximum number of database file cycles to be stored on the tape. The number of cycles to be kept is determined by the values specified in the following DBMODEL parameters:

- The "wha" parameter on each FILE statement in prefix.MICS.PARMS(DBMODEL) specifies the maximum number of WEEKS cycles that are to be stored on the history archive tape. If the number is zero, the file is not archived.

- The "mha" parameter on each FILE statement in prefix.MICS.PARMS(DBMODEL) specifies the maximum number of MONTH cycles that are to be stored on the history archive tape. If the number is zero, the file is not archived.

The"wha" and "mha" parameters for an active file can easily be changed by the CA MICS System Administrator. Simply alter the DBMODEL parameters, then submit prefix.MICS.CNTL(CYCLEGEN). Section 6.2.3, Tailoring Archive Files, in the System Modification Guide explains how to add and delete history archive files.

HOW THE HISTORY ARCHIVE TAPES ARE CREATED

The weekly and monthly history archive tapes are created in the HISTW job or in step WEEK300 of the WEEKLY job (for weekly history), and in the HISTM job or in step MONTH300 of the MONTHLY job (for monthly history), respectively. The logic used to create the tapes is described below:

```
    Old master + updates = new master
```

The old master is the (0) generation level of a history archive file. It resides on tape and contains observations from up to "n" cycles, where "n" is the limit specified in the "wha" or "mha" parameter in DBMODEL. The observations in these cycles have been concatenated into one SAS data set on tape.

The updates are the observations in the 01 cycle of the database unit file.

The new master is the (+1) generation level of the history archive file. A SAS SET statement is used to concatenate the observations from the 01 timespan cycle with those on the old master tape. Any observation on the old master tape that exceeds the "wha" or "mha" limit is dropped during the DATA step. Note that the new history archive file is neither summarized nor sorted during this process (see HISTORY ARCHIVE RETRIEVAL CONSIDERATIONS below).

Each file on the weekly history archive tape is entered in the OS catalog as:

```
    tapeprefix.MICS.HISTW.iiifff.GggggV00
```

Each file on the monthly history archive tape is entered in the OS catalog as:

```
    tapeprefix.MICS.HISTM.iiifff.GggggV00
```

where "tapeprefix" is the high-level qualifier specified in the TAPEPREFIX parameter of prefix.MICS.PARMS(JCLDEF), "iiifff" is the information area and filename, and "gggg" is the absolute number of a generation data group (GDG).

RETRIEVING DATA FROM THE HISTORY ARCHIVE TAPES

The history archive tapes may be accessed through a MICF inquiry or a batch job. Refer to Section 4.3.13.3, Using CA MICS Archive Tapes With MICF, for more details about accessing the archive tapes in MICF inquiries.

The JCL specifications that are used to access the files on the history archive tapes are different from those used to access the files on the database unit. Database unit files are referenced as:

```
    //S01 EXEC MICSSHRx
    //SYSIN  DD *
    DATA;
      SET &iiit..HARCPUnn;
```

where x is the database unit ID, "&iiit" is a CA MICS/SAS macro variable representing the information area (iii), timespan (t) of the file to be accessed, and nnn is cycle. Coding the macro variable as compared to a timespan ddname eliminates the need to know if a particular file has been defined to DETAIL, DETAIL2 or another split database file. Refer to Section 2.3.3.3.2.4 for information about the CA MICS Database Split Table.

Files on the weekly history archive tape are referenced as:

```
    //S01 EXEC MICSNDBx
    //ddname DD DSN=tapeprefix.MICS.HISTW.iiifff<gdgnumber>,
    // DISP=OLD
    //SYSIN  DD *
    DATA;
```

```
      SET ddname.iiifff;
```

Files on the monthly history archive tape are referenced in a similar manner:

```
    //ddname DD DSN=tapeprefix.MICS.HISTM.iiifff<gdgnumber>,
    // DISP=OLD
```

where x is the database unit ID, "ddname" is assigned by you, "tapeprefix" is the high-level qualifier specified in the TAPEPREFIX parameter, "iiifff" is the information area and filename, and "<gdgnumber>" is either the absolute GDG number or the relative GDG number. Note that the timespan and cycle number are not used to access these files. The observations from all of the timespan cycles have been concatenated into one SAS data set on the tape.

Let's look at an example. If you specified twelve (12) generations for the HISTMGDG parameter and "PRIMARY" for the TAPEPREFIX parameter, you could retrieve the BATJOB file from the current monthly history archive tape by its absolute generation number:

```
    //RETRIEVE DD DSN=PRIMARY.MICS.HISTM.BATJOB.G0012V00,
    // DISP=OLD
```

or by its relative generation number:

```
    //RETRIEVE DD DSN=PRIMARY.MICS.HISTM.BATJOB(0),DISP=OLD
```

HISTORY ARCHIVE RETRIEVAL CONSIDERATIONS

1. You can reference only one file on a tape volume at a time in a SAS DATA step or procedure. If you need more than one file from a history archive tape, use the SAS COPY procedure to copy the desired files to disk; then use the disk files for your requirements.

2. Remember that each file on the history archive tape contains observations from multiple cycles of weekly (or monthly) data. The majority of these observations duplicate data contained on the (-1) and older generations of the history archive tape because they are transferred from the old master to the new master until they reached their cutoff period. You should only need to reference one generation of a history archive file since it would contain all information back to the cutoff date as defined in DBMODEL.

3. We mentioned above that files on the history archive tape are not summarized as they are written to tape. That is, the tape contains summarized data, but since it was built from multiple CA MICS file cycles, there may be multiple records for a given key combination (e.g., year, month, zone, etc.). You should use the CA MICS Summarization Facility to resummarize the data before using it for your requirements.

4.3.13.3 Using CA MICS Archive Tapes With MICF

If you specify ARCHIVE for CA MICS file cycle on the execution-time parameters panel, MICF will process the archive tape in place of the corresponding online file--weekly archive history for WEEKS files, monthly archive history for MONTHS files, and archive audit for DETAIL/DAYS files.

Most of the inquiries we distribute let you override CA MICS file cycles at execution; therefore, most distributed inquiries automatically support archive tape.

The following steps describe accessing the CA MICS archive tapes with a new inquiry.

1. Compose and test your inquiry against the CA MICS online database as usual. Specify YES for the option to allow execution-time override for CA MICS file cycles.

2. When you execute the inquiry, specify A (ARCHIVE) instead of a cycle number. If an archive tape is available for the file, timespan, and DBID you are processing, MICF will allocate and read the corresponding archive tape file.

Note the following important restrictions on MICF archive tape access:

- o Dynamic tape drive allocation is not allowed. You must execute your inquiry in batch mode and you cannot use archive tape with the MICF production reporting interface.

- o You can override the archive tape generation data group (the default is generation 0); however, only one archive tape is processed (i.e., you cannot concatenate multiple archive audit generations.) Specify ARCHIVE-n where n is the generation number (for example, ARCHIVE-1 for the "-1" generation.)

- o All CA MICS file selections in the inquiry that allow execution-time cycle override must specify CA MICS files present in the corresponding timespan's archives, that is, archive access must be valid for every file to which the execution-time override applies.

- o Archive file access is not available to user-written report inquiries because you do not specify cycles for the CA MICS File Allocation option. You should continue to use the External File Allocation option to allocate archive tape files for user-written reports.

4.3.14 How to Turn on Source in a CA MICS Environment

It is occasionally necessary to expand the SAS source listing in order to find the context of an error in a CA MICS operational job. This process is called turning on source and is accomplished in CA MICS by coding the following symbolic parameter on the EXEC statement of the problem step of the job:

```
    SYSPARM=DEBUG
```

If the problem step executes the MICSCOM, MICSDBx, MICSDOC, MICSNDBX, MICSSHRX , MICSDM, MICSDUx, or MICSGU procedure, turn on source by coding SYSPARM=DEBUG on the EXEC statement as follows:

```
    //STEP1    EXEC procedurename,SYSPARM=DEBUG
```

USAGE CONSIDERATIONS:

If a SYSPARM value has already been specified on the EXEC statement, DEBUG must be specified as the first value, a blank must be used as the delimiter, and the whole symbolic parameter value must be enclosed in quotes. The following example demonstrates this:

```
    //FINRECAP  EXEC MICSCOM,SYSPARM='DEBUG NODB'
```

Review the LINES= parameter in your JCL to avoid an abend due to excessive lines.

Remove the SYSPARM value as soon as the problem step has been executed in order to reduce paper consumption in other CA MICS operational jobs.

4.3.15 CA MICS Operational Space Utilization Report

The CA MICS Operational Space Utilization report is an optional report that can be output during the CA MICS daily update. It provides information about allocated and free space for the following data sets:

o Database
o Incremental
o Multwork
o Restart Work
o SAS work

This information can be used to help estimate workspace size by identifying data sets that are approaching capacity or have been over-allocated. It can also identify the need to adjust the size of the CA MICS databases.

The CA MICS Work Space Utilization is available for the daily update process for CA MICS components, incremental update jobs, and the Accounting and Chargeback DAY199 step.

New field-developed applications (FDAs) will automatically include support for the Operational Space Utilization report and existing FDAs can easily be modified to include support.

The remainder of this chapter covers the following topics:

1 - Activation
2 - Report Sample
3 - Field-Developed Application Support

4.3.15.1 Specification

The CA MICS Operational Space Utilization report is controlled by the WORKSPACE statement in prefix.MICS.PARMS(EXECDEF).

```
REPORT WORKSPACE YES | NO
```

Specify YES to activate it, NO to deactivate it. The default if unspecified is NO.

If activated, a report will be generated to the MICSLOG for,

o Daily update steps of CA MICS components (DAYnnn)

o FDAs, if support has been added (see section 4.3.15.3)

o Incremental update jobs

o Daily Accounting and Chargeback update (DAY199)

The CA MICS Operational Space Utilization report will not be generated for any other DAILY steps or for other CA MICS operational jobs (for example, WEEKLY or MONTHLY).

Note: This information can optionally be written to a file, ADMSPC. The Operational Space Utilization Report and the ADMSPC file can each be created independently of the other. For more information on this file, see the Planning, Installation, Operation and Maintenance Guide section 4.3.16.

4.3.15.2 Report Sample

Figure 4-15 is a sample of the CA MICS Operational Space Utilization report.

```
-------------------------------------------------------------
15.20.51 BAS64300I *========================================================================================================*
15.20.51 BAS64301I |                                  CA MICS Operational Space Utilization                                 |
15.20.51 BAS64301I |                                                                                                        |
15.20.51 BAS64301I |     Work                          Tracks       Tracks       Tracks     Pct Trks   Max Trks     Max Pct |
15.20.51 BAS64301I |  File Class    Libname   Vols      Alloc        Used         Free        Used       Used        Used   |
15.20.51 BAS64301I | ------------   --------  ----   ----------   ----------   ----------   --------  ----------   -------- |
15.20.51 BAS64301I | SAS Work       WORK         1        1,500          491        1,009     32.73%       1,426     95.07% |
15.20.51 BAS64301I |                                                                                                        |
15.20.51 BAS64301I | MultWork       WORK01       1          750          505          245     67.33%         560     74.67% |
15.20.51 BAS64301I |                WORK02       1          750          489          261     65.20%         519     69.20% |
15.20.51 BAS64301I |                WORK03       1          500          381          119     76.20%         467     93.40% |
15.20.51 BAS64301I |                WORK04       1          500          371          129     74.20%         450     90.00% |
15.20.51 BAS64301I |                WORK05       1          500           12          488      2.40%         296     59.20% |
15.20.51 BAS64301I |                WORK06       1          500          135          365     27.00%         382     76.40% |
15.20.51 BAS64301I |                          ====   ==========   ==========   ==========   ========  ==========   ======== |
15.20.51 BAS64301I | MultWork                             3,500        1,893        1,607     54.09%       2,674     76.40% |
15.20.51 BAS64301I |                                                                                                        |
15.20.51 BAS64301I | Incremental    IUDAYS       1          225           90          135     40.00%          90     40.00% |
15.20.51 BAS64301I |                IUDETAIL     1        1,425        1,420            5     99.65%       1,420     99.65% |
15.20.51 BAS64301I |                                                                                                        |
15.20.51 BAS64301I | Database       DAYS         1        1,500        1,330          170     88.67%       1,473     98.20% |
15.20.51 BAS64301I |                DETAIL       1        2,610        2,498          112     95.71%       2,499     95.75% |
15.20.51 BAS64301I |                MONTHS       1          150          133           17     88.67%         133     88.67% |
15.20.51 BAS64301I |                WEEKS        1          285          258           27     90.53%         258     90.53% |
15.20.51 BAS64301I |                                                                                                        |
15.20.51 BAS64301I *========================================================================================================*
```

Figure 4-15. CA MICS Work Space Utilization Report

```
Report field descriptions
------------------------

Work File Class -- Groups work data set by functional usage.

LIBNAME -- The name used by SAS to reference the
   data set. This is the same as the DD name.

Volumes -- The number of volumes on which the data set
   resides.

Tracks Allocated -- The number of tracks in the data set at
   time the report was run.

Tracks Used --  The number of tracks currently in use.
   The space within a SAS data library is managed internally
   by SAS.  When a member of a data library is deleted, SAS
   reclaims this space for new members. Because of this, the
   free space reported  may not equal the free space reported
   by z/OS. This value represents the percent of free space
   in the SAS data library at the time the report was run.

Tracks Free --

Pct Trks Used --

Max Trks Used --
   This value is the highest amount of space that has been
   used used by SAS and indicates a high water mark of usage.

Max Pct Used --

Minimum % Free -- The minimum amount of free space in the
   data set.


% Over Allocated -- Percent of space the is unused. Unused
   space resides above the high water mark and is space that
   SAS has not accessed.

   Large values may indicate that the data library
   has been over allocated and the space allocations may
   be reduced.
```

4.3.15.3 Field-Developed Application Support

The CA MICS Operational Space Utilization report will be generated for all CA MICS Data Integration Applications (DIAs) and for any new field-developed applications (FDAs).

For FDAs created before the introduction of the CA MICS Work Space Utilization report, a small manual change needs to be made to provide support. Insert the flagged line in sharedprefix.MICS.INCLLIB(DAYnnn):

```
%INCLUDE  SOURCE(DYcccSUM);
%INCLUDE  SOURCE(NSPURWRU);  *<-- Work Space Report;
%INCLUDE  SOURCE(DYcccAGE);
```

Note that the invocation of NSPURWRU should come after the DYcccSUM (daily summarization) and before DYcccAGE (daily aging), where ccc is the component ID. This gives you a snapshot of the maximum work space in use.

4.3.16 Space Utilization File (ADMSPC)

During the INCRccc and DAILY job steps (by request) the ADMSPC file can be created to provide a snapshot of CA MICS work and database space usage. This allows the CA MICS administrator to track utilization over time to ensure adequate space is being allocated to fulfill operational processing requirements. A frequent review and adjustment of space usage will help avoid insufficient space-related abends as well as ensure that space is not overallocated. Over allocating space reduces the disk space available to other batch jobs running simultaneously with the CA MICS incremental or DAILY update job.

You can access and report the ADMSPC file with a MICF inquiry. See the distributed BASLA5 inquiry for an example of accessing the ADMSPC file.

Additionally, you can create the Space Utilization Report during the INCRccc and DAILY job steps by specifying the WORKSPACE REPORT YES statement in prefix.MICS.PARMS(EXECDEF) of the database unit. The report can be activated independently of the ADMSPC file.

4.4 Administering CA MICS

To administer CA MICS, the System Administrator needs to set up and maintain the online user environment.

To do so, the System Administrator needs to specify CA MICS Administrator Facility (MAF) parameters and grant users' authorization to perform functions using the Authorization Assignment features of MAF. MAF is an option of the CA MICS Workstation Facility.

The following topics are addressed in this section:

1 - CA MICS Workstation Facility
2 - MAF Parameters
3 - Authorization Assignment
4 - CA MICS Administrator Utilities
5 - Data Entry
6 - Software Maintenance
7 - Adding Your Load Libraries to CA MICS
8 - CA MICS Administration in an SMS-Controlled Environment

For information on MICF administration, see the MICF Reference Guide.

4.4.1 CA MICS Workstation Facility

```
---------------------  CA MICS Workstation Facility (MWF)  --------------------


                      Option ===>



                             1 - Documentation Access (DOC)
                                 Browse, print, and/or cross reference CA MICS documentation.
                             2 - CA MICS Information Center Facility (MICF)
                                 Compose and/or replay CA MICS database inquiries.
                             3 - SAS With CA MICS Libraries (MSAS)
                                 Use interactive SAS with CA MICS libraries and macros.
                             4 - Management Support Applications (APPL)
                                 Accounting and Chargeback, Capacity Planner,
                                 Performance Manager and StorageMate.
                             5 - CA MICS Administrator Facility (MAF)
                                 Operational status and tracking, installation, modification,
                                 maintenance, and authorization.
                             T - Tutorial
                             X - Exit


                      .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

CA MICS provides online access to the Database, documentation, and scheduled maintenance (the Product Service Packs or "PSPs") through the CA MICS Workstation Facility (MWF). The Workstation's dialogs run as ISPF applications.

ACTIVATING MWF

The initial invocation of MWF automatically proceeds to the dialog for CA MICS Administrator Facility Parameters (MAF option 0). This allows for the immediate definition of required MWF parameters to operate the dialogs and also determines where the parameter information itself will be stored.

Successive MWF invocations use these stored parameter definitions, so the first panel displayed is the MWF primary options panel (above).

MAF Parameters apply to all users of the online facilities of CA MICS. Individual users can override selected MAF parameters. The first user to invoke MWF will be required to specify this information for MWF use, therefore it is important to complete these parameters correctly during the initial invocation. The parameter information is then carried in ISPF tables that are stored into and retrieved from sharedprefix.MICS.ISPTLIB. For this reason, you MUST specify the correct name of your sharedprefix.MICS.ISPTLIB data set during initialization.

Upon initial invocation, each new user of MWF is prompted to establish a user-level ISPF tables data set. The userid.MICS.USER.TABLES data set is then used to store information unique to each user, from session to session. The userid.MICS.USER.TABLES data set is discussed later in this section in more detail, but is also mentioned here because the Administrator that activates MWF globally is also a user, with his or her own userid.MICS.USER.TABLES data set. In addition, you can change the default userid.MICS.USER.TABLES data set name. The DSN applies to ALL users. You should NOT change the DSN after other users start using CA MICS online facilities.

```
---------------------  CA MICS Administrator Facility (MAF) ---------------------


Option ===>




         0 - CA MICS Administrator Facility Parameters
             Change default CA MICS environmental parameters.
         1 - Operational Status and Tracking
             Submit operational jobs and check Database status.
         2 - PSP Online Services
             Accept and apply PSP maintenance.
         3 - Authorization Assignment
             Review and adjust user authorization to CA MICS facilities.
         4 - CA MICS Administrator Utilities
             Submit CA MICS Administrator Utilities.



 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

CA MICS ADMINISTRATOR FACILITY PARAMETERS (MAF OPTION 0)

The MAF Parameters panel (see sample at the beginning of section 4.4.2) is displayed when the CA MICS Workstation Facility (MWF) parameter table (MWF$SITE) is not found, which is normal if you are installing the CA MICS ISPF environment (MWF) the first time.

MWF initialization cannot continue until you select each option from this menu and enter the required parameters. Alternatively, you can enter the CANCEL command to terminate MWF initialization.

The sharedprefix.MICS.ISPTLIB(MWF$SITE) ISPF table must be available to CA MICS through the normal ISPF tables library ddname, ISPTLIB. Follow the instructions in the Setting Up the CA MICS ISPF Environment section (3.3.6) of this guide to ensure that MWF$SITE can be found through the ISPTLIB logical ddname.

CA MICS ADMINISTRATOR FACILITY (MAF)

The CA MICS System Administrator uses the CA MICS Administrator Facility (MAF) to support and control the

CA MICS complex. Access to this process is limited to authorized users. MAF applications are discussed in other sections of this guide.

USER ACCESS TO MWF

Most functions provided by the Workstation Facility require initialization before they can be used. Some of this initialization is accomplished by the standard JCLGEN and Component Generation processes involved in installing and maintaining CA MICS.

Additional information is then specified to the online administrative- and user-level parameter panels in each MWF application. See the MICF User Guide, MICF Reference Guide, and Document Access Guide for more information. See the MAF Parameters section (4.4.2) of this guide for information about the CA MICS Administrator Facility (MAF) parameters.

Refer to individual product guides for more information on the online services available with each management support application.

The only administrative panel for the MSAS dialog is option 3 of the MAF Parameters menu (MWF;MAF;0;3). Also, see the Data Entry section (4.4.5) of this guide for information on data entry through the MSAS dialog.

4.4.2 MAF Parameters

```
-------------------  CA MICS Administrator Facility Parameters ------------------
Option ===>


        1 - CA MICS Data Sets for ISPF Applications
            Specify sharedprefix, ISPF tables data set name, and DDNAMEs
        2 - CA MICS Global Parameters for ISPF Applications
            Specify operational options and terminal display attributes
        3 - SAS Global Parameters for ISPF Applications
            Specify SAS library data set names and SAS options
        4 - Data Set Allocation Parameters
            Specify parameters used to allocate ISPF application data sets


--------------------------------------------------------------------------------
```

The CA MICS Administrator Facility Parameters process (MAF option 0) provides access to the specifications that control the operation and appearance of the CA MICS Workstation Facility (MWF) and the interactive CA MICS applications available under MWF. Most MAF parameters apply to all CA MICS users (i.e., they are specified only once and apply globally). Selected MAF parameters are default values that individual CA MICS users can override.

A fifth option will be displayed on this panel during your initialization of MWF. The fifth option lets you tailor the CA MICS sharedprefix data set names. Once MWF has been initialized, the option will no longer be available. See the online tutorial for details.

The standard MAF Parameters menu options are described in the sections that follow.

4.4.2.1 CA MICS Data Sets for ISPF Applications

```
--------------------  CA MICS Data Sets for ISPF Applications -------------------
Command ===>

Data set name conventions:
  CA MICS sharedprefix                         ===> CPM981.CMN
  High level qualifier for user data set names ===> ________ (Optional)
  Include "MICS." in data set names            ===> NO       (YES/NO)

CA MICS ISPF tables data set ===> 'CPM981.CMN.MICS.ISPTLIB'
  (Default 'sharedprefix.MICS.ISPTLIB')

CA MICS ISPF tables DDNAMEs:
  Table input DDNAME           ===> ISPTLIB  (Default ISPTLIB)
  Read-only table input DDNAME ===> MWFTLIBR (Default MWFTLIBR)
  Table output DDNAME          ===> MWFTABL  (Default MWFTABL)
  User tables DDNAME           ===> MWFUTABL (Default MWFUTABL)


--------------------------------------------------------------------------------
```

Use the CA MICS Data Sets for ISPF Applications panel to specify the CA MICS ISPF tables data set name (the sharedprefix.MICS.ISPTLIB data set), the complex level data set naming conventions, and conventions for naming user level data sets (private catalogs).

DATA ENTRY INSTRUCTIONS

CA MICS sharedprefix - Required. No default. 1-19 character data set name prefix. This is the data set name prefix for shared (or CA MICS complex level) data sets.

High level qualifier for user data set names - Optional. No default. 1-8 characters.

o Private (user level) CA MICS data set names are generated with the user's TSO prefix or TSO logon-id (if prefix = logon-id) as the high level qualifier (e.g., DSN=userid.MICS.USER.TABLES).

o Specify this parameter if your installation conventions require a high level data set name qualifier other than the user's prefix or logon-id (e.g., DSN=xxxxxxxx.userid.MICS.USER.TABLES).

- o You may also specify an ISPF variable name from the shared pool if you store the non-userid prefix into this variable before invoking MWF. Specify ?name where name is the shared pool variable name that contains the high level qualifier.

- o The CA MICS user data set name prefix is derived as follows:

  - If High level qualifier is specified, then DSN prefix is xxxxxxxx.userid where xxxxxxxx is the high level qualifier value.

  - Else, if the TSO prefix is different from userid, then DSN prefix is pppppppp.userid where pppppppp is the TSO user prefix.

  - Else, DSN prefix is userid.

Include "MICS." in data set names - Required. Defaults to YES. Specify YES (or Y) to generate default data set names with "MICS." as a data set name index level (DSN=sharedprefix.MICS....). Specify NO (or N) to construct default data set names without "MICS." as a data set name index level (DSN=sharedprefix.....).

CA MICS ISPF tables data set - Required. Default value is derived from your specifications for the CA MICS sharedprefix and Include "MICS." in data set names parameters (e.g., 'sharedprefix.MICS.ISPTLIB'). 1-44 character fully qualified data set name enclosed in quotes. This is the name of the data set containing the CA MICS ISPF tables (i.e., the CA MICS ISPTLIB data set).

The following four fields are for display purposes only and cannot be updated. These DDNAMEs are essential to successful CA MICS processing.

- o Table input DDNAME - ISPTLIB. This is the ddname used for reading ISPF tables.

- o Read-only table input DDNAME - MWFTLIBR. This is the ddname used for read-only access to ISPF tables. The ISPF tables data set name is dynamically concatenated to a temporary data set allocated to this ddname (this prevents enqueue lock-outs for read-only access).

- o Table output DDNAME - MWFTABL. This is the ddname used for updating ISPF tables. The ISPF tables data set is

```
  dynamically allocated to this ddname.

o User tables DDNAME - MWFUTABL.  The private (or user
  level) ISPF tables data set is dynamically allocated to
  this ddname.
```

4.4.2.2 CA MICS Global Parameters for ISPF Applications

```
----------------  CA MICS Global Parameters for ISPF Applications  --------------
Command ===>

******************************** Top of data ******************************
MWF menu specifications:  Provide PDF selection under MWF ===> NO  (YES/NO)
                          PDF primary option panel name   ===> ISR@PRIM

CA MICS complex name:  Title (or name) for SAS log
    ===> TESTING AT SAS V913
                       Title (or name) for reports
    ===> TESTING AT SAS V913

DSPRINT services parameters:  DSPRINT command name   ===> DSPRINT
                              Default DSPRINT device ===> ________

Fill blank fields with the following character ===> _   (1 Character)
Automatically move cursor to next field        ===> NO  (YES/NO)
Include printed output in DTF index      ===> NO  (YES/NO)
Update shared DTF index                  ===> YES (YES/NO)

Service Desk Interface Options
   Active ===> NO   (Yes/No)

CA MICS ISPF application trace options:
  Trace ISPF table status  ===> YES (YES/NO)  (Defaults to YES)
  Trace CPU resource usage ===> NO  (YES/NO)  (Defaults to NO)
  Trace ISPF dialog flow   ===> NO  (YES/NO)  (Defaults to NO)
******************************* Bottom of data *****************************
```

Use the CA MICS Global Parameters for ISPF Applications panel to specify parameters related to the ISPF environment provided by the CA MICS Workstation Facility (MWF). These include whether ISPF Program Development Facility (PDF) is selectable under MWF and the PDF primary options panel name, the name of the CA MICS complex for SAS log and report titles, and the data entry field "pad" character. This panel can also be used to activate certain trace (or debugging) facilities for the CA MICS ISPF applications.

DATA ENTRY INSTRUCTIONS

Provide PDF selection under MWF - Required. Defaults to NO. Specify YES (or Y) to include ISPF/PDF as an option on the MWF menu (that is, PDF can be selected from MWF). Specify NO (or N) if MWF is selected from your site's ISPF primary options menu (that is, PDF should not be selected from MWF).

PDF primary option panel name - Required if YES is specified for Provide PDF selection under MWF. Defaults to ISR@PRIM. 1-8 character name. This is the name of the panel to be displayed when the ISPF/PDF option is selected from the MWF menu.

Title (or name) for SAS log - Required. No default. 1-60 characters; ' (single quote) and " (double quote) are not allowed. This appears as the title on SAS log output from CA MICS interactive applications and generally corresponds to the CA MICS complex name (like Production CA MICS Complex).

Title (or name) for reports - Required. No default. 1-60 characters; ' (single quote) and " (double quote) are not allowed. This appears as the title for reports generated under CA MICS interactive applications and generally corresponds to the data center name (like ABC Company).

DSPRINT command name - Optional. Defaults to DSPRINT. 1-8 character name. This is the TSO command used to invoke DSPRINT services. Leave this field blank to indicate that a DSPRINT command is not supported.

Default DSPRINT device - Optional. No default. 1-8 character name. This is the default DSPRINT (that is, local 3270 printer) device name.

Fill blank fields with the following character - Optional. Defaults to _ (underscore). 1 character. This character appears in empty data entry fields. Leave this parameter blank to leave empty fields blank.

Automatically move cursor to next field - Required. Defaults to NO. Specify YES (or Y) to automatically move the cursor to the next data entry field after you enter a value (that is, automatically skip the cursor over text fields). Specify NO (or N) to disable this feature.

Include printed output in DTF index - Required. Defaults to NO. Specifying YES (or Y) will cause any output (report/micslog/saslog) generated by a MICF inquiry to be written to the user's private DTF index. Including report output in the DTF index facilitates the use of the CA MICS Q&R RMWSLOAD utility to download report as well as CSV output to the workstation for subsequent publishing. Specify NO (or N) to disable this feature. Note, this setting has no effect on output generated by Production Reporting Jobstream. Report output is always written to the Production Reporting index.

Update shared DTF index - Required. Defaults to YES. Specifying YES (or Y) causes any entries recorded to the user's private DTF index to also be written to the complex level DTF index. Specify NO (or N) to disable this feature. Disabling this feature only affects user generated output. Output generated by Production Reporting Jobstream is always written to the complex level (shared) DTF index data set.

The following parameters control the options for the Service Desk Interface. Note that the URL, Userid and Password for Service Desk should be stored in the CA90's Soap Server. See Section 4.0 for more information on Service Desk interface setup.

Active (Yes or No). Defaults to NO. Specify YES to activate or NO to disable the CA MICS Service Desk interface for all MWF applications.

The following parameters control trace (or debugging) facilities of the ISPF applications. Do not change the defaults for these parameters unless instructed to do so by CA Technical Support. Trace options can consume significant system resources and generate voluminous outputs. You can make temporary changes to these parameters using the MWFTRACE command.

Trace ISPF table status - Required. Defaults to YES. Specify YES (or Y) to track the status of ISPF tables used by the ISPF applications. You can review ISPF table status using the MWFTRACE command.

Trace CPU resource usage - Required. Defaults to NO. Specify YES (or Y) to log to your ISPF log cumulative CPU time and other resource usage measurements at selected points in ISPF application processing.

WARNING! This option can generate voluminous outputs and can require increasing the size of your ISPF log.

Trace ISPF dialog flow - Required. Defaults to NO. Specify YES (or Y) to log the flow of ISPF application logic to your ISPF log.

WARNING! Do not specify YES unless instructed to do so by CA Technical Support. This option will generate voluminous output, require increasing the size of your ISPF log, and seriously degrade your response time.

4.4.2.3 SAS Global Parameters for ISPF Applications

```
----------------  SAS Global Parameters for ISPF Applications  ----------------

 Command ===>


SAS version               ===> SAS913   (Example:  SAS802)
Entry point               ===> SAS      (Example:  SASHOST)
Foreground interface      ===> SASCP    (SASCP/CALL)
SAS/Graph Installed?      ===> YES      (YES/NO)

SAS libraries (enter fully qualified data set names enclosed in quotes):
   Load library           ===> 'CPM981.SAS92.TS2M0.LIBRARY'
   Auto-call macros       ===> 'CPM981.SAS92.TS2M0.W0.AUTOLIB'
   Help library           ===> 'CPM981.SAS92.TS2M0.ENW0.SASHELP'
   Messages library       ===> 'CPM981.SAS92.TS2M0.ENW0.SASMSG'
   Opt. config file 1     ===> 'CPM981.SAS92.TS2M0.CONFIG(BATCH)'
   Opt. config file 2     ===> 'CPM981.SAS92.TS2M0.CONFIG(COMMON)'
   Opt. config file 3     ===> 'CPM981.SAS92.TS2M0.CONFIG(ENW0)'
   Opt. config file 4     ===> 'CPM981.SAS92.TS2M0.CONFIG(SITE)'

Optional installation SAS related data sets (enter fully qualified DSNAMES):
   Augment load library   ===> 'SYSHARED.SAS.LOAD.MODULES'
           Macro library  ===> 'SYSHARED.SAS.MACROS'
           Object library ===> 'SYSHARED.SAS.FORMATS'
           Source library ===> 'SYSHARED.SAS.SOURCE'
   Override load library  ===> ______________________________________________
            Macro library ===> ______________________________________________
            Object library ===> ______________________________________________
            Source library ===> ______________________________________________




.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Use the SAS Global Parameters for ISPF Applications panel to specify the SAS libraries and parameters to be used in interactive or batch execution of SAS within the CA MICS ISPF applications.

DATA ENTRY INSTRUCTIONS

SAS version - Required. Three alphabetic characters followed by three numeric characters. The three numeric characters must be a valid SAS version number. Refer to the CA MICS System Administrator Guide for information about the levels of SAS currently supported.

Entry point - Required. Defaults to SASXAL. 1-8 character name (e.g., SASXAL or SASHOST). This is the program name used to invoke SAS, either interactively or in batch, within the ISPF applications.

Foreground interface - Required. Defaults to SASCP. SASCP or CALL. Specify SASCP (or S) to use the SASCP TSO command to invoke SAS in foreground. Specify CALL (or C) to use the TSO CALL command to invoke SAS in foreground. Warning: If you specify CALL for this parameter, you will not be able to use the MSAS option or execute MICF inquiries under SAS/DMS.

SAS/Graph Installed? - Required. Defaults to YES. Specify YES if SAS/Graph is installed. Specify NO if SAS/Graph is not installed. Specifying NO will cause any MICF inquiries that produce color graphics to be omitted from the display of the MICF Shared Inquiry Catalog.

SAS libraries - No default. The following parameters define the libraries for invoking SAS, either interactively or in batch, within the ISPF applications. The parameters are 1-44 character fully qualified data set names enclosed in quotes. Each refers to a SAS system library established for CA MICS processing.

- o Load library - Required. The SAS load module library data set name.

- o Auto-call macros - Required. The SAS auto-call macro library data set name.

- o Help library - Required. The SAS help library data set name.

- o Messages library - Required. The SAS messages library data set name.

- o Opt. config file 1 - 4 - Optional. You can use this parameter to point CA MICS to your installation's SAS parameter defaults so that CA MICS applications will use the same defaults as your other SAS applications. CA MICS will override your installation default SAS parameters as required. See the sharedprefix.MICS.PARMS members MABATCH, MATSO, and ICFBATCH for details. Specify a fully-qualified data set name and member name if the configuration file is a member of a partitioned data set (PDS).

Optional installation SAS related data sets - No defaults. The following parameters define installation data sets CA MICS will allocate for SAS processing, either interactively or in batch, within the ISPF applications. The parameters are 1-44 character fully qualified data set names enclosed in quotes.

You can use the installation data set parameters to add your installation's load modules, SAS formats, SAS macros, and/or SAS source modules to CA MICS ISPF application processing. You can also override CA MICS modules by concatenating replacement modules in front of the distributed CA MICS modules. Two installation data sets are supported for each type of allocation.

- o Use the augmenting library parameter to make installation modules available to the CA MICS ISPF applications.

- o Use the override library parameter ONLY to replace CA MICS modules, for example, to implement user modifications to CA MICS.

The augmenting installation data set parameters are:

- o Augment load library - Optional. Use this parameter to make your SAS related load modules available to the CA MICS ISPF applications.

- o Augment macro library - Optional. Use this parameter to make your SAS macros available to the CA MICS ISPF applications.

- o Augment object library - Optional. Use this parameter to make your user-defined SAS formats, compiled SAS macros, and other SAS objects available to the CA MICS ISPF applications.

- o Augment source library - Optional. Use this parameter to make your SAS source members available to the CA MICS ISPF applications.

The overriding installation data set parameters are:

- o Override load library - Optional. Use this parameter to override distributed CA MICS load modules.

- o Override macro library - Optional. Use this parameter to override distributed CA MICS auto-call macro members.

- Override object library - Optional. Use this parameter to override distributed CA MICS user-defined SAS formats, compiled SAS macros, and other SAS objects.

- Override source library - Optional. Use this parameter to override distributed CA MICS source modules.

4.4.2.4 Data Set Allocation Parameters

```
----------------------  Data Set Allocation Parameters  ----------------------
Command ===> RIGHT                                            Scroll ===> CSR

SCROLL RIGHT TO ENTER SMS PARAMETERS

Line Cmds: S Select data set to specify additional DASD allocation parameters

      Data                                        --  Space Alloc. --    DASD
Cmd    Set          Data Set Name Suffix          Typ  Prim  2nd  Dir    Unit
 -  --------  ----------------------------------  --- ----- ----- ---  --------
 _  BDTFOUT   MICS.USER.BATCH***.DTFOUT           CYL 2     1     25   ________
 _  BICFUSER  MICS.USER.BATCH***.ICFUSER          TRK 2     2     ___  ________
 _  BLOG      MICS.USER.BATCH***.LOG              TRK 2     2     ___  ________
 _  BMICSLOG  MICS.USER.BATCH***.MICSLOG          TRK 2     2     ___  ________
 _  BREPORT   MICS.USER.BATCH***.REPORT           TRK 2     2     ___  ________
 _  CAPACITY  MICS.USER.CAPACITY                  TRK 20    10    ___  ________
 _  CAPPDS    MICS.USER.CAPPDS                    TRK 5     5     ___  ________
 _  DTFINDEX  MICS.USER.DTF.INDEX                 TRK 5     5     ___  ________
 _  DTFOUT    MICS.USER.DTFOUT                    CYL 2     1     25_  ________
 _  ISPCTLX   MICS.USER.TEMP*.CNTL                TRK 2     5     ___  ________
 _  LOG       MICS.USER.LOG                       TRK 2     2     ___  ________
 _  MAFCNTL   MICS.USER.BATCH***.MAFCNTL          TRK 2     2     ___  ________
 _  MICSLOG   MICS.USER.MICSLOG                   TRK 2     2     ___  ________
 _  OUTPUT    MICS.USER.OUTPUT                    CYL 2     1     25   ________
 _  PCATALOG  __________________________________  TRK 2     1     1    ________
 _  PDTFINDX  __________________________________  TRK 5     5     ___  ________
 _  PIXFOUT   __________________________________  CYL 2     1     25   ________
 _  PMICSLOG  __________________________________  TRK 2     2     ___  ________
 _  PMWFPDS   __________________________________  CYL 2     1     25   ________
 _  PMWFSAS   __________________________________  CYL 2     1     ___  ________
 _  PREPORT   __________________________________  TRK 10    5     ___  ________
 _  PRINT     MICS.USER.BATCH***.PRINT            TRK 2     2     ___  ________
 _  REPORT    MICS.USER.REPORT                    TRK 2     2     ___  ________
 _  SASPROF   MICS.USER.SAS.PROFILE               TRK 2     5     ___  ________
 _  SORTWK    __________________________________  TRK 10    5     ___  SYSDA
 _  TABLES    MICS.USER.TABLES                    CYL 2     5     20   ________
 _  TEMPPDS   __________________________________  TRK 2     5     10   SYSDA
 _  TEMPFILE  __________________________________  TRK 2     2     ___  SYSDA
 _  TEMPSAS   __________________________________  TRK 2     2     ___  SYSDA
 _  UCATALOG  __________________________________  TRK 2     1     1    ________
 _  UDTFINDX  __________________________________  TRK 5     5     ___  ________
 _  UIXFOUT   __________________________________  CYL 2     1     25   ________
 _  UMICSLOG  __________________________________  TRK 2     2     ___  ________
 _  UMWFPDS   __________________________________  CYL 2     1     25   ________
 _  UMWFSAS   __________________________________  CYL 2     1     ___  ________
 _  UREPORT   __________________________________  TRK 10    5     ___  ________
 _  USER      MICS.USER.DATA                      TRK 10    5     ___  ________
```

```
_  UTILLOG   __________________________________  TRK 2     2     ___  ________
_  UTILMLOG  __________________________________  TRK 2     2     ___  ________
_  WORK      __________________________________  TRK 10    5     ___  ________
```

Use the Data Set Allocation Parameters option to update parameters controlling allocation and cataloging of private (or user level) data sets created by the online facilities. These parameters include the amount of disk space allocated for the data set and the unit/volume specification for data set allocation. You can also modify the data set name assigned to selected data sets.

When you select the Data Set Allocation Parameters option, a scrollable list of data sets is displayed. The most commonly modified data set allocation parameters are displayed on this panel. If you need to change additional parameters, you can select a row from this scrollable display to access a fixed panel containing all of the data set allocation parameters.

Warning: If you change the data set name high level qualifier or the data set name suffix for the user ISPF tables data set (TABLES), current users will be unable to access specifications they saved prior to these parameter changes.

DATA ENTRY INSTRUCTIONS

This panel presents a scrollable list of DASD data sets the online facility may allocate. Each row of the display describes one data set. Descriptions of the display columns follow.

Data Set - Identifies the data set to which the parameters apply. Data Set cannot be changed.

Data Set Name Suffix - Required for permanent MICF user data sets. Must be blank for temporary data sets and for MICF production reporting data sets (PCATALOG, PIXFOUT, PMWFPDS, PMWFSAS, PMICSLOG, and PREPORT). The user data set prefix specified for your data center is appended to this parameter in order to generate the data set name. Some data set parameter entries apply to a generic set of data sets (for example, PRINT applies to the print inquiry output staging data sets) and are allocated with data set names containing a logical version or cycle number. Include one or more asterisks in the data set name suffix for these data sets to indicate where the logical version number should appear.

The next four parameters define the DASD space allocation.

Typ (space units) - Required. BLK, CYL, or TRK. The type of space allocation for this data set. May be abbreviated as B, C, or T.

Prim (primary quantity) - Required. 1-99999. The initial amount of space to be allocated (that is, the number of cylinders, tracks, or blocks).

2nd (secondary quantity) - Optional. 0-99999. The amount of space to be allocated when the space specified in primary quantity has been used. Leave this parameter blank or enter 0 to limit the data set to the primary quantity.

Dir (directory blocks) - Required if data set organization is PO or POU. 1-999. The number of directory blocks to be allocated. Leave this parameter blank if data set organization is PS, PSU, DA, or DAU.

Generic unit - Optional. 1-8 characters. The DASD unit name specification for data set allocation. This parameter should match the unit specification you normally use when allocating data sets. It may be optional at your site, but when specified will usually have a value such as SYSDA. Refer to your data center standards for valid values.

The following three columns can be viewed and/or updated by scrolling right:

Storage Class

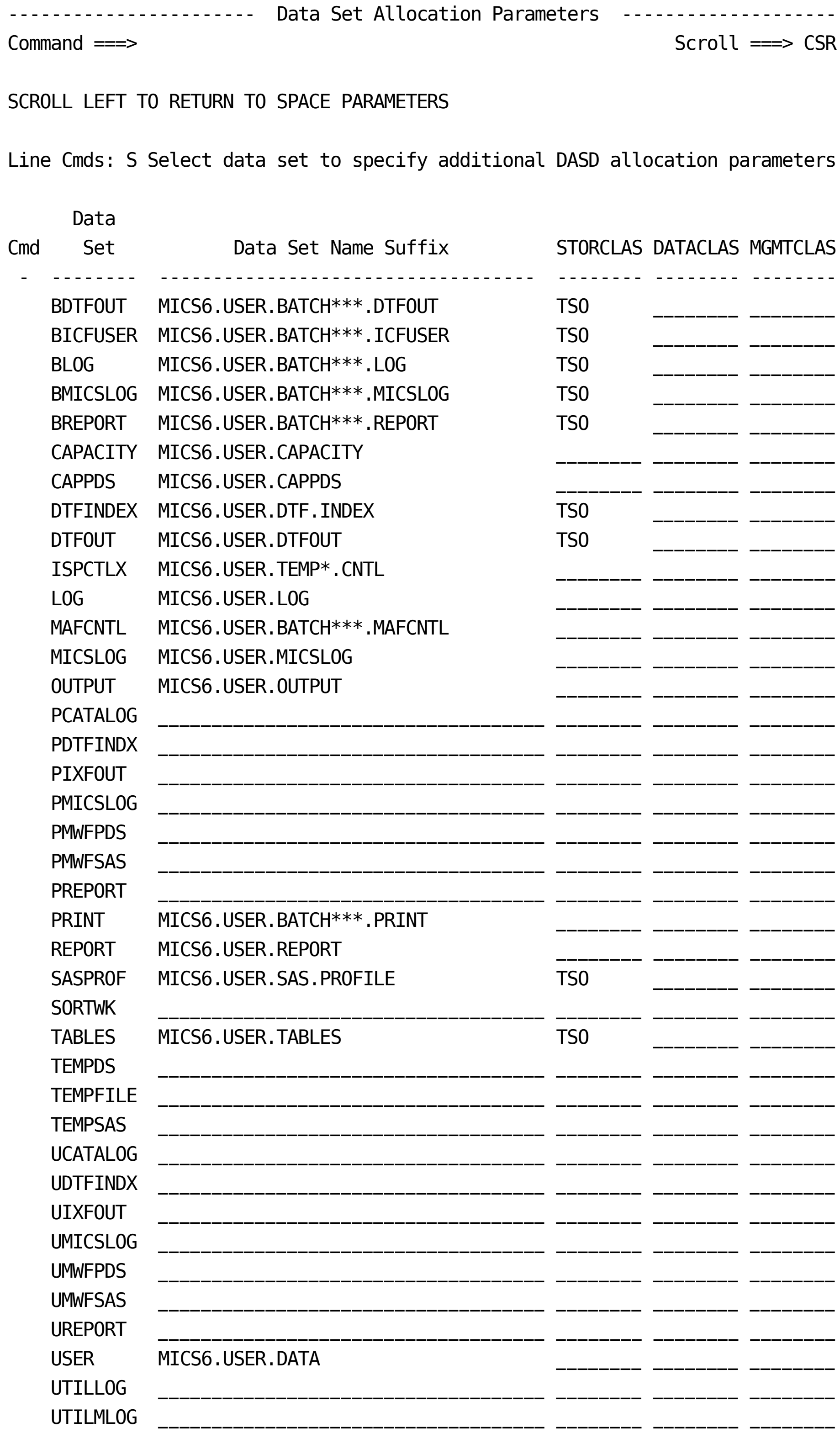
The 1-8 character name specification for Storage Class (STORCLAS) to be used for data set allocation. Refer to your data center standards for valid values.

Data Class

The 1-8 character name specification for Data Class (DATACLAS) to be used for data set allocation. Refer to your data center standards for valid values.

Management Class

The 1-8 character name specification for Management Class (MGMTCLAS) to be used for data set allocation. Refer to your data center standards for valid values.



```
-----------------------  Data Set Allocation Parameters  --------------------
Command ===>                                                 Scroll ===> CSR

SCROLL LEFT TO RETURN TO SPACE PARAMETERS

Line Cmds: S Select data set to specify additional DASD allocation parameters

      Data
Cmd    Set           Data Set Name Suffix          STORCLAS DATACLAS MGMTCLAS
 -  --------  ----------------------------------  -------- -------- --------
    BDTFOUT   MICS6.USER.BATCH***.DTFOUT           TSO      ________ ________
    BICFUSER  MICS6.USER.BATCH***.ICFUSER          TSO      ________ ________
    BLOG      MICS6.USER.BATCH***.LOG              TSO      ________ ________
    BMICSLOG  MICS6.USER.BATCH***.MICSLOG          TSO      ________ ________
    BREPORT   MICS6.USER.BATCH***.REPORT           TSO      ________ ________
    CAPACITY  MICS6.USER.CAPACITY                  ________ ________ ________
    CAPPDS    MICS6.USER.CAPPDS                    ________ ________ ________
    DTFINDEX  MICS6.USER.DTF.INDEX                 TSO      ________ ________
    DTFOUT    MICS6.USER.DTFOUT                    TSO      ________ ________
    ISPCTLX   MICS6.USER.TEMP*.CNTL                ________ ________ ________
    LOG       MICS6.USER.LOG                       ________ ________ ________
    MAFCNTL   MICS6.USER.BATCH***.MAFCNTL          ________ ________ ________
    MICSLOG   MICS6.USER.MICSLOG                   ________ ________ ________
    OUTPUT    MICS6.USER.OUTPUT                    ________ ________ ________
    PCATALOG  ____________________________________ ________ ________ ________
    PDTFINDX  ____________________________________ ________ ________ ________
    PIXFOUT   ____________________________________ ________ ________ ________
    PMICSLOG  ____________________________________ ________ ________ ________
    PMWFPDS   ____________________________________ ________ ________ ________
    PMWFSAS   ____________________________________ ________ ________ ________
    PREPORT   ____________________________________ ________ ________ ________
    PRINT     MICS6.USER.BATCH***.PRINT            ________ ________ ________
    REPORT    MICS6.USER.REPORT                    ________ ________ ________
    SASPROF   MICS6.USER.SAS.PROFILE               TSO      ________ ________
    SORTWK    ____________________________________ ________ ________ ________
    TABLES    MICS6.USER.TABLES                    TSO      ________ ________
    TEMPDS    ____________________________________ ________ ________ ________
    TEMPFILE  ____________________________________ ________ ________ ________
    TEMPSAS   ____________________________________ ________ ________ ________
    UCATALOG  ____________________________________ ________ ________ ________
    UDTFINDX  ____________________________________ ________ ________ ________
    UIXFOUT   ____________________________________ ________ ________ ________
    UMICSLOG  ____________________________________ ________ ________ ________
    UMWFPDS   ____________________________________ ________ ________ ________
    UMWFSAS   ____________________________________ ________ ________ ________
    UREPORT   ____________________________________ ________ ________ ________
    USER      MICS6.USER.DATA                      ________ ________ ________
    UTILLOG   ____________________________________ ________ ________ ________
    UTILMLOG  ____________________________________ ________ ________ ________
    WORK      ____________________________________ _________ _________ _________
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

The panel above is the resulting display from the RIGHT command shown on the previous panel. From this scrollable display you can select a row as shown above using the S line command. This changes your display window to a fixed panel containing all of the data set allocation parameters for a single row from the scrollable display. A sample of the fixed panel is shown below, illustrating that you can still scroll up and down, as well as locate (L) on data set value.

```
----------------------  Data Set Allocation Parameters  -----------------------
Command ===>

Temporary data set:  WORK - SAS Work File
   DSN prefix            : XYZ123
   DSN suffix         ===> ___________________________________
   Space units        ===> TRK             (BLK, TRK, or CYL)
   Primary quantity   ===> 10              (In above units)
   Secondary quantity ===> 5               (In above units)
   Directory blocks   ===> ___             (For partitioned data sets)
   Round block alloc. ===> NO              (YES/NO)
   Block size         ===> 23040           (Maximum block size)
   Record length         : 512              Volume serial      ===> ______
   Data set org.         : PS               Expiration date    ===> ________
   Record format         : FS               Retention period   ===> ______
   Generic unit       ===> ________        (Generic group name or unit address
   Storage Class      ===> ________        (SMS Storage Class)
   Data Class         ===> ________        (SMS Data Class)
   Management Class   ===> ________        (SMS Management Class)
   Additional allocation parameters:
      DCB     ===> ___________________________________________________________
      Other   ===> ___________________________________________________________
-------------------------------------------------------------------------------
```

```
----------------------  Data Set Allocation Parameters  -----------------------
Command ===>

Permanent data set:  TABLES - CA MICS User ISPF Tables DS
   DSN prefix            : XYZ123
   DSN suffix         ===> MICS.USER.TABLES
   Space units        ===> CYL            (BLK, TRK, or CYL)
   Primary quantity   ===> 2              (In above units)
   Secondary quantity ===> 5              (In above units)
   Directory blocks   ===> 20             (For partitioned data sets)
   Round block alloc. ===> NO             (YES/NO)
   Block size         ===> 9040           (Maximum block size)
   Record length         : 80              Volume serial      ===> ______
   Data set org.         : PO              Expiration date    ===> ________
   Record format         : FB              Retention period   ===> ______
   Generic unit       ===> ________       (Generic group name or unit address
   Storage Class      ===> TSO_____       (SMS Storage Class)
   Data Class         ===> ________       (SMS Data Class)
   Management Class   ===> ________       (SMS Management Class)
   Additional allocation parameters:
      DCB     ===> ___________________________________________________________
      Other   ===> ___________________________________________________________
-------------------------------------------------------------------------------
```

Use the Data Set Allocation Parameters panel to modify parameters used in allocating data sets for SAS execution and other online facility processes. This panel displays allocation parameters for one data set. The data set identifier and title are displayed at the top of the panel. You can use the UP and DOWN commands to display data set allocation parameters for other data sets.

DATA ENTRY INSTRUCTIONS

DSN prefix - This field displays the user data set name prefix defined for your data center. DSN prefix cannot be changed.

DSN suffix - Required for permanent MICF user data sets. Must be blank for temporary data sets and for MICF production reporting data sets (PCATALOG, PIXFOUT, PMWFPDS, PMWFSAS, PMICSLOG, and PREPORT). The user data set prefix specified for your data center is appended to this parameter in order to generate the data set name. Some data set parameter entries apply to a generic set of data sets (for example, PRINT applies to the print inquiry output staging data sets) and are allocated with data set names containing a logical version or cycle number. Include one or more asterisks in the data set name suffix for these data sets to indicate where the logical version number should appear.

Space units - Required. BLK, CYL, or TRK. The type of space allocation for this data set. May be abbreviated as B, C, or T.

Primary quantity - Required. 1-99999. The initial amount of space to be allocated (that is, the number of cylinders, tracks, or blocks).

Secondary quantity - Optional. 0-99999. The amount of space to be allocated when the space specified in Primary quantity has been used. Leave this parameter blank or enter 0 to limit the data set to the primary quantity.

Directory blocks - Required if data set organization is PO or POU. 1-999. The number of directory blocks to be allocated. Leave this parameter blank if data set organization is PS, PSU, DA, or DAU.

Round block alloc. - Required. Specify YES (or Y) to round block allocation to cylinders. Specify NO (or N) for block allocation without rounding. Must be NO when Space units is

CYL or TRK.

Block size - Required. 1 to 32760. The size of the physical blocks of records for this data set. If Record format is F, Block size must equal Record length. If Record format is FB or FS, Block size must be divisible by Record length. If Record format is V, Block size must be equal to Record length plus four. If Record format is VB, Block size minus four must be greater than or equal to Record length.

Record length - The length of the records for this data set. If Record format is V, this is the maximum record length. Record length cannot be changed.

Volume serial - Optional. 1-6 character volume serial number. To allocate this data set on a specific DASD volume, specify this parameter. The volume serial number may be optional or defaulted at your site, or in some cases it may be implicitly assigned based on the specification of the unit name.

Data set org. - The type of data set being allocated (for example, PO, PS, DA, etc.). Data set org cannot be changed.

Expiration date - Optional. Data set expiration date in yyddd or yyyy/ddd format. (Note that yyyy/ddd format is supported with MVS/XA and above). Specify this parameter if your data center requires expiration dates for DASD data set allocations.

Record format - The format of records in this data set. Record format cannot be changed.

Retention period - Optional. Must be blank when Expiration date is specified. 1-99999. The number of days the data set should be kept before being deleted by your data center's DASD management routines.

Generic unit - Optional. 1-8 characters. The DASD unit name specification for data set allocation. This parameter should match the unit specification you normally use when allocating data sets. It may be optional at your site, but when specified will usually have a value such as SYSDA. Refer to your data center standards for valid values.

Storage Class - Optional. 1-8 characters. The Storage Class name specification for data set allocation. Refer to your data center standards for valid values.

Data Class - Optional. 1-8 characters. The Data Class name specification for data set allocation. Refer to your data center standards for valid values.

Management Class - Optional. 1-8 characters. The Management Class name specification for data set allocation. Refer to your data center standards for valid values.

Additional allocation parameters - Optional. Additional parameters your data center requires to allocate new DASD data sets. Specify these parameters in JCL statement format, separated by commas. Two data entry fields are provided:

DCB parameters - DCB subparameters (for example, OPTCD=W)

Other parameters - Other data set allocation constructs, for example, VOL=REF=dsname.

4.4.3 Authorization Assignment

```
  -------------------------  Authorization Assignment  ------------  ROW 1 OF 9


 Command ===>                                                  Scroll ===> CSR




  Line Cmds: S Select  I Insert  R Repeat  D Delete

                                                   1 1 1 1 1 1 1 1 1 1 2 2 2
  Cmd  User ID  Comment             1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
   -   -------- -------------------- - - - - - - - - - - - - - - - - - - - - - -
   _   ACCT.    Accounting Staff    Y Y Y Y N N N N N N N N N Y Y Y N N N Y N N
   _   CAP.     Capacity Planners   Y Y Y Y N N N N N N N N N N N N Y N N Y N N
   _   DATAENT. Data Entry Clerks   Y Y N Y N N N N N N N N Y N N Y N N N N N N
   _   HLPUSR.  Help Desk Staff     Y Y Y Y N Y Y N Y N N N N N N N N N N Y N N
   _   MGMT.    Managers            Y Y N N N N N N N N N N N N N N N N Y N N N
   _   OPRUSR.  Operations Staff    Y Y N Y N N Y N Y N N N N N N N N N N Y Y N
   _   SYS.     Systems Pgming Staff Y Y Y Y N Y Y N N N N N N N N N N N N N Y N
   _   SYSUSR1  CA MICS Administrator Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y N N Y Y N
   _   USER.    General Users       Y Y Y N N N N N N N N N N N N N N N N N N N
  ****************************** BOTTOM OF DATA  ********************************



 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Use Authorization Assignment to review and update specifications controlling access to the CA MICS ISPF applications. Authorization Assignment is the application security provided by the CA MICS Workstation Facility (MWF). You can use Authorization Assignment to limit access to each of the MWF applications and also to limit the use of facilities within those applications.

Authorization Assignment does NOT secure the CA MICS database. Authorization Assignment can block access to MICF facilities, but that does NOT prevent database access. You must use your installation's data set security facilities (for example, RACF) to limit access to the CA MICS data sets. See the CA MICS Data Sets section (4.3.10) of this guide for a list of CA MICS data set names and each data set's purpose.

AUTHORIZATION ASSIGNMENT PANEL

The Authorization Assignment panel presents a scrollable display of the MWF Authorization table. Each row of the display specifies the MWF authorization for an individual user or a group of users. You can update individual rows, insert (add) new rows, and/or delete existing rows of this table. Each column of the display is described below.

User ID - Required. Specify a seven-character TSO user ID (or logon ID) to define MWF authorization for a specific TSO user. Specify one to seven characters followed by a period (.) to define MWF authorization for a range of users (that is, specify ABC. to specify authorization for user IDs from ABC through ABC9999). When a user accesses MWF, the authorization table will be searched for an exact match with the user's TSO logon ID. If an exact match is not found, then the table will be scanned for a user ID range that includes this user ID. If such a range is found, then authorization specifications in that row will apply. If no match is found, access to MWF is rejected.

Comment - Optional. One to twenty characters. This is provided for your convenience to include a meaningful label for the authorization assignment. It could be the user's name or a description of the user group defined by the authorization specification. This parameter provides the initial default author specification for MICF inquiry composition.

The remaining columns of the display are the actual authorization specifications. They may be updated directly on this scrollable display or through the fixed data entry panel displayed when an authorization row is selected using the S line command or SELECT primary command. Specify Y (Yes) to allow access to the corresponding application or application function. Specify N (No) to refuse access.

```
USER AUTHORIZATION PANEL


 -----------------------------  User Authorization  ----------------------------

 Command ===>



 Line Cmds: S Select  I Insert  R Repeat  D Delete

                                                      1 1 1 1 1 1 1 1 1 1 2 2 2
 Cmd  User ID  Comment              1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
  -   -------- -------------------- - - - - - - - - - - - - - - - - - - - - - -
  _   ACCT.    Accounting Staff     Y Y Y Y N N N N N N N N N Y Y Y N N N Y N N
  _   CAP.     Capacity Planners    Y Y Y Y N N N N N N N N N N N N Y N N Y N N
  _   DATAENT. Data Entry Clerks    Y Y N Y N N N N N N N N Y N N Y N N N N N N
  _   HLPUSR.  Help Desk Staff      Y Y Y Y N Y Y N Y N N N N N N N N N N Y N N
  _   MGMT.    Managers             Y Y N N N N N N N N N N N N N N N N Y N N N
  _   OPRUSR.  Operations Staff     Y Y N Y N N Y N Y N N N N N N N N N N Y Y N
  _   SYS.     Systems Pgming Staff Y Y Y Y N Y Y N N N N N N N N N N N N N Y N
  _   SYSUSR1 CA MICS Administrator Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y N N Y Y N
  _   USER.    General Users        Y Y Y N N N N N N N N N N N N N N N N N N N
 ****************************** BOTTOM OF DATA  ********************************




.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

The User Authorization panel displays the contents of the selected row from the MWF authorization table. Specify YES (or Y) following an authorization category's prompt to allow access to the corresponding application or application facility. Specify NO (or N) to refuse access.

AUTHORIZATION CATEGORY DEFINITIONS

In order to access MWF and use CA MICS facilities, a user must have READ access to most CA MICS sharedprefix and prefix level data sets. Additional READ and UPDATE authority is required in order to use certain MWF application facilities. Specific access requirements are documented below.

The following authorization categories control access to CA MICS online applications selected from the MWF primary menu.

1 - Documentation Access

Allows the user to select the Documentation Access option to search and browse CA MICS documentation.

2 - MICF

Allows the user to select the CA MICS Information Center Facility (MICF) option and to access the CA MICS database using MICF facilities.

19 - Immediate Reports and Graphics Review

Causes the user to skip directly to the MICF Reports and Graphics option for online review of production reports and other inquiry outputs. This category is intended for those users who access the CA MICS online facilities primarily to review production reports and graphics (for example, managers). Using this option, these "reports only" users can review current production reports with a single panel interaction (that is, they can select CA MICS and immediately see reports).

3 - SAS with CA MICS Libraries

Allows the user to select the SAS With CA MICS Libraries (MSAS) option and to access the CA MICS database and libraries with the SAS Display Manager (interactive SAS).

4 - Management Support Applications

Allows the user to select the Management Support Applications option. Additional authorization is required to access a specific management support product's ISPF application.

5 - CA MICS Administrator

Allows the user to select the CA MICS Administrator Facility (MAF) option and to access all of the CA MICS administrator facilities including Authorization Assignment, Product Support Program (PSP) Online Services, and Operational Status and Tracking. Requires UPDATE authority for most sharedprefix and prefix CA MICS libraries, including the sharedprefix.MICS.ISPTLIB data set.

The following authorization categories control access to MICF

Administration facilities.

6 - Shared Inquiry Catalog

Allows the user to update the shared inquiry catalog. New inquiries can be added to the catalog and existing entries can be deleted. Requires UPDATE authority for the sharedprefix.MICS.ISPTLIB data set.

7 - Shared Inquiry Output Cat

Allows the user to update the shared inquiry output catalog. New reports and color graphics can be added to the catalog and existing entries can be deleted. Requires UPDATE authority for the sharedprefix.MICS.ISPTLIB, sharedprefix.MICS.MWFPDS.DATA, and sharedprefix.MICS.MWFSAS.DATA data sets.

8 - Shared MICF Options

Allows the user to update the shared options specifications. Requires UPDATE authority for the sharedprefix.MICS.ISPTLIB data set.

20 - Production Reporting

Allows the user to access the MICF Production Reporting Administration option. Requires UPDATE authority for the sharedprefix.MICS.ISPTLIB data set so the user can insert new production reporting definitions or change CA MICS production job and/or database unit associations. However, users can update individual production reporting job stream definitions if they are given UPDATE authority for the specific, unique data sets associated with the production reporting job stream.

The following authorization categories relate to the Documentation Access application.

9 - Document Printing

Allows the user to print CA MICS documentation. Requires UPDATE authority for the sharedprefix.MICS.ISPTLIB data set.

10 - Reserved

Reserved for future use.

The following authorization categories relate to the Accounting and Chargeback (ACT) application.

14 - ACT Access

Allows the user to select the Accounting and Chargeback option from the Management Support Applications menu.

15 - ACT Parameter Generations

Allows the user to select the Parameter Specification option from the Accounting and Chargeback primary menu. Requires UPDATE authority for the sharedprefix.MICS.ISPTLIB and the sharedprefix.MICS.TABLES data sets.

16 - ACT Manual Data Entry

Allows the user to select the Application Database Data Entry option from the Accounting and Chargeback primary menu. Requires UPDATE authority for the sharedprefix.MICS.TABLES data set.

The following authorization categories relate to the Capacity Planner (CAP) application.

17 - CAP Shared Catalog Admin.

Allows the user to update the shared capacity planning file catalog. New file definitions can be added to the catalog, existing entries can be deleted, shared files can be updated (executed), and saved forecasts can be replaced. Requires UPDATE authority for the sharedprefix.MICS.ISPTLIB and sharedprefix.MICS.CAPACITY data sets.

18 - Reserved

Reserved for future Capacity Planner use.

The following authorization category relates to the Network Service Reporter (NSR) application.

21 - NSR Administration

Allows the user to select the Network Service Reporter Administration option from the Management Support Applications menu. Requires UPDATE authority for the sharedprefix.MICS.ISPTLIB data set and the prefix CA MICS libraries for CA MICS unit databases in which you have installed the Network Service Reporter.

The following authorization category is reserved for future use.

22 - Reserved

4.4.4 CA MICS Administrator Utilities

```
 ----------------------  CA MICS Administrator Utilities ----------------------


Option ===>




       1 - Compression Candidacy Report
           Identify good candidates for file compression.
       2 - Compression Recommendation Reports
           Recommend space and compression tailoring changes.
       3 - Component ESSENTIAL and DERIVED Options Report
           Show complex, component, and file level settings
           for the ESSENTIAL and DERIVED options.
       4 - Database Modification Report
           Report on all user modifications, additions, and
           deletions.
       5 - Database Health Check Utility
           Provide information on database errors, dead
           cycles, and attributes.
       6 - CA MICS Space Utilization and Reporting
           Provide information on CA MICS database space usage
       T - Tutorial
       X - Exit




 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

```
CA MICS Administrator Utilities help you, the system
administrator, make decisions about the configuration of your
site's environment. Here is how to use the menu:

o Options 1 and 2 let you determine the benefits of
  implementing the SAS System's data compression facility.

o Option 3 helps you identify the database options that are
  in effect for the component.

o Option 4 helps you identify user changes to the database.

o Option 5 provides information regarding database errors,
  dead cycles, and attributes.
```

- Option 6 provides reports to monitor the space used by the CA MICS databases.

- Option 7 helps you identify the inactive data elements for a component.

- Option 8 lets you quickly print a SAS format or informat.

- Option 9 provides a method for you to enter values to be assigned during the creation of the HARPHY file.

- Option 10 helps you identify data elements that have missing values throughout the CA MICS database so you can evaluate them and determine if they should be inactivated.

The CA MICS Administrator Utilities menu options are described in the following sections:

```
 1 - Compression Candidacy Report
 2 - Compression Recommendation Reports
 3 - Component ESSENTIAL and DERIVED Options Report
 4 - Database Modification Report
 5 - Database Health Check Utility
 6 - Space Utilization and Reporting
 7 - Component Inactive Data Elements Report
 8 - SAS Format/Informat Print Utility
 9 - Physical Hardware Parameter Creation
10 - Missing Data Elements Report Utility
```

4.4.4.1 Compression Candidacy Report

```
 Command ===>


-------------------  Compression Candidacy Report  --------------------




Unit Database ID          ===> _

Analyze:  Detail timespan ===> Y  (Y/N)
          Days   timespan ===> Y  (Y/N)
          Weeks  timespan ===> Y  (Y/N)
          Months timespan ===> Y  (Y/N)
          Years  timespan ===> Y  (Y/N)


Limit report to:
  Products          ===> ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
  Information areas ===> ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
  Files             ===> ______  ______  ______  ______  ______




.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

The candidacy report analyzes each CA MICS Database unit and reports the unit's compression status, current space allocation, and estimated DASD space savings.

To produce the Compression Candidacy Report, choose option 1 from the CA MICS Administrator Utilities menu (MWF;5;4;1). The following panel is displayed.

Fill in values, as described below, and press END or use the PF3 key to submit the inquiry.

Unit Database ID   the one-character identifier assigned to the unit on which you want to report. A list of valid values for your site is available by placing a question mark (?) in the field.

Analyze:
Detail timespan
Days timespan
Weeks timespan
Months timespan
Years timespan

Y (Yes) or N (No). Defaults to Y. These fields enable you to control the extent to which you want to report on the effects of implementing compression. For example, if you are not interested in compressing Years timespan data, set its Analyze value to N.

Limit report to:
Products
Information areas
Files

three-character product identifiers, three-character information area identifiers, or six-character file identifiers. These fields enable you to restrict the report to specific products, information areas, or files.

Note: If you use choose to limit the report, information about the products, information areas, and files that are not selected are accounted for in the "Overhead space" category.

For example, if your site heavily uses the CICS Analyzer and you want to implement compression in phases, you might choose to implement compression first for the CICS Analyzer. Specify CIC in the Products field to report on the effects of implementing compression on this product's files.

A portion of the DETAIL timespan section of the report, which is run in batch mode on the first cycle of data in the unit, follows.

```
                             Compression Candidacy Report
                             ----------------------------

CA MICS complex name: MDI.DIA6100-COMPRESSION COMPLEX

Timespan: DETAIL    Libref: PDETAIL                       Date: 26APR00  Page  1

Percent savings limit specified:   0.00 %
Total uncompressed file space:  1377.1679

----------+-----+------+------+-----+-------------------+---------------------------+
          |     | % of | % of |---ESTIMATE all cycles---|--ACTUAL values 1st cycle--|
CCC File  |Match| Total| Total|Cycle| uncmprss|  cmprss | uncmprss|  cmprss |cmprss |
          |Limit| Saved| Space|Count|(M/bytes)|(M/bytes)|(M/bytes)|(M/bytes)|percent|
----------+-----+------+------+-----+---------+---------+---------+---------+-------+
ACT ACTJBJ NOTSUP  0.00   0.01    2     0.1055    0.1055    0.0264    0.0264   0.00
ACT ACTJBP NOTSUP  0.00   0.70    2     9.6240    9.6240    4.0342    4.0342   0.00
ACT ACTJBS NOTSUP  0.00   0.02    2     0.2637    0.2637    0.0264    0.0264   0.00
ACT ACTJDA NOTSUP  0.00   3.46    2    47.6191   47.6191   23.8623   23.8623   0.00
-------------------------------------------------------------------------------------
ACT     4     0    0.00   4.18    8    57.6123   57.6123   27.9492   27.9492   0.00

                                  ... information omitted ...
-------------------------------------------------------------------------------------
SMF BAT_JS NEVER   0.00   0.12    1     1.6875    1.6875    1.6875    1.6875   0.00
SMF BAT_ST YES     7.59  12.83   10   176.7233   72.1934   24.7852   10.1250  59.15
SMF BAT_TP YES     0.00   0.04   10     0.5273    0.5273    0.0527    0.0527   0.00
SMF BAT_TS YES     1.67   3.81   10    52.5000   29.5313    0.8438    0.4746  43.75
SMF BATATP YES     0.00   0.02   10     0.2637    0.2637    0.0264    0.0264   0.00
SMF BATJOB YES     1.27   2.61   10    35.8887   18.4570    1.8457    0.9492  48.57
SMF BATPGM YES     6.55  12.78   10   175.9976   85.7988    4.2188    2.0566  51.25
SMF BATSFA NEVER   0.00   0.00    1     0.0264    0.0264    0.0264    0.0264   0.00
SMF BATSFH NEVER   0.00   0.00    1     0.0264    0.0264    0.0264    0.0264   0.00
SMF BATSFI NEVER   0.00   0.04    1     0.5273    0.5273    0.5273    0.5273   0.00
SMF BATSFJ NEVER   0.00   0.04    1     0.5537    0.5537    0.5537    0.5537   0.00
SMF BATSFP NEVER   0.00   0.01    1     0.1582    0.1582    0.1582    0.1582   0.00
SMF BATSFS NEVER   0.00   1.34    1    18.4043   18.4043   18.4043   18.4043   0.00
SMF BATSFW NEVER   0.00   0.03    1     0.4482    0.4482    0.4482    0.4482   0.00
SMF BATSF4 NEVER   0.00   0.00    1     0.0264    0.0264    0.0264    0.0264   0.00
SMF BATSPL YES     0.00   0.27   10     3.6914    3.6914    0.0264    0.0264   0.00
SMF OPSAVL YES     0.02   0.03    1     0.4219    0.1318    0.4219    0.1318  68.75
SMF OPSCON YES     0.03   0.20    1     2.8213    2.4521    2.8213    2.4521  13.08
SMF OPSOPI YES     0.05   0.08    1     1.1074    0.4746    1.1074    0.4746  57.14
-------------------------------------------------------------------------------------
SMF    19     0   17.17  34.26   82   471.8010  235.3799   58.0078   38.6279  33.41

                                  ... information omitted ...
-------------------------------------------------------------------------------------

PDETAIL

       96     0   36.31 100.00  383  1377.1679  877.1845  435.8760  316.0371  27.49

=====================================================================================
                            Compression Candidacy Report
                            ----------------------------

CA MICS complex name: MDI.DIA6100-COMPRESSION COMPLEX

Timespan: DETAIL    Libref: PDETAIL                       Date: 26APR00  Page  5

Percent savings limit specified:   0.00 %
Total uncompressed file space:  1377.1679

All space values are in M/bytes.

===========================Allocation Summary===========================
|Current Library:                                                      |
|                                                                      |
|  Dataset name: MDI.DIA6100P.DETAIL                                   |
|                                                                      |
|  Total space allocated                    |  1548.7559 |  1548.7559 | (Current Allocation)
|  Uncompressed file space      |   406.5820 |            |            |
|  Compressed file space        |+  524.7334 |            |            |
|  Overhead space               |+    3.9551 |            |            |
|                               |------------|            |            |
|  Minimum space used           |=  935.2705 |-  935.2705 |            |
|                                ============|------------|            |
|  Free space in library                    |=  613.4854 |            |
|  Highest used space                       |-   53.0508 |            |
|                                           |------------|            |
|  Allocated space never used (future use)   |=  560.4346 |            |
========================================================================
```

```
                            Compression Candidacy Report
                            ----------------------------

CA MICS complex name: MDI.DIA6100-COMPRESSION COMPLEX

Timespan: DETAIL    Libref: PDETAIL                       Date: 26APR00  Page  6

Percent savings limit specified:   0.00 %
Total uncompressed file space:  1377.1679

                  +--------+-------+-----------+--------+
                  |Percent | Total | Est. DASD |  Est.  |
                  |Savings | File  |  Savings  | Percent|
                  | Limit  | Count | (M/bytes) | Savings|
                  |--------+-------+-----------+--------|
                  | 7.50   |     1 |  104.5300 |  7.59 %|
                  | 6.50   |     2 |  194.7287 | 14.14 %|
                  | 6.00   |     3 |  278.1201 | 20.20 %|
                  | 2.30   |     4 |  311.0577 | 22.59 %|
                  | 1.90   |     5 |  337.8802 | 24.53 %|
                  | 1.80   |     6 |  363.1927 | 26.37 %|
                  | 1.70   |     7 |  387.2697 | 28.12 %|
                  | 1.60   |     9 |  432.8268 | 31.43 %|
                  | 1.20   |    10 |  450.2584 | 32.69 %|
                  | 0.60   |    11 |  459.4131 | 33.36 %|
                  | 0.50   |    12 |  467.4873 | 33.95 %|
                  | 0.20   |    17 |  484.2451 | 35.16 %|
                  | 0.10   |    20 |  489.5968 | 35.55 %|
                  | 0.00   |    96 |  499.9834 | 36.31 %|
                  +--------+-------+-----------+--------+
```

The sample report shows the amount of space used by each product's files in the DETAIL timespan. You can use it to determine whether or not implementing compression for a given product in a given timespan will enable you to achieve the space savings you desire.

Run this report using a typical day's input data (probably not a holiday or weekend day) because the report uses data from the first cycle it finds.

The report is divided into sections, one for each timespan on which you chose to report. In each timespan section, information is presented by product, with specifics shown for each file. An allocation summary section is provided for each timespan.

The header information in each section is:

CA MICS complex name   the name of the CA MICS complex from which the report is run.

Timespan   the timespan being reported.

Libref   the name of the library that contains the data.

Date   the date on which the report was run.

Percent savings limit specified   this field contains 0.00% to ensure that all files are evaluated as candidates for compression.

Total uncompressed file space   the estimated amount of space, measured in megabytes, that would be consumed if all file cycles were uncompressed.

The fields in each timespan section are:

CCC   the three-character product identifier.

File   the six-character file identifier.

Match Limit   YES, NOTSUP, NEVER, or blank. YES means that the file should be compressed because it matches the savings limit. NOTSUP means that the file does not support compression. NEVER means that the file must not be compressed.

```
                Blank means that the file should not be
                compressed because it does not match the
                savings limit.

 % of Total     the amount of space saved by compressing this
 Saved          file in this timespan, expressed as a
                percent of this timespan's total space used
                by all uncompressed files.

 % of Total     the amount of space consumed by the
 Space          uncompressed file in this timespan,
                expressed as a percent of this timespan's
                total space used by all uncompressed files.

ESTIMATE all cycles

  Cycle Count   the estimated number of cycles of this file
                in this timespan.

  uncmprss      the estimated number of megabytes of space
  (M/bytes)     that would be occupied by all cycle files in
                this timespan if all cycles of the file were
                not compressed.

  cmprss        the estimated number of megabytes of space
  (M/bytes)     that would be occupied by all cycle files in
                this timespan if all cycles of the file were
                compressed.

ACTUAL values 1st cycle

  uncmprss      the actual number of megabytes of space that
  (M/bytes)     would be occupied by the file in this
                timespan if the file was not compressed.

  cmprss        the actual number of megabytes of space that
  (M/bytes)     would be occupied by the file in this
                timespan if the file was compressed.

  cmprss        the difference between the compressed and
  percent       uncompressed sizes.  A negative value means
                that the compressed file is larger than the
                uncompressed file.


In addition to the header information, the fields in each
allocation summary section are:

Current Library
```

Dataset name the name of the data set on which you are reporting.

Total space allocated the actual amount of space currently allocated for this data set.

Uncompressed file space the actual amount of space used by files that are currently uncompressed.

Compressed file space the actual amount of space used by files that are currently compressed.

Overhead space the amount of space used by products, information areas, or files not selected and overhead for directories and the like.

Minimum space used the amount of space used by compressed files, uncompressed files, and overhead.

Free space in library the difference between the current allocation and the minimum space used.

Highest used space the largest amount of space used in the library. (This is space that was used and then freed.)

Allocated space never used (future use) the difference between the amount of free space in the library and the largest amount of space used in the library.

The summary section also presents a table that you can use to help choose the maximum amount of compression you seek to achieve. The columns in the table are:

Percent Savings Limit the timespan savings percentage. A value of 0 means that all files eligible for compression would be compressed, even if those files would produce 0% savings in the timespan. A value of of 1.70 means that only files that save 1.7% or more of the space used by the timespan would be compressed. A value of **** means that the files' sizes increase when they are compressed (negative compression).

Total File Count the number of files that meet or exceed the percent savings limit value.

Est. DASD Savings (M/bytes)

the estimated amount of DASD space that is saved by compressing the files that meet the percent savings limit value.

Est. Percent Savings

the estimated percentage of DASD space that is saved by compressing the files that meet the percent savings limit value. (The percentage is calculated using the uncompressed size of the library.)

Using the sample report's table, if the System Administrator were to set the site's compression limit at 0.10, 20 files would be eligible for compression. To find out which files, browse the "% of Total Saved" column in timespan section of the report and note the files whose values are 0.10 or greater. By compressing those 20 files, the site would save an estimated 35% of its current DASD space allocation for the DETAIL timespan.

If the site chose 0% as its objective, it would need to compress an additional 76 files, but would achieve only 0.7% additional DASD space savings.

4.4.4.2 Compression Recommendation Reports

The compression recommendation reports enable you to maximize your use of compression. The reports are the

```
  1 - Compression Recommendation Report
  2 - Library Allocation Analysis Report
  3 - COMPRESS Parameter Member Update Report
```

and are described in the sections following this introduction.

To produce the Compression Recommendation Reports, choose option 2 from the CA MICS Administrator Utilities menu (MWF;5;4;2). The following panel is displayed:

```
 Command ===>


--------------------  Compression Recommendation Reports  ---------------------




Unit Database ID          ===> _

Analyze:  Detail timespan ===> Y  (Y/N)
          Days   timespan ===> Y  (Y/N)
          Weeks  timespan ===> Y  (Y/N)
          Months timespan ===> Y  (Y/N)
          Years  timespan ===> Y  (Y/N)

Percent saving lower limit ===> _____  (0.00-99.99 percent)


Limit reports to:
  Products          ===> ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
  Information areas ===> ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
  Files             ===> ______  ______  ______  ______  ______



.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Fill in values, as described below, and press END or use the PF3 key to submit the inquiry.

```
Unit Database ID        the one-character identifier assigned
                        to the unit on which you want to
                        report.  A list of valid values for
                        your site is available by placing
                        a question mark (?) in the field.

Analyze:                Y (Yes) or N (No). Defaults to Y.
 Detail timespan        These fields enable you to control
 Days timespan          the extent to which you want to
 Weeks timespan         report on the effects of implementing
 Months timespan        compression.  For example, if you are
 Years timespan         not interested in compressing Years
                        timespan data, set its Analyze value
                        to N.

Percent saving          a number ranging from 0.00 to 99.99.
lower limit             This is the percentage savings that
                        limits which files are chosen for
                        compression.  (A file is selected for
                        compression if it contributes at
                        least this amount of savings toward
                        the total amount of space consumed by
                        the uncompressed timespan files.)
                        Use the information from the
                        Compression Candidacy Report to help
                        you choose a value for this field.

Limit reports to:       three-character product identifiers,
 Products               three-character information area
 Information areas      identifiers, or six-character file
 Files                  identifiers.  These fields enable you
                        to restrict the reports to specific
                        products, information areas, or
                        files.

                        Note:  if you use choose to limit the
                        reports, information about the
                        products, information areas, and
                        files that are not selected are
                        accounted for in the "Overhead space"
                        category.

                        For example, if your site heavily
                        uses the CICS Analyzer and you want
                        to implement compression in phases,
                        you might choose to implement
                        compression first for the CICS
```

```
                                 Analyzer.  Specify CIC in the
                                 Products field to report on the
                                 effects of implementing compression
                                 on this product's files.
```

4.4.4.2.1 Compression Recommendation Report

The Compression Recommendation Report is similar to the Compression Candidacy Report discussed in the prior section. The Recommendation report presents its output by timespan and in three sections:

- the detail for each timespan, including the Percent savings limit specified field, which reflects the percent saving lower limit you entered on the panel, and the Match Limit field, which contains one of the following:

  YES — if the file achieves at least your specified savings limit

  NOTSUP — if the file does not support compression

  NEVER — if the file must not be compressed

  (blank) — if the file does not achieve your specified savings limit

- the allocation summary, including an additional recommended allocation block which is described below

- the table of percent savings limits and their estimated DASD savings

A sample of the allocation summary section of the report follows.

```
                          Compression Recommendation Report
                          ---------------------------------

CA MICS complex name: MDI.DIA6100-COMPRESSION COMPLEX

Timespan: DETAIL    Libref: PDETAIL                       Date: 28APR00  Page  5

Percent savings limit specified:   0.10 %
Total uncompressed file space:  1318.7339

All space values are in M/bytes.

===========================Allocation Summary===========================
|Current Library:                                                      |
|                                                                      |
|  Data set name: MDI.DIA6100P.DETAIL                                  |
|                                                                      |
|  Total space allocated                     |  1548.7559 |  1548.7559 | (Current Allocation)
|  Uncompressed file space      |   360.0967 |            |            |
|  Compressed file space        |+  523.3359 |            |            |
|  Overhead space               |+    3.9551 |            |            |
|                               |------------|            |            |
|  Minimum space used           |=  887.3877 |-  887.3877 |            |
|                                ============|------------|            |
|  Free space in library                     |=  661.3682 |            |
|  Highest used space                        |-  121.9219 |            |
|                                            |------------|            |
|  Allocated space never used (future use)   |=  539.4463 |            |
|                                             ============|            |
|                                                         |            |
|---------------------------------------------------------|            |
|                                                         |            |
|Recommendation:  Compress   21 files.                    |            |
|                                                         |            |
|  Est. uncompressed file space |   172.2568 |            |            |
|  Est. compressed file space   |+  676.6727 |            |            |
|  Same amount overhead space   |+    3.9551 |            |            |
|                               |------------|            |            |
|  Estimated minimum space      |=  852.8846 |   852.8846 |            |
|                                ============|            |            |
|  Same amount free space in library         |+  661.3682 |            |
|                                            |------------|            |
|  Estimated maximum space required          |= 1514.2528 |- 1514.2528 | (New Allocation)
|                                             ============|------------|
|  Estimated allocation saving/increase                   |=   34.5031 | (Saving)
|  Estimated percent change in allocation                 |     2.23 % |
========================================================================
```

The fields in the Current Library section are identical to the fields in the Compression Candidacy Report. The new section, Recommendation, includes the following fields:

Recommendation: Compress nn files
the number of files you should compress to achieve your desired savings. This is the number of files that matched your limit.

Est. uncompressed file space
the estimated amount of space that would be occupied by all of the cycle files whose percent change values were less than your specified value.

Est. compressed file space
the estimated amount of space that would be occupied by all of the cycle files whose percent change values were greater than or equal to your specified value.

Same amount overhead space
the amount of space that is currently consumed for overhead.

Estimated minimum space
the sum of the estimated uncompressed and compressed file space and overhead.

Same amount free space in library
the amount of free space that exists in the current data set's allocation.

Estimated maximum space required
the sum of the estimated maximum space and free space. This is the amount of space to allocate if you choose to allocate new libraries when you implement compression.

Estimated allocation saving/increase
the difference between the current allocation and the new allocation. A negative number denotes a size increase.

Estimated percent change in allocation
the quotient of the new allocation and the current allocation, expressed as a percentage. A negative percentage means that the new allocation is larger than the current allocation.

4.4.4.2.2 Library Allocation Analysis Report

The library allocation report summarizes the amount of space currently allocated by each of the unit's timespan libraries. It also suggests a new database allocation value, and shows the difference between the current amount of space and the new allocation.

You can use this information if you decide to reallocate your libraries.

A sample report follows.

```
                               Library Allocation Analysis Report
                               ----------------------------------

CA MICS Complex Name: MDI.DIA6100-COMPRESSION COMPLEX

Unit Id:  P   Unit Type: PRIMARY                          Date: 28APR00  Page  1

Percent savings limit specified:    0.10 %
-----------+----------+----------+---------+-------+---------+---------------------------------------------
           | Current  |   New    |  Amount |Percent|         |
Library    |Allocation|Allocation|  Saved  | Saved | Results | Data set Name
           |(M/bytes) |(M/bytes) |(M/bytes)|       |         |
-----------+----------+----------+---------+-------+---------+---------------------------------------------
PDETAIL     1548.7559  1514.2528    34.5031   2.23 % Saving    MDI.DIA6100P.DETAIL
-----------------------------------------------------------------------------------------------------------

Unit:  P    1548.7559  1514.2528    34.5031   2.23 % Saving

===========================================================================================================
```

4.4.4.2.3 COMPRESS Parameter Member Update Report

The COMPRESS parameter member update report shows you how to revise the FILE statements in prefix.MICS.PARMS(COMPRESS) to achieve the compression benefits shown on the other Compression Recommendation Reports.

The report shows all of the FILE statements, highlighting those that need to be changed by showing the changed timespan indicator in lowercase text and placing a changed flag to the right of the statement. (Changing the parameter does not implement compression, though. CYCLEGEN, BACKUP, and RESTORE must be successfully completed to make the change, as outlined in Section 6.2.2.4 of the CA MICS System Modification Guide.)

A sample report is shown below:

```
                    COMPRESS Parameter Member Update Report
                    ---------------------------------------

CA MICS Complex Name: MDI.DIA6100-COMPRESSION COMPLEX

Unit Id:  P   Unit Type: PRIMARY                          Date: 28APR00  Page  1

Percent savings limit specified:    0.10 %


 (Use this report to update the FILE statements in the MDI.DIA6100P.PARMS(COMPRESS) member.
  All changes are shown in lower case.)



------------------------------------------------------------
FILE  SMF BAT_JS n Y Y Y Y      (Changed)
FILE  SMF BAT_ST Y Y Y Y Y
FILE  SMF BAT_TP Y Y Y Y Y
FILE  SMF BAT_TS n Y Y Y Y      (Changed)
FILE  SMF BATATP Y Y Y Y Y
FILE  SMF BATJOB n Y Y Y Y      (Changed)
FILE  SMF BATPGM y Y Y Y Y      (Changed)
FILE  SMF BATSFH N N N N N
FILE  SMF BATSPL n Y Y Y Y      (Changed)
FILE  SMF BATWDA Y Y Y Y Y
FILE  SMF OPSAVL Y Y Y Y Y
FILE  SMF OPSCON Y Y Y Y Y
FILE  SMF OPSCTF Y Y Y Y Y
FILE  SMF OPSOPI Y Y Y Y Y
```

4.4.4.3 Component ESSENTIAL and DERIVED Options Report

```
 Command ===>


----------  Component ESSENTIAL and DERIVED Options Report  -----------



Product:  ===> ___



.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

The Component ESSENTIAL and DERIVED Options Report provides a listing of your complex and component level ESSENTIAL and DERIVED options, as well as the options that are in effect for each file for the selected component. These options can be tailored to exclude groups of data elements as a means of controlling the size of your database.

NOTE: This report is not supported for the CA MICS Accounting and Chargeback Option (ACT). The ACT component forces all accounting related files to ESSENTIAL ALL and DERIVED DISK, regardless of customer settings.

The report shows the ESSENTIAL and DERIVED database options:

o The complex level options apply to all components in the database complex and are set in sharedprefix.MICS.PARMS(CPLXDEF). The complex options can be overridden by using component level options.

o The component level options apply only to a component and are set in sharedprefix.MICS.GENLIB(cccGENIN). The COMPOPT statement defines the ESSENTIAL and DERIVED data element options for a component. The component options can be overridden by using file level options.

o The file level options apply to a file within a component and are set in sharedprefix.MICS.GENLIB(cccGENIN) by using the FOPT statement.

Refer to Chapter 2 of the CA MICS Planning, Installation, Operation and Maintenance Guide for more information on complex level database options and Chapter 4 of the CA MICS System Modification Guide for the component and file level options.

To produce the Component ESSENTIAL and DERIVED Options Report, choose option 3 from the CA MICS Administrator Utilities menu (MWF 5;4;3).

Fill in a value, as described below, and press END or use the PF3 key to submit the inquiry.

Product   Required, no default. Enter one of the following three-character (ccc) component identifiers.

```
AUM  AST  BAS  CIC  DB2  DEX  HSM
IDM  IMS  MQS  MTI  PWR  RMF  SMF
SNT  SRL  TDM  TSO  VCA  VMC  WEB
```

A sample report listing for the Component ESSENTIAL and DERIVED Options Report follows.

In the following example, the Component ESSENTIAL and DERIVED Options Report provides a listing of the complex, component, and file level options that are in effect for the DB2 Analyzer.

```
+---------------------------------------------------------------------------------------------------------------------------------+
                                          Component ESSENTIAL and DERIVED Options Report                                 Page 1

  CA MICS Component: DB2                                                            Last Generated: Wed, Dec 10, 2003 at 7:32:28
+---------------------------------------------------------------------------------------------------------------------------------+

  +----------------------------------+                           +-----------------------------+  +-----------------------------+
  | Area: DBP - DB2 Buffer Pool Area |                           |       ESSENTIAL OPTION      |  |        DERIVED OPTION       |
  +----------------------------------+                           +-----------------------------+  +-----------------------------+
                                                                 In                               In
    File     File Label                                          Effect  FOPT    COMPOPT CPLXDEF  Effect  FOPT    COMPOPT CPLXDEF
    ------   -------------------------------------------------   ------- ------- ------- -------  ------- ------- ------- -------
    DBP_DB   DB2 Buffer Pool SuSpend                             ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK
    DBPDBP   DB2 Plan Buffer Pool Activity                       ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK
    DBPDBU   DB2 User Buffer Pool Activity                       ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK
    DBPDSB   DB2 Buffer Pool Activity                            ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK


  +----------------------------------+                           +-----------------------------+  +-----------------------------+
  | Area: DB2 - DB2 Information Area |                           |       ESSENTIAL OPTION      |  |        DERIVED OPTION       |
  +----------------------------------+                           +-----------------------------+  +-----------------------------+
                                                                 In                               In
    File     File Label                                          Effect  FOPT    COMPOPT CPLXDEF  Effect  FOPT    COMPOPT CPLXDEF
    ------   -------------------------------------------------   ------- ------- ------- -------  ------- ------- ------- -------
    DB2_DD   DB2 Data Base Activity Suspend File                 ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK
    DB2_DS   DB2 System Activity Suspend File                    ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK
    DB2_SH   DB2 System DDF Suspend File                         ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK
    DB2DDP   DB2 Plan DDF Activity File                          ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK
    DB2DDU   DB2 User DDF Activity File                          ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK
    DB2DDY   DB2 System DDF Statistics File                      ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK
    DB2DPP   DB2 Package/DBRM Activity File                      ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK
    DB2DSD   DB2 Data Base Activity File                         ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK
    DB2DSP   DB2 Plan Activity File                              ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK
    DB2DSU   DB2 User Activity File                              ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK
    DB2DSY   DB2 System Activity File                            ALL     DEFAULT DEFAULT ALL      DISK    DEFAULT DEFAULT DISK

+---------------------------------------------------------------------------------------------------------------------------------+
```

4.4.4.4 Database Modification Report

```
 Command ===>


----------------------  Database Modification Report  ----------------



 Product:  ===> ___

 Report Selection: Y - Generate Report
                   N - Suppress Report

   ===> _    Product Maintenance Blocks
   ===> _    Product User Maintenance Blocks/Comments
   ===> _    Product Global Statement Modifications
   ===> _    Product Common Data Element Modifications
   ===> _    Product File Level Modifications
             ===> _   Include Data Element Timespan Changes


   ===> _    Complex ($DEGENIN) Common Data Element Modifications

   NOTE: The $DEGENIN report is the only report that does not
         require a specified product.



.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Note: The Database Modification Report generation requires significant temporary WORK and SORT space. If you receive SAS errors due to SORT failure or out-of-space conditions, increase the following MICF resource specifications. From the CA MICS Workstation Facility (MWF) main menu, select:

```
2;0;6 - Data Set Allocation Parameters
        Increase space for the SORTWK and WORK parameters

2;0;3 - MICF Parameter Update
        Increase the Sort Work parameter value to 6
```

The Database Modification Report provides a comprehensive list of user modifications that were made to the CA MICS database. The Database Modification Report is comprised of six reports that can be run individually or in combination with each other. These reports generate listings of additions, deletions, exceptions, and modifications to the database at the product (reports 1 thru 5) and complex levels (report C).

To produce the Database Modification Report, choose option 4 from the CA MICS Administrator Utilities menu (MWF 5;4;4).

Enter the following options as described below and type END or press the PF3 key to begin report generation.

Product

Required for all reports except the Complex ($DEGENIN) Common Data Element Modifications Report, no default. Enter one of the following three-character (ccc) component identifiers.

ACT AUM AST BAS CIC DB2
DEX HSM IDM IMS MQS MTI
PWR RMF SMF SNT SRL TDM
TSO VCA VMC WEB

Product Maintenance Blocks

Required, no default. Enter Y (yes) to include product maintenance block modifications in the report. Enter N (no) to exclude product maintenance block modifications from the report.

Product User Maintenance Blocks/Comments

Required, no default. Enter Y (yes) to include product user maintenance block/comment modifications in the report. Enter N (no) to exclude product user maintenance block/comment modifications from the report.

Product Global Statement Modifications

Required, no default. Enter Y (yes) to include product global statement modifications in the report. Enter N (no) to exclude product global statement modifications from the report.

Product Common Data Element Modifications

Required, no default. Enter Y (yes) to include product common data element modifications in the report. Enter N (no) to exclude product common data element modifications from the report.

Product File Level Modifications
: Required, no default. Enter Y (yes) to include product file level modifications in the report. Enter N (no) to exclude product file level modifications from the report.

Include Data Element Timespan Changes
: Required, no default. Enter Y (yes) to include data element timespan changes in the report. Enter N (no) to exclude data element timespan changes from the report.

Note: This option is valid only when the Product File Level Modifications option is selected.

Complex ($DEGENIN) Common Data Element Modifications
: Required, no default. Enter Y (yes) to include complex common data element modifications in the report. Enter N (no) to exclude complex common data element modifications from the report.

A sample report listing for the Database Modification Report follows.

```
 The Product Maintenance Blocks Report provides a listing of the maintenance blocks for changes that affect database
 content and structure, which are changes made to sharedprefix.MICS.GENLIB(cccGENIN).  In the following example, the
 report shows the product maintenance blocks for the MQSeries Analyzer.

+----------------------------------------------------------------------------------------------------------------------------------+
                                      D A T A B A S E   M O D I F I C A T I O N   R E P O R T
                                                                                                                         Page 1
  CA MICS Component: MQS
  Report: REPORT 1 - Product Maintenance Blocks                                   MQS Last Generated: Mon, Jan 5, 2004  at 14:12:40
+----------------------------------------------------------------------------------------------------------------------------------+

 */* CHANGE | PROBLEM|MMYY|VENDOR CHANGE DESCRIPTION                 */  00003000
 */*--------+--------+----+------------------------------------------*/  00004000
 */*--------+--------+----+------------------------------------------*/  00005000
 */* MQS0000|        |1297|CA MICS MQSeries Initial Release          */  00006000
 */* MQS6005|        |0498|Consolidated Maintenance                  */  00006005
 */* MQS6080|        |1001|Support for MQSeries V5R2                 */  00006080
 */* MQS6150|        |1002|Supp of MQS Task Acct, Queue, CF, & Maint */  00006150
 */* MQS6350|        |0403|Supp for DB2, Report Enhancemnts, & Maint */  00006350
 */* MQS6370|        |1003| Report Enhancements & Consolidated Maint */  00006370
 */*--------+----+---------------------------------------------------*/  00007000
+----------------------------------------------------------------------------------------------------------------------------------+
```

The Product User Maintenance Blocks/Comments Report provides a listing of the maintenance blocks and comments added by a user. In the following example, the report shows a maintenance block entered by a user to record the addition of new data elements to the Task Accounting File for the MQSeries Analyzer.

```
+-----------------------------------------------------------------------------------------------------------------------------------+
                                      D A T A B A S E   M O D I F I C A T I O N   R E P O R T
                                                                                                                            Page 1
 CA MICS Component: MQS
 Report: REPORT 2 - User Maintenance Blocks/Comments                           MQS Last Generated: Mon, Jan 5, yyyy  at 14:12:40
+-----------------------------------------------------------------------------------------------------------------------------------+

*/*        |        |1103| New Data Elements Added to Task Accounting*/ 00009100

+-----------------------------------------------------------------------------------------------------------------------------------+
```

The Product Global Statement Modifications Report provides a listing of the changes made to the COMP, COMPJOB, COMPOPT, DBMDATA, GEN, INPUTSOURCE, and OPTION global statements for a component. In the following example, the report shows that a change was made to remove SMF type 116 records from the INPUTSOURCE statement for the MQSeries Analyzer.

```
+-----------------------------------------------------------------------------------------------------------------------------------+
                                      D A T A B A S E   M O D I F I C A T I O N   R E P O R T
                                                                                                                            Page 1
 CA MICS Component: MQS
 Report: REPORT 3 - Product Global Statement Modifications                     MQS Last Generated: Mon, Jan 5, yyyy  at 14:12:40
+-----------------------------------------------------------------------------------------------------------------------------------+

  Statement                                                                                      Sequence
  Type                   Statement Value                                                         Number
  ------------           ------------------------------------------------------------            --------
  COMP          Master:  MQS 051 32000 VBS ACCT MQS Component                                    00013000
                User:    MQS 051 32000 VBS ACCT MQS Component                      <-No Change  00013000

  COMPJOB       Master:  DWMY EDWM AWM                                                           00016000
                User:    DWMY EDWM AWM                                             <-No Change  00016000

  COMPOPT       Master:  DEFAULT DEFAULT                                                         00014000
                User:    DEFAULT DEFAULT                                           <-No Change  00014000

  DBMDATA1      Master:  30 1440 5 50 6                                                          00017000
                User:    30 1440 5 50 6                                            <-No Change  00017000

  GEN           Master:  GENFILES                                                                00012000
                User:    GENFILES                                                  <-No Change  00012000

  INPUTSOURCE1  Master:  SMF 115 116                                                             00015000
                User:    SMF 116                                                   <-Modified   00015000

  OPTION        Master:                                                                          00018000
                User:                                                              <-No Change  00018000

+-----------------------------------------------------------------------------------------------------------------------------------+
 The Product Common Data Element Modifications Report provides a listing of the changes made to common data elements
 for a component.  Common data elements are present in multiple files within a component.  In the following example,
 the report shows that the label for data element MQSATYP has been changed for the MQSeries Analyzer.

+-----------------------------------------------------------------------------------------------------------------------------------+
                                     D A T A B A S E   M O D I F I C A T I O N   R E P O R T
                                                                                                                          Page 1
  CA MICS Component: MQS
  Report: REPORT 4 - Product Common Data Element Modifications                  MQS Last Generated: Mon, Jan 5, yyyy  at 14:12:40
+-----------------------------------------------------------------------------------------------------------------------------------+

   +---------------------------------------------+
   | MQS Common File Data Element MODIFICATIONS |
   +---------------------------------------------+

     Element   Label
     --------  ---------------------------------------
     MQSATYP   MQ Type of Connection

           Modified              Modification                                                Seq Number  Source
          ------------           -------------------------------------------------------------  ----------  --------
           LABEL        Master:  Type Of Connection                                          00020610    MQSGENIN
                        User:    MQ Type of Connection                                       00020610    MQSGENIN

+-----------------------------------------------------------------------------------------------------------------------------------+
```

The Product File Level Modifications Report provides a listing of the changes made to file options and data element attributes for all of the files in a component. In the following example, a partial report is shown for the MQSeries Analyzer. The data element modification sections of the report show that the label for the common data element MQSATYP has been changed in the Message Manager Accounting (MQAMMA) file and in the Queue Activity (MQAQAA) file. For the MQAQAA file, the overview section of the report shows that the Detail timespan has been turned on (Y) and the cycles have been changed to 03. Because the "Include Data Element Timespan Changes" option was selected, the modification section of the report shows that QAABUFFP has been turned off (N) in the Days, Weeks, Months, and Year timespans.

```
+-----------------------------------------------------------------------------------------------------------------------------+
                                      D A T A B A S E   M O D I F I C A T I O N   R E P O R T
                                                                                                                     Page 1
 CA MICS Component: MQS
 Report: REPORT 5 - Product File Level Modifications                           MQS Last Generated: Mon, Jan 5, yyyy  at 14:12:40
+-----------------------------------------------------------------------------------------------------------------------------+
 AREA: MQA  MQS Accounting Information Area
 FILE: MQAMMA  Message Manager Accounting File

   +----------------------+
   | MQAMMA FILE OVERVIEW |
   +----------------------+
                         DET DAY WKS MON YRS AUD WHS MHS
                         --- --- --- --- --- --- --- ---
     Timespans    Master:  Y   Y   Y   Y   N   N   Y   Y
                  User:    Y   Y   Y   Y   N   N   Y   Y    <-No Change

     CYCLES       Master:  03  10  06  06  00  00  053 024
                  User:    03  10  06  06  00  00  053 024  <-No Change

     COMPRESS     Master:  Y   Y   Y   Y   Y
                  User:    Y   Y   Y   Y   Y                <-No Change

                           Essential  Derived  Type
                           ---------  -------  ----
     FOPT File    Master:  DEFAULT    DEFAULT  USER
      Options     User  :  DEFAULT    DEFAULT  USER         <-No Change

   +---------------------------------------+
   | MQAMMA File Data Element MODIFICATIONS |
   +---------------------------------------+

     Element   Label
     --------  ---------------------------------------
     MQSATYP   MQ Type of Connection

           Modified             Modification                                                  Seq Number  Source
           ------------         ------------------------------------------------------------  ----------  --------
           LABEL        Master:  Type Of Connection                                            00452000    MQSGENIN
                        User:    MQ Type of Connection                                         00452000    MQSGENIN

+-----------------------------------------------------------------------------------------------------------------------------+
+---------------------------------------------------------------------------------------------------------------------------------+
                                      D A T A B A S E   M O D I F I C A T I O N   R E P O R T
                                                                                                                         Page 2
  CA MICS Component: MQS
  Report: REPORT 5 - Product File Level Modifications                           MQS Last Generated: Mon, Jan 5, yyyy  at 14:12:40
+---------------------------------------------------------------------------------------------------------------------------------+
  AREA: MQA  MQS Accounting Information Area
  FILE: MQAQAA  Queue Activity File

   +----------------------+
   | MQAQAA FILE OVERVIEW |
   +----------------------+
                           DET DAY WKS MON YRS AUD WHS MHS
                           --- --- --- --- --- --- --- ---
     Timespans    Master:  N   E   Y   Y   N   N   Y   Y
                  User:    Y   E   Y   Y   N   N   Y   Y    <-Modified

     CYCLES       Master:  00  10  06  06  00  00  053 024
                  User:    03  10  06  06  00  00  053 024  <-Modified

     COMPRESS     Master:  Y   Y   Y   Y   Y
                  User:    Y   Y   Y   Y   Y                <-No Change

                           Essential  Derived  Type
                           ---------  -------  ----
     FOPT File    Master:  DEFAULT    DEFAULT  USER
      Options     User  :  DEFAULT    DEFAULT  USER         <-No Change

   +----------------------------------------+
   | MQAQAA File Data Element MODIFICATIONS |
   +----------------------------------------+

     Element   Label
     --------  ---------------------------------------
     MQSATYP   MQ Type of Connection

           Modified              Modification                                                  Seq Number  Source
          ------------          -------------------------------------------------------------  ----------  --------
           LABEL        Master:  Type Of Connection                                            01415100    MQSGENIN
                        User:    MQ Type of Connection                                         01415100    MQSGENIN

     Element   Label
     --------  ---------------------------------------
     QAABUFFP  Buffer Pool Number

           Modified              Modification                                                  Seq Number  Source
           ------------          -------------------------------------------------------------  ----------  --------
           Timespan     Master:  0 0 0 0 0                                                     01463000    MQSGENIN
                        User:    0 N N N N                                                     01463000    MQSGENIN

+---------------------------------------------------------------------------------------------------------------------------------+
```

The Complex ($DEGENIN) Common Data Element Modifications Report provides a listing of all additions, deletions, exceptions, and modifications made to the CA MICS database at the complex level. In the following example, the report shows that a new data element has been added at the complex level.

```
+-----------------------------------------------------------------------------------------------------------------------------------+
                                      D A T A B A S E   M O D I F I C A T I O N   R E P O R T
                                                                                                                          Page 1
 CA MICS Component: BAS
 Report: REPORT C - Complex ($DEGENIN) Common Data Element Modifications       BAS Last Generated: Wed, Dec 10, yyyy at 7:30:31
+-----------------------------------------------------------------------------------------------------------------------------------+

   +----------------------------------------+
   | $DEGENIN File Data Element ADDITIONS |
   +----------------------------------------+

     Element   Label
     --------  ---------------------------------------
     AVTYAREX  New Data Element                                        NAME/NAMX     TYPE
                                           Data                        Statement   Statement
           Data Element Info        Timespans  Length  Output Format  Seq Number  Seq Number
           -----------------------  ---------  ------  -------------- ----------  ----------
           Essential (NAME)         DETAIL        $8  .               00021200    00021100
           Common Retained (XR)     DAYS          $8  .
           ALIAS: None              WEEKS         $8  .
           ALTNAME: None            MONTHS        $8  .
                                    YEARS         $8  .
+-----------------------------------------------------------------------------------------------------------------------------------+
```

4.4.4.5 Database Health Check Utility

The CA MICS Database Health Check Utility is an administrative tool that scans a CA MICS database for abnormalities. It produces a report that includes the current status of each working database and symptoms of problems. It also provides the ability to delete dead cycles from your database.

The following subsections provide more details concerning the Database Health Check Utility:

1 - Panel Entries
2 - Symptom Descriptions
3 - Sample Output
4 - Database Delete (DBHDELET)
5 - Exclude List

4.4.4.5.1 Panel Entries

```
------------------------  Database Health Check  -----------------------
Command ===>


Unit Database ID          ===> P

Generate Deletes          ===> Y
   Dead Cycles            ===> Y
   Unknown Files          ===> N
   Inactive Files         ===> N

   Delete Window          ===> 120   (minutes)

   Edit Exclude List      ===> N

Press END to continue.
```

To generate the Database Health Check Utility, choose option 5 from the CA MICS Administrator Utilities menu (MWF 5;4;5).

Fill in values, as described below, and press END or use the PF3 key to submit the inquiry.

Unit Database ID

Defaults to the primary database ID. Specify the one-character identifier assigned to the unit to analyze. A list of valid values is displayed by placing a question mark (?) in the field.

Print Duplicate Formats

Defaults to N. Specify Y to have the Database Health Check Utility generate a print of any formats that exist in multiple format catalogs.

Generate Deletes

Defaults to N. Specify Y to have the Database Health Check Utility generate delete control statements. Note that the prefix.MICS.CNTL(DBHDELET) job needs to be run to execute the deletions. See section 4.4.4.5.4 in the PIOM for more information on the Database Health Check Utility Delete job.

Dead Cycles

Defaults to N. Specify Y to have the Database Health Check Utility generate control statements to delete dead cycles (file cycles greater than defined cycles in prefix.MICS.PARMS(DBMODEL)). Note that the prefix.MICS.CNTL(DBHDELET) job needs to be run to execute the deletions.

Unknown Files

Defaults to N. Specify Y to have the Database Health Check Utility generate control statements to delete unknown files. These are members of the SAS data library that are not recognized as CA MICS files, possibly because they were created by the user.

Inactive Files

Defaults to N. Specify Y to have the Database Health Check Utility generate control statements to delete inactive files. Inactive files are defined as files that have been deactivated in a particular timespan or for files belonging to components not active in the unit

Duplicate Formats

Defaults to N. Specify Y to have the Database Health Check Utility generate control statements to delete formats identified as duplicates. Duplicate formats are defined as a format with the same name and entrytype as one which resides in another format catalog. This can be at the complex (MCOLIB) and/or unit (MUOLIB) catalog level and includes the USERFMT1, MICSFMTS and USERFMT2 catalogs.

Note: The Database Health Check Utility does not generate control statements for the delete utility (DBHDELET) for formats that reside in the MICSFMTS format catalog(s). It will only generate control statements for the USERFMT1 and USERFMT2 format catalogs. This is intended as a precaution to avoid inadvertent loss of a CA MICS format.

Delete Window

Defaults to 120. Specifies the amount of time in minutes that prefix.MICS.CNTL(DBHDELET) can run after the Database Health Check Utility has completed. After this window, prefix.MICS.CNTL(DBHDELET) will no longer act on generated control statements.

Edit Exclude List   Defaults to N. Changing this value to Y activates an ISPF Edit session to modify the exclude list. See section 4.4.4.5.5 in the PIOM for more information on the exclude list.

4.4.4.5.2 Symptom Descriptions

The following is a list of possible symptoms that the CA MICS Database Health Check Utility may generate. The list also includes the symptom's description and solution.

Symptom   : Database Format Not at Current SAS Level

Description: Analyzed data library is at a lower level of SAS than the SAS level running on your system.

Action   : Reallocate the library under the current SAS release and copy the contents from the old library to the new.

Symptom   : Data Library Not Optimally Blocked

Description: The blocksize of the SAS data library is not specified for optimal performance. Half-track blocking provides better performance than smaller blocksizes.

Action   : Reallocate data set with half-track blocksize; then use SAS PROC COPY to copy contents. Swap data set names.

Symptom   : Defined Cycles Exceeded (Dead Cycle)

Description: The number of cycles in the database are greater than what is defined in DBMODEL.

Action   : Run the prefix.MICS.CNTL(DBHDELET) job. For more information, see section 4.4.4.5.4.

Symptom   : Unknown File

Description: File is not affiliated with a CA MICS component.

Action   : Check file for applicability and delete if necessary.

Symptom   : Database Update Not Completed

Description: A CA MICS operational job (for example, the DAILY job) has not completed successfully. The

```
             Database Health Check utility requires that all
             updates be complete.

Action     : Run the prefix.MICS.CNTL(RSTATUS) job to
             identify failed process and restart accordingly.


Symptom    : Invalid Cycle

Description: The cycles specified are either non-numeric or
             not valid for a CA MICS file.

Action     : Check file for applicability and delete if
             necessary.


Symptom    : Incorrect File Location

Description: The component that this file is affiliated with
             is not installed in this unit.

Action     : Check file for applicability and delete if
             necessary.


Symptom    : Residual Update Files

Description: Identified files are left over from a CA MICS
             database update process (for example, DAILY).

Action     : Check file for applicability and delete if
             necessary.


Symptom    : File Not Active in Unit

Description: This file is turned off in this unit.

Action     : Turn this file on in
             sharedprefix.MICS.GENLIB(cccGENIN) and run
             sharedprefix.MICS.CNTL(cccCGEN) to activate this
             change.

Symptom    : Incremental Checkpoint File for Undefined
             Component

Description: The incremental checkpoint file is turned on for
             a component not defined in this unit.
```

Action : Check file for applicability and delete if necessary.

Symptom : Component Not Active in Unit for File

Description: The component that this file is affiliated with is not active in this unit.

Action : Check file for applicability and delete if necessary.

Symptom : Dummy File found

Description: This file is defined as a DUMMY file.

Action : This dummy file is needed when a database is first created, but after at least one component file exists in the database, the dummy file may be deleted. Note that the space occupied by this file is insignificant.

Symptom : Incremental Checkpoint File for Inactive Component

Description: The incremental checkpoint file is turned on for an inactive component.

Action : Check file for applicability and delete if necessary.

Symptom : Files Flagged for Deletion

Description: These files are marked for deletion. Y was specified for GENERATE DELETE when the Database Health Check Utility was invoked.

Action : To complete the deletion process, submit prefix.MICS.CNTL(DBHDELET). For more information, see section 4.4.4.5.4.

Symptom : Special ledgers found but Ledger Prorating not active

```
Description: Accounting and Chargeback special ledgers were
             found in the months timespan. However, the
             Ledger Prorating option is not active.

Action     : Check file for applicability and delete if
             necessary.  For more information on proration
             options, see section 4.8.2 in the CA MICS
             Accounting and Chargeback Option User Guide.


Symptom    : Special ledgers found but not requested

Description: Accounting and Chargeback special ledgers were
             found in the months timespan. However, the
             option to use special ledgers is not active.

Action     : Check file for applicability and delete if
             necessary.  For more information on proration
             options, see section 4.8.2 in the CA MICS
             Accounting and Chargeback Option User Guide.


Symptom    : No File Activity. All Cycles have zero
             observation

Description: Every cycle for the named file has zero
             observations.

Action     : This informational message identifies files that
             have no activity. A file is considered inactive
             if all cycles in a timespan have zero
             observations.  Note that this is not necessarily
             an indicator of an error in the CA MICS
             database, but can be a result of one or more of
             the following:

             o A data source not being collected. For
               example, the HARCPU file may show as inactive
               if the RMF type 70 record is not being
               collected.

             o Missing input data.  An example of this is if
               the NODATA option is specified and the input
               data set is empty, or contains data that has
               already been processed.

Symptom    : Format found in multiple catalogs of same
             Library
```

```
Description: A format of the same name and entrytype resides
             in multiple catalogs of the same library.
             Library would either be the MCOLIB or MUOLIB.

Action     : This warning message identifies a possible
             duplicate format condition.  This may or may not
             be significant depending on the format itself
             and the catalog location.

             Most CA MICS processes define the format search
             order as the following:

             USERFMT1 MICSFMTS USERFMT2

             The order of the library (complex versus unit)
             may vary.  For more information about format
             catalog search order, see section 4.5 of the
             System Modification guide.

Symptom    : Format found in both the MCOLIB and MUOLIB

Description: A format of the same name and entrytype resides
             in catalogs at both the complex and unit levels.

Action     : This warning message identifies a possible
             duplicate format condition.  This may
             or may not be significant depending on the
             format itself and the catalog location.

             Most CA MICS processes define the format search
             order as the following:

             USERFMT1 MICSFMTS USERFMT2

             The order of the library (complex versus unit)
             may vary.  For more information about format
             catalog search order, see section 4.5 of the
             System Modification guide.

Symptom    : Format Catalog Not Found

Description: Either the USERFMT1 or USERFMT2 catalog does
             not exist in the format library.

Action     : This warning message identifies a missing
             format catalog.  A typical CA MICS installation
             should have a USERFMT1 MICSFMTS and USERFMT2
             at the complex level (MCOLIB) as well as each
             unit (MUOLIB).
```

4.4.4.5.3 Sample Output

A sample report listing for the Database Modification Report follows:

```
                               CA MICS Database Health Check             14:00 Monday, September 19, yyyy
                                      Symptom Summary
                           Database Title: MICS PRIMARY DATA BASE
                           Database ID and Prefix: P - MDI.U3PUN
            Count    Symptom
        ---------------------------------------------------------------------------
        |      115 | Residual database update file                                |
        +----------+--------------------------------------------------------------+
        |        5 | Dummy file found                                             |
        +----------+--------------------------------------------------------------+
        |        3 | Defined cycles exceeded (dead cycle)                         |
        +----------+--------------------------------------------------------------+
        |        3 | Incorrect File Location                                      |
        +----------+--------------------------------------------------------------+
        |        3 | Files flagged for deletion                                   |
        +----------+--------------------------------------------------------------+
        |        2 | File not active in unit                                      |
        +----------+--------------------------------------------------------------+
        |        8 | Format found in multiple catalogs of same Library            |
        +----------+--------------------------------------------------------------+
        |        1 | Invalid Cycle                                                |
        +----------+--------------------------------------------------------------+
        |        1 | Invalid Incremental Checkpoint dataset                       |
        ---------------------------------------------------------------------------
```

```
                                CA MICS Database Health Check             14:00 Monday, September 19, yyyy
                                       Delete Report
                           Database Title: MICS PRIMARY DATA BASE
                           Database ID and Prefix: P - MDI.U3PUN
                     Member
Timespan  Libname     Name    Delete Reason
--------------------------------------------------------------------------------------------
DETAIL    PDETAIL   BATJOB99  Defined cycles exceeded (dead cycle)
                    BATSFA04  Defined cycles exceeded (dead cycle)
                    HARCPU10  Defined cycles exceeded (dead cycle)
```

```
                                        CA MICS Database Health Check                14:00 Monday, September 19, yyyy
                                                 Symptom Detail
                                     Database Title: MICS PRIMARY DATA BASE
                                     Database ID and Prefix: P - MDI.U3PUN
                       Timespan(DDNAME): DETAIL(PDETAIL); Dataset Name: MDI.U3PUN.DETAIL
Symptom                                             Detail
------------------------------------------------------------------------------------------------------------------
Defined cycles exceeded (dead cycle)                File=BATJOB99 Defined Cycles=10
                                                    File=BATSFA04 Defined Cycles=01
                                                    File=HARCPU10 Defined Cycles=01
Residual database update file                       File=ACTJCS00
                                                    File=ACTJMG00
                                                    File=ACTJRN00
                                                    File=BATJOB00
                                                    File=HARCCU00
                                                    File=HARCFC00
                                                    File=HARCFD00
                                                    File=HARCFH00
                                                    File=HARCFP00
                                                    File=HARCFR00
Invalid Cycle                                       File=BATJOBQ0
File not active in unit                             File=AUMTTA00
Invalid Incremental Checkpoint dataset              File=RMFCKPTX
Dummy file found                                    File=DUMMY
Residual database update file                       File=HARCCU00
                                                    File=HARCFC00
                                                    File=HARCFD00
                                                    File=HARCFH00
                                                    File=HARCFP00
Incorrect File Location                             File=ADMEXC01 should be in PDAYS1
                                                    File=ADMEXC02 should be in PDAYS1
                                                    File=ADMIHL01 should be in PDAYS1
Residual database update file                       File=HARCCUW0
                                                    File=HARCFPW0
                                                    File=HARCFRW0
                                                    File=HARCPUW0
Dummy file found                                    File=DUMMY
Residual database update file                       File=BAT_TPM0
                                                    File=BATATPM0
                                                    File=BATRENM0
                                                    File=IMS_ISM0
                                                    File=PWRPDAM0
                                                    File=PWRPRAM0
File not active in unit                             File=ACTRSL01
Dummy file found                                    File=DUMMY
```

```
                                         CA MICS Database Health Check                14:00 Monday, September 19, yyyy
                                                 Symptom Detail
                                     Database Title: MICS PRIMARY DATA BASE
                                     Database ID and Prefix: P - MDI.U3PUN
                       Timespan(DDNAME): OTHER(MCOLIB)
Symptom                                             Detail
------------------------------------------------------------------------------------------------------------------
Format found in multiple catalogs of same Library   Format=CPLXSID Catalogs=MICSFMTS and USERFMT1
                                                    Format=CPLXSIN Catalogs=MICSFMTS and USERFMT1
                                                    Format=CPLXZNM Catalogs=MICSFMTS and USERFMT1
                                                    Format=DEVTYPS Catalogs=MICSFMTS and USERFMT1
                                                    Format=DSORG Catalogs=MICSFMTS and USERFMT1
                                                    Format=MONEY Catalogs=MICSFMTS and USERFMT1
                                                    Format=TAW Catalogs=USERFMT1 and USERFMT2
                                                    Format=YR Catalogs=MICSFMTS and USERFMT1
```

4.4.4.5.4 Database Delete (DBHDELET)

The Database Health Check Utility provides options to delete files in the CA MICS database that have been identified as "dead cycles," "unknown," or "inactive" as well as the option to delete "duplicate formats."

To delete these files or formats identified by the Database Health Check Utility, perform the following:

Submit 'prefix.MICS.CNTL(DBHDELET)'

After the job completes, check the output to ensure that the job completed successfully. Review the MICSLOG output file for any error messages.

Note: DBHDELET MUST run within the Delete Window specified at the time the CA MICS Database Health Check utility was run. NO changes should be made to the database after the Database Health Check Utility is run and before the DBHDELET is run. If either of the above conditions exists, DBHDELET will abend with a U0998 error.

The LIST option allows DBHDELET to produce all reports but will not physically delete files. This mode can be used in conjunction with the exclude list to review exactly which files will be deleted.

LIST is specified on the SYSPARM keyword of the EXEC statement of the DBHDELET job before submission. For example:

```
// ... JOB ...
//DBHDELET EXEC MICSDBx,SYSPARM='LIST'
// ....
```

A sample report listing for the deleted files follows:

```
                                         CA MICS Database Health Check              10:50 Tuesday, September 20, yyyy
                                                 Delete Report

                                           LIBNAME     MEMBER
                                          --------    --------

                                           DETAIL     BATJOB99
                                           DETAIL     BATSFA04
                                           DETAIL     HARCPU10
```

4.4.4.5.5 Exclude List

When the Generate Deletes option is specified for execution, there may be files or formats marked for deletion that the data center may not want to delete. These may be user-generated files and formats or files being saved for archival purposes.

To address this issue, an exclude list has been provided. Files listed in the exclude list will not be deleted when the DBHDELET program is executed.

The exclude list resides in prefix.MICS.PARMS(DBHEXCL) and can be edited directly or via the Database Health Check Utility options panel.

The following statements are supported:

```
DDNAME ddname member...
TIMESPAN timespan member...
FORMAT libname catalog format...
```

where:

DDNAME   Excludes members in the specified DDNAME.

TIMESPAN   Excludes members in all DDNAMES of a split timespan. For example, specifying timespan DETAIL with exclude files in DETAIL, DETAIL1, and so on.

FORMAT   Excludes formats. Note that formats in the MICSFMTS catalog are not eligible for deletion, therefore excludes do not need to be entered for the MICSFMTS catalogs.

ddname   Is the DDNAME containing the members to be excluded.

timespan   Is the timespan containing the members to be excluded. This value can be either DETAIL, DAYS, WEEKS, MONTHS, or YEARS.

member   Is the member name to be excluded from deletion. Members can be either specific or generic. Generic members end in a colon (:). For example, BATJOB: will exclude all members starting with BATJOB.

Comments start with an * in column 1.

Note: No validation of the syntax is done during edit.

The following are example exclude list statements:

```
DDNAME DAYS BATJOB:
TIMESPAN WEEKS ACTLGR01
```

4.4.4.6 CA MICS Space Utilization and Reporting

CA MICS Space Utilization and Reporting provides the ability to report on CA MICS database usage down to the individual file level. When using standard z/OS space reporting tools against the CA MICS database, only the space used by a timespan can be reported. With CA MICS Space Utilization and Reporting, it is possible to determine the space used by individual CA MICS files and the free space available within the SAS library.

Information provided by CA MICS Space Utilization and Reporting can be used to fine-tune file cycle definitions, determine where to split a database, and estimate the maximum size of a CA MICS database.

The following sections provide more details concerning CA MICS Space Utilization and Reporting:

1 - CA MICS Database Utilization Facility
2 - CA MICS Database Estimator

4.4.4.6.1 CA MICS Database Utilization Facility

CA MICS Database Utilization Facility provides comprehensive information on how much space is used in a CA MICS database.

The CA MICS Database Utilization Facility contains the following reports:

o The Invalid File Report provides a list of files that cannot be identified as belonging to the CA MICS database structure. Invalid files can be the result of residual files from an incomplete database update, user-created files, and obsolete files.

  Invalid files will be reported in the 'Unknown' category of the Information Area and File summary reports. The CA MICS Database Health Check Utility can be run to help resolve issues with identified invalid files.

o The Libname Summary Report provides overview information for each timespan in the database.

o The Information Area Summary Report shows space utilization for each Information Area within a timespan. This report is useful for determining where a database can be split.

o The File Summary Report shows space utilization for each file in a timespan. Note that all cycles for a file are comprised in a single line entry.

The following subsections provide more details concerning the Database Space Utilization Reports:

1 - Panel Entries
2 - Sample Output

4.4.4.6.1.1 Panel Entries

```
 Command ===>

------------------  CA MICS Space Utilization and Reporting ---------------



       1 - Database Space Utilization Report
           Report on Database Space Usage.
       2 - Database Space Estimator Report
           Estimate Maximum Database Size.
       T - Tutorial
       X - Exit
```

```
 Command ===>


-------------------------  Database Space Utilization -----------------



Unit Database ID          ===> _

Space Reporting Unit      ===> L (Bytes, bLocks, Tracks, Cylinders)

Report Types

Summary                  ===> Y
Information Area         ===> Y
Detail                   ===> Y

Press END to continue.


 .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

To generate the Database Space Utilization reports, select option 6 from the CA MICS Administrator Utilities menu, and then option 1 (MWF 5;4;6;1).

Fill in values, as described below, and press END or use the PF3 key to submit the inquiry.

```
Unit Database ID        Defaults to the primary database ID.
                        Specify the one-character identifier
                        assigned to the unit that you want to
                        analyze.  A list of valid values for
                        your site is available by placing a
                        question mark (?) in the field.

Space Reporting Unit    Defaults to L (bLocks).  This report
                        can display space in either Bytes
                        (B), bLocks (L), Tracks (T), or
                        Cylinders (C).

Report Types
------------
 Summary                Defaults to Y (Yes).  Specify Y (Yes)
                        to produce the Libname Summary
                        Report.

 Information Area       Defaults to Y (Yes).  Specify Y (Yes)
                        to produce the Information Area
                        Summary Report.

 Detail                 Defaults to Y (Yes).  Specify Y (Yes)
                        to produce the File Summary Report.
```

4.4.4.6.1.2 Sample Output

A sample report listing for the CA MICS Database Utilization Facility follows:

The Invalid File report lists files within the CA MICS database that cannot be identified as belonging to a CA MICS component.

```
                         CA MICS Space Utilization and Reporting           14:41 Friday, January 27, yyyy   1
                                Database Space Utilization
                                      Invalid Files

                      The following files are considered invalid
                    and may skew results. It is recommended that the
                     CA MICS Database Health Check Utility be run.

                           TimeSpan     DDName      Member
                          ----------  ----------  ----------
                           MONTHS      PMONTHS     BAT_TPM0
                           MONTHS      PMONTHS     BATATPM0
                           MONTHS      PMONTHS     BATRENM0
                           MONTHS      PMONTHS     BATSPLM0
                           MONTHS      PMONTHS     IMS_IAM0
                           MONTHS      PMONTHS     IMS_ISM0
                           DETAIL      PDETAIL     BATJOBQ0
                           DETAIL      PDETAIL     BATJOBYY
                           DETAIL      PDETAIL     RMFCKPTX
                           DETAIL      PDETAIL     XXXJIB01
                           DETAIL      PDETAIL     XXXJOBYY
                           DETAIL      PDETAIL     XXXJOB01
                           DETAIL      PDETAIL     XXXYYY
                           DETAIL      PDETAIL     XXXYYY00
                           DETAIL      PDETAIL     XXXYYY01
```

```
Column Descriptions
-------------------

Timespan: The CA MICS timespan where the invalid file
          resides.

DDname: The DDname of the invalid file.

Member: The SAS member name of the invalid file.
```

The Libname Summary report lists space utilizations for each physical data set of the CA MICS database.

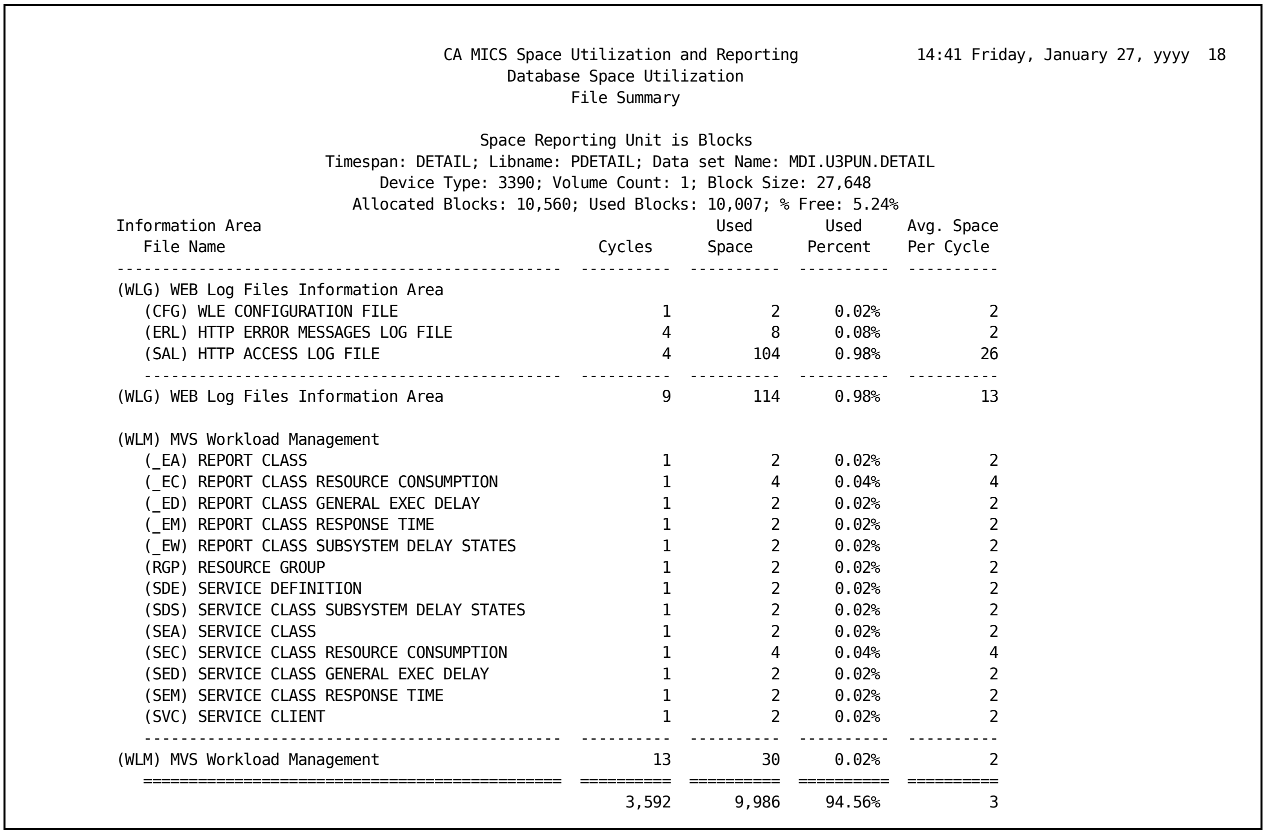
```
                                             CA MICS Space Utilization and Reporting           14:41 Friday, January 27, yyyy   2
                                                    Database Space Utilization
                                                          Libname Summary

                                                    Space Reporting Unit is Blocks
                                                                                  Used     Allocated      Free        Free
Libname   Data Set Name                                Block Size   Members      Space       Space        Space      Percent
--------  -------------------------------------------  ----------  ----------  ----------  ----------  ----------  ----------
PDETAIL   MDI.U3PUN.DETAIL                                 27,648         520      10,007      10,560         574       5.44%
PDAYS     MDI.U3PUN.DAYS                                   27,648         628       3,130       3,130          13       0.42%
PDAYS1    MDI.U3PUN.DAYS1                                  27,648           3          14      12,171      12,168      99.98%
PWEEKS    MDI.U3PUN.WEEKS                                  27,648          87         543         920         382      41.52%
PMONTHS   MDI.U3PUN.MONTHS                                 27,648         207         998       1,176         185      15.73%
PYEARS    MDI.U3PUN.YEARS                                  27,648           1           4           6           5      83.33%
                                                                               ----------  ----------  ----------  ----------
                                                                                   14,696      27,963      13,327      47.66%
```

```
Column Descriptions
-------------------

 Libname: The SAS libname of the timespan.

 Data Set name: The z/OS data set name of the timespan.

 Block size: The physical block size of the data set.

 Members: Total members in the SAS data sets. This includes
   valid and invalid (as listed in the Invalid Files report)
   file entries.

 Used Space: Space currently used in the timespan.

 Allocated Space: Total space currently allocated to the
    timespan. This is computed as Used and Free space.

 Free Space: The space available in the timespan.

 Free Percent: Percentage of allocated free space in the
    timespan.

 The Information Area Summary report shows space utilization for each Information Area
within the timespan.
```

```
                                      CA MICS Space Utilization and Reporting           14:41 Friday, January 27, yyyy   3
                                             Database Space Utilization
                                              Information Area Summary

                                          Space Reporting Unit is Blocks
                            Timespan: DETAIL; Libname: PDETAIL; Data set Name: MDI.U3PUN.DETAIL
                                  Device Type: 3390; Volume Count: 1; Block Size: 27,648
                               Allocated Blocks: 10,560; Used Blocks: 10,007; % Free: 5.24%

                                                                          Used       Used     Avg Space
            Information Area                                  Members     Space     Percent    per File
            ----------------------------------------------  ----------  ----------  ----------  ----------
            (???) Unknown                                            7          12       0.11%           2
            (ACT) Accounting Component                               6          10       0.09%           2
            (ADM) Administrative Activity Information Area           2           2       0.02%           1
            (AST) ASTEX                                              4           7       0.07%           2
            (AUM) MeasureWare Information Area                      18          79       0.75%           4
            (BAT) Batch Activity Information Area                   50       1,480      14.02%          30
            (CIC) CICS Activity Information Area                    11          65       0.62%           6
            (DBP) DB2 Buffer Pool Area                               2           3       0.03%           2
            (DB2) DB2 Information Area                              15          78       0.74%           5
            (DEA) VAX/VMS Accounting Information Area                6          14       0.13%           2
            (DEM) VAX/VMS Monitor Information Area                   6         145       1.37%          24
            (DES) VAX/VMS System Usage Information Area              5          39       0.37%           8
            (HAR) Hardware Utilization                              22         360       3.41%          16
            (HFS) USS File System Information Area                   3           9       0.09%           3
            (HSB) DFHSM Inventory Backup Information Area            4       2,777      26.30%         694
            (HSM) DFHSM Inventory Primary Information Area          10         835       7.91%          84
            (IDM) IDMS Information Area                              5          15       0.14%           3
            (IMS) IMS Activity Information Area                      7          46       0.44%           7
            (MQA) MQS Accounting Information Area                    4          16       0.15%           4
            (MQS) MQS Performance Information Area                  14          27       0.26%           2
            (NVS) Netview Session Information Area                   9          16       0.15%           2
            (OPS) Operations Information Area                        3           6       0.06%           2
            (PWR) VSE/POWER Information Area                        19          44       0.42%           2
            (SCP) MVS System Control Program                        36         625       5.92%          17
            (SNT) Network Information Area                          33          66       0.63%           2
            (SRL) System Reliability Information                    52         452       4.28%           9
            (TCP) TCP/IP Information Area                           15          36       0.34%           2
            (TDM) TANDEM Information Area                           23         165       1.56%           7
            (TSO) TSO Information Area                              23         333       3.15%          14
            (VCA) VTOC/Catalog Activity Information Area             7          23       0.22%           3
            (VMC) Accounting and Exception Information Are          12          23       0.22%           2
            (VMX) VM/XA and VM/ESA System Information Area          37       1,973      18.68%          53
            (WEB) WEB Information Area                              28          61       0.58%           2
            (WLG) WEB Log Files Information Area                     9         114       1.08%          13
            (WLM) MVS Workload Management                           13          30       0.28%           2
                                                            ----------  ----------  ----------  ----------
                                                                   520       9,986      94.56%          19
```

```
Heading Descriptions
------- ------------

 Timespan: The CA MICS database timespan.

 Libname: The DDname used to reference the timespan data set.

 Data Set Name: The fully qualified data set name of the
  timespan.

 Device Type: The device type of the data set. Note that the
  device type is the same for all volumes of a multi-volume
  data set.

 Volume Count: Number of volumes of the timespan data set.

 Block Size: The z/OS block size of the data set.

 % Free: Percentage of free space in the timespan data set.

 For each of the following items, units can be Bytes, Blocks,
  Tracks, or Cylinders depending upon options selected when
  the report is run.

 Allocated units: The amount of space currently allocated to
  timespan data set.

 Used units: The amount of space currently used in the
  timespan data set.

Column Descriptions
-------------------

 Information Area: Name the CA MICS Information Area with the
    timespan.

 Members: Total members in this Information Area. This
    includes valid and invalid (as listed in the Invalid
    Files report) file entries.

 Used Space: Space currently used in the Information Area.

 Used Percent: Percentage of allocated space used by the
    Information Area within the timespan.

 Avg. Space per File: The average space used for each file
    within the timespan.
```

The File Summary report shows space utilization for each CA MICS File within the timespan.



```
                                        CA MICS Space Utilization and Reporting            14:41 Friday, January 27, yyyy  18
                                               Database Space Utilization
                                                      File Summary

                                              Space Reporting Unit is Blocks
                             Timespan: DETAIL; Libname: PDETAIL; Data set Name: MDI.U3PUN.DETAIL
                                   Device Type: 3390; Volume Count: 1; Block Size: 27,648
                                Allocated Blocks: 10,560; Used Blocks: 10,007; % Free: 5.24%
Information Area                                                   Used        Used     Avg. Space
   File Name                                        Cycles        Space      Percent    Per Cycle
-------------------------------------------------  ----------  ----------  ----------  ----------
(WLG) WEB Log Files Information Area
   (CFG) WLE CONFIGURATION FILE                              1           2      0.02%           2
   (ERL) HTTP ERROR MESSAGES LOG FILE                        4           8      0.08%           2
   (SAL) HTTP ACCESS LOG FILE                                4         104      0.98%          26
   ----------------------------------------------  ----------  ----------  ----------  ----------
(WLG) WEB Log Files Information Area                         9         114      0.98%          13

(WLM) MVS Workload Management
   (_EA) REPORT CLASS                                        1           2      0.02%           2
   (_EC) REPORT CLASS RESOURCE CONSUMPTION                   1           4      0.04%           4
   (_ED) REPORT CLASS GENERAL EXEC DELAY                     1           2      0.02%           2
   (_EM) REPORT CLASS RESPONSE TIME                          1           2      0.02%           2
   (_EW) REPORT CLASS SUBSYSTEM DELAY STATES                 1           2      0.02%           2
   (RGP) RESOURCE GROUP                                      1           2      0.02%           2
   (SDE) SERVICE DEFINITION                                  1           2      0.02%           2
   (SDS) SERVICE CLASS SUBSYSTEM DELAY STATES                1           2      0.02%           2
   (SEA) SERVICE CLASS                                       1           2      0.02%           2
   (SEC) SERVICE CLASS RESOURCE CONSUMPTION                  1           4      0.04%           4
   (SED) SERVICE CLASS GENERAL EXEC DELAY                    1           2      0.02%           2
   (SEM) SERVICE CLASS RESPONSE TIME                         1           2      0.02%           2
   (SVC) SERVICE CLIENT                                      1           2      0.02%           2
   ----------------------------------------------  ----------  ----------  ----------  ----------
(WLM) MVS Workload Management                               13          30      0.02%           2
   ==============================================  ==========  ==========  ==========  ==========
                                                         3,592       9,986     94.56%           3
```

```
Heading Descriptions
------- -------------

 Timespan: The CA MICS database timespan.

 Libname: The DDname used to reference the timespan data set.

 Data Set Name: The fully qualified data set name of the
 timespan.

 Device Type: The device type of the data set. Note that the
    device type is the same for all volumes of a multi-volume
    data set.

 Volume Count: Number of volumes of the timespan data set.

 Block Size: The z/OS block size of the data set.

 % Free: Percentage of free space in the timespan data set.

 For each of the following items, units can be Bytes, Blocks,
    Tracks, or Cylinders depending upon options selected when
    the report is run.

 Allocated units: The amount of space currently allocated to
    timespan data set.

 Used units: The amount of space currently used in the
    timespan data set.

Column Descriptions
-------------------

 Information Area: Name of the CA MICS Information Area
    within the timespan.

 File Name: Name of the CA MICS file within the timespan.

 Cycles: Total iterations (cycles) of this file within the
    timespan.

 Used Space: Space currently used by the file.  Note that the
    space used at the Libname summarization level may not
    equal the allocated space in the timespan.  This
    difference is attributed to internal SAS usage.

 Used Percent: Percentage of allocated space used by the File
    within the timespan.
```

```
Avg. Space per Cycle: The average space used for each cycle
   of the CA MICS file within the timespan.
```

4.4.4.6.2 CA MICS Database Estimator

The Database Estimator is used to estimate the maximum size of a CA MICS database. Space can be reported in bytes, blocks, tracks, or cylinders simply by changing the execution panel option.

It is important to understand that this report is not suitable for modeling or 'what-if' analysis. Modeling implies that space estimates can be made before the first update. Therefore, the Database Space Estimator cannot predict space that will be used by a component before it is installed into a unit or space for moving components from one unit to another.

Since the Database Space Estimator bases its estimate on existing files in the CA MICS database and from information in prefix.MICS.PARMS(DBMODEL), it can be used for:

o Estimating the maximum size of the CA MICS database in as few as a single database update;

o Monitoring space increases or decreases following a configuration change (e.g. DBMODEL);

o Tuning database size and cycling definitions.

The Database Space Estimator also reports on the estimated maximum size of each physical data set comprising the CA MICS database. This is obtained by adding to the allocated space the maximum size of each remaining secondary extent. Note that this value does not take into account the availability of secondary extents or multiple volumes.

The Database Space Estimator contains the following reports:

o The Invalid File Report provides a list of files that cannot be identified as belonging to the CA MICS database structure. Invalid files can be the result of residual files from an incomplete database update, user created files, and obsolete files.

  Invalid files will be reported in the 'Unknown' category of the Information Area and File Summary reports. The CA MICS Database Health Check Utility can be run to help resolve issues with identified invalid files. For more information about the CA MICS Database Health Check Utility, see section 4.4.4.5

- o The Libname Summary provides overview information for each timespan in the database.

- o The Information Area Summary Report shows space utilization for each Information Area within a timespan. This report is useful for determining where a database can be split.

- o The File Summary Reports show space utilization for each file in a timespan. Note that all cycles for a file are comprised in a single line entry.

The following subsections provide more details concerning the CA MICS Database Space Estimator Reports:

1 - Panel Entries
2 - Sample Output

4.4.4.6.2.1 Panel Entries

```
 Command ===>


-------------------------  Database Space Estimator -----------------



Unit Database ID          ===> _

Space Reporting Unit      ===> L (Bytes, bLocks, Tracks, Cylinders)

Report Types

Summary                  ===> Y
Information Area         ===> Y
Detail                   ===> Y

Press END to continue.
```

. . . .

To generate the Database Space Estimator reports, select option 6 from the CA MICS Administrator Utilities menu, and then option 1 (MWF 5;4;6;1).

Fill in values, as described below, and press END or use the

```
PF3 key to submit the inquiry.

Unit Database ID        Defaults to the primary database ID.
                        Specify the one-character identifier
                        assigned to the unit that you want to
                        analyze.  A list of valid values for
                        your site is available by placing a
                        question mark (?) in the field.

Space Reporting Unit    Defaults to L (bLocks).  This report
                        can display space in either Bytes
                        (B), bLocks (L), Tracks (T), or
                        Cylinders (C).

Report Types
------------

 Summary                Defaults to Y (Yes).  Specify Y (Yes)
                        to produce the Libname Summary
                        Report.

 Information Area       Defaults to Y (Yes).  Specify Y (Yes)
                        to produce the Information Area
                        Summary Report.

 Detail                 Defaults to Y (Yes).  Specify Y (Yes)
                        to produce the File Summary Report.
```

4.4.4.6.2.2 Sample Output

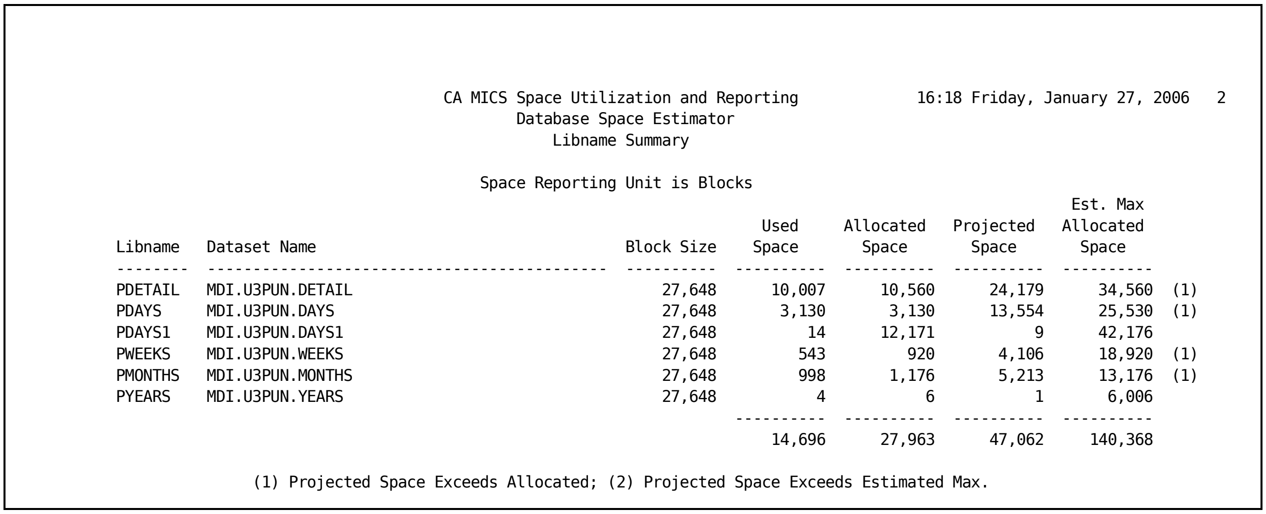
A sample report listing for the Database Space Estimator follows:

The Invalid File report lists files within the CA MICS database that cannot be identified as belonging to a CA MICS component.

```
                                   CA MICS Space Utilization and Reporting            16:18 Friday, January 27, 2006   1
                                           Database Space Estimator
                                                Invalid Files

                                  The following files are considered invalid
                                and may skew results. It is recommended that the
                                 CA MICS Database Health Check Utility be run.

                                     TimeSpan     DDName      Member
                                    ----------  ----------  ----------
                                     MONTHS      PMONTHS     BAT_TPM0
                                     MONTHS      PMONTHS     BATATPM0
                                     MONTHS      PMONTHS     BATRENM0
                                     MONTHS      PMONTHS     BATSPLM0
                                     MONTHS      PMONTHS     IMS_IAM0
                                     MONTHS      PMONTHS     IMS_ISM0
                                     DETAIL      PDETAIL     BATJOBQ0
                                     DETAIL      PDETAIL     BATJOBYY
                                     DETAIL      PDETAIL     RMFCKPTX
                                     DETAIL      PDETAIL     XXXJIB01
                                     DETAIL      PDETAIL     XXXJOBYY
                                     DETAIL      PDETAIL     XXXJOB01
                                     DETAIL      PDETAIL     XXXYYY
                                     DETAIL      PDETAIL     XXXYYY00
                                     DETAIL      PDETAIL     XXXYYY01
```

```
Column Descriptions
-------------------

Timespan: The CA MICS timespan where the invalid file
          resides.

DDname: The DDname of the invalid file.

Member: The SAS member name of the invalid file.
```

The Libname Summary report lists space utilizations for each physical data set timespan of the CA MICS database.



```
                                   CA MICS Space Utilization and Reporting            16:18 Friday, January 27, 2006   2
                                           Database Space Estimator
                                               Libname Summary

                                         Space Reporting Unit is Blocks
                                                                                                       Est. Max
                                                                              Used      Allocated  Projected  Allocated
Libname   Dataset Name                                         Block Size     Space       Space      Space      Space
--------  --------------------------------------------  ----------  ----------  ----------  ----------  ----------
PDETAIL   MDI.U3PUN.DETAIL                                        27,648      10,007      10,560      24,179      34,560  (1)
PDAYS     MDI.U3PUN.DAYS                                          27,648       3,130       3,130      13,554      25,530  (1)
PDAYS1    MDI.U3PUN.DAYS1                                         27,648          14      12,171           9      42,176
PWEEKS    MDI.U3PUN.WEEKS                                         27,648         543         920       4,106      18,920  (1)
PMONTHS   MDI.U3PUN.MONTHS                                        27,648         998       1,176       5,213      13,176  (1)
PYEARS    MDI.U3PUN.YEARS                                         27,648           4           6           1       6,006
                                                                          ----------  ----------  ----------  ----------
                                                                              14,696      27,963      47,062     140,368

                 (1) Projected Space Exceeds Allocated; (2) Projected Space Exceeds Estimated Max.
```

```
Column Descriptions
-------------------

 Libname: The SAS libname of the timespan.

 Dataset name: The z/OS data set name of the timespan.

 Block size: The DCB block size of the data set.

 Used Space: Space currently used in the timespan.

 Allocated Space: Total space currently allocated to the
    timespan. This is computed as Used and Free space.

 Projected Space: Estimate of used space within a timespan
    when all files are at maximum cycles.

 Estimated Max Allocated Space: Estimate of the maximum size
    of the timespan if the maximum size of each remaining
    secondary extent is used.
```

The Information Area Summary report shows space utilization for each Information Area within the timespan.

```
                                           CA MICS Space Utilization and Reporting            16:18 Friday, January 27, 2006   3
                                                   Database Space Estimator
                                                   Information Area Summary

                                               Space Reporting Unit is Blocks
                                 Timespan: DETAIL; Libname: PDETAIL; Dataset Name: MDI.U3PUN.DETAIL
                                       Device Type: 3390; Volume Count: 1; Block Size: 27,648
                                   Allocated Blocks: 10,560; Used Blocks: 10,007; % Free: 5.24%
                                              Estimated Max Blocks: 34,560

                                                                 Used                    Avg. Space   Projected    Projected
Information Area                                 Files         Space         %Used       per Cycle        Space        %Used
----------------------------------------------   ------   ----------    ----------    ----------   ----------   ----------
(???) Unknown                                         7           12         0.11%             2           12        0.03%
(ACT) Accounting Component                            6           10         0.09%             2           35        0.10%
(ADM) Administrative Activity Information Area        2            2         0.02%             1            2        0.01%
(AST) ASTEX                                           4            7         0.07%             2            7        0.02%
(AUM) MeasureWare Information Area                   18           79         0.75%             4          247        0.71%
(BAT) Batch Activity Information Area                50        1,480        14.02%            30        4,063       11.76%
(CIC) CICS Activity Information Area                 11           65         0.62%             6          377        1.09%
(DBP) DB2 Buffer Pool Area                            2            3         0.03%             2            3        0.01%
(DB2) DB2 Information Area                           15           78         0.74%             5          126        0.36%
(DEA) VAX/VMS Accounting Information Area             6           14         0.13%             2          131        0.38%
(DEM) VAX/VMS Monitor Information Area                6          145         1.37%            24          145        0.42%
(DES) VAX/VMS System Usage Information Area           5           39         0.37%             8           41        0.12%
(HAR) Hardware Utilization                           22          360         3.41%            16          360        1.04%
(HFS) USS File System Information Area                3            9         0.09%             3           41        0.12%
(HSB) DFHSM Inventory Backup Information Area         4        2,777        26.30%           694        9,255       26.78%
(HSM) DFHSM Inventory Primary Information Area       10          835         7.91%            84        2,831        8.19%
(IDM) IDMS Information Area                           5           15         0.14%             3           50        0.14%
(IMS) IMS Activity Information Area                   7           46         0.44%             7          359        1.04%
(MQA) MQS Accounting Information Area                 4           16         0.15%             4           16        0.05%
(MQS) MQS Performance Information Area               14           27         0.26%             2           39        0.11%
(NVS) Netview Session Information Area                9           16         0.15%             2           24        0.07%
(OPS) Operations Information Area                     3            6         0.06%             2            6        0.02%
(PWR) VSE/POWER Information Area                     19           44         0.42%             2          104        0.30%
(SCP) MVS System Control Program                     36          625         5.92%            17          703        2.03%
(SNT) Network Information Area                       33           66         0.63%             2          150        0.43%
(SRL) System Reliability Information                 52          452         4.28%             9        1,118        3.23%
(TCP) TCP/IP Information Area                        15           36         0.34%             2           44        0.13%
(TDM) TANDEM Information Area                        23          165         1.56%             7          592        1.71%
(TSO) TSO Information Area                           23          333         3.15%            14          349        1.01%
(VCA) VTOC/Catalog Activity Information Area          7           23         0.22%             3          111        0.32%
(VMC) Accounting and Exception Information Are       12           23         0.22%             2          103        0.30%
(VMX) VM/XA and VM/ESA System Information Area       37        1,973        18.68%            53        2,395        6.93%
(WEB) WEB Information Area                           28           61         0.58%             2          115        0.33%
(WLG) WEB Log Files Information Area                  9          114         1.08%            13          198        0.57%
(WLM) MVS Workload Management                        13           30         0.28%             2           30        0.09%
                                                 ------   ----------    ----------    ----------   ----------   ----------
                                                    520        9,986        94.56%             2       24,182       69.97%   (1)

                         (1) Projected Space Exceeds Allocated; (2) Projected Space Exceeds Estimated Max.
```

```
Heading Descriptions
------- -------------

 Timespan: The CA MICS database timespan.

 Libname: The DDname used to reference the timespan data set.

 Dataset Name: The fully qualified data set name of the
 timespan.

 Device Type: The device type of the data set. Note that the
    device type is the same for all volumes of a multi-volume
    data set.

 Volume Count: Number of volumes of the timespan data set.

 Block Size: The z/OS block size of the data set.

 % Free: Percentage of free space in the timespan data set.

 For each of the following items, units can be Bytes, Blocks,
    Tracks, or Cylinders depending upon options selected when
    the report is run.

 Allocated units: The amount of space currently allocated to
    timespan data set.

 Used units: The amount of space currently used in the
    timespan data set.

 Estimated Max units: The estimated maximum space of the
    timespan data set. This is computed by adding the
    requested size of the remaining secondary extents on the
    last volume of the timespan data set to the allocated
    space.

Column Descriptions
-------------------

 Information Area: Name the CA MICS Information Area within
    the timespan.

 Files: Total members in this Information Area. This includes
    valid and invalid (as listed in the Invalid Files report)
    file entries.

 Used Space: Space currently used in the Information Area.
    Note that the space used at the Libname summarization
```

```
   level may not equal the allocated space in the timespan.
   This difference is attributed to internal SAS usage.

%Used: Percentage of allocated space used by the Information
   Area within the timespan.

Avg Space/Cycle: Average space used by each cycle of the
   file within the timespan.

Projected Space: Space used by this information area when
   all cycles of all files are created.

Projected %Used: Percentage of the Estimated Maximum used.
```

The File Summary report shows space utilization for each CA MICS File within the timespan.

```
                                              CA MICS Space Utilization and Reporting             16:18 Friday, January 27, 2006  18
                                                      Database Space Estimator
                                                            File Summary

                                                  Space Reporting Unit is Blocks
                                 Timespan: DETAIL; Libname: PDETAIL; Dataset Name: MDI.U3PUN.DETAIL
                                       Device Type: 3390; Volume Count: 1; Block Size: 27,648
                                    Allocated Blocks: 10,560; Used Blocks: 10,007; % Free: 5.24%
                                                    Estimated Max Blocks: 34,560

Information Area                                              Used                Avg. Space  Def'd   Estimated   Estimated
   File Name                                       Cycles     Space     %Used     per Cycle   Cycles    Space       %Used
-------------------------------------------------  ------  ----------  --------  ----------  ------  ----------  ----------
(WEB) WEB Information Area
   (HTP) HTTP SESSION MANAGER INTERVAL FILE             4           8    0.08%            2       7          14       0.04%
   (MIJ) J2EE METHOD INTERVAL FILE                      4          16    0.15%            4       7          28       0.08%
   (SAF) WAS SERVER ACTIVITY FILE                       4           8    0.08%            2       7          14       0.04%
   (SIF) WAS SERVER INTERVAL FILE                       4           8    0.08%            2       7          14       0.04%
   (SRV) WEBCONTAINER SERVLET INTERVAL FILE             4           8    0.08%            2       7          14       0.04%
   (WCF) HTTP SERVER CONFIGURATION FILE                 1           1    0.01%            1       1           1       0.00%
   (WPR) HTTP SERVER PERFORMANCE FILE                   4           8    0.08%            2       7          14       0.04%
   ----------------------------------------------          ----------  --------                      ----------  ----------
(WEB) WEB Information Area                                         61    0.58%                              115       0.33%

(WLG) WEB Log Files Information Area
   (CFG) WLE CONFIGURATION FILE                         1           2    0.02%            2       1           2       0.01%
   (ERL) HTTP ERROR MESSAGES LOG FILE                   4           8    0.08%            2       7          14       0.04%
   (SAL) HTTP ACCESS LOG FILE                           4         104    0.98%           26       7         182       0.53%
   ----------------------------------------------          ----------  --------                      ----------  ----------
(WLG) WEB Log Files Information Area                              114    1.08%                              198       0.57%

(WLM) MVS Workload Management
   (_EA) REPORT CLASS                                   1           2    0.02%            2       1           2       0.01%
   (_EC) REPORT CLASS RESOURCE CONSUMPTION              1           4    0.04%            4       1           4       0.01%
   (_ED) REPORT CLASS GENERAL EXEC DELAY                1           2    0.02%            2       1           2       0.01%
   (_EM) REPORT CLASS RESPONSE TIME                     1           2    0.02%            2       1           2       0.01%
   (_EW) REPORT CLASS SUBSYSTEM DELAY STATES            1           2    0.02%            2       1           2       0.01%
   (RGP) RESOURCE GROUP                                 1           2    0.02%            2       1           2       0.01%
   (SDE) SERVICE DEFINITION                             1           2    0.02%            2       1           2       0.01%
   (SDS) SERVICE CLASS SUBSYSTEM DELAY STATES           1           2    0.02%            2       1           2       0.01%
   (SEA) SERVICE CLASS                                  1           2    0.02%            2       1           2       0.01%
   (SEC) SERVICE CLASS RESOURCE CONSUMPTION             1           4    0.04%            4       1           4       0.01%
   (SED) SERVICE CLASS GENERAL EXEC DELAY               1           2    0.02%            2       1           2       0.01%
   (SEM) SERVICE CLASS RESPONSE TIME                    1           2    0.02%            2       1           2       0.01%
   (SVC) SERVICE CLIENT                                 1           2    0.02%            2       1           2       0.01%
   ----------------------------------------------          ----------  --------                      ----------  ----------
(WLM) MVS Workload Management                                      30    0.28%                               30       0.09%
   ==============================================          ==========  ========                      ==========  ==========

                         (1) Projected Space Exceeds Allocated; (2) Projected Space Exceeds Estimated Max.
```

```
Heading Descriptions
------- ------------

 Timespan: The CA MICS database timespan.

 Libname: The DDname used to reference the timespan data set.

 Dataset Name: The fully qualified data set name of the
 timespan.

 Device Type: The device type of the data set. Note that the
    device type is the same for all volumes of a multi-volume
    data set.

 Volume Count: Number of volumes of the timespan data set.

 Block Size: The z/OS block size of the data set.

 % Free: Percentage of free space in the timespan data set.

 For each of the following items, units can be Bytes, Blocks,
    Tracks, or Cylinders depending upon options selected when
    the report is run.

 Allocated units: The amount of space currently allocated to
    timespan data set.

 Used units: The amount of space currently used in the
    timespan data set.

 Estimated Max units: The estimated maximum space of the
    timespan data set. This is computed by adding the
    requested size of the remaining secondary extents on the
    last volume of the timespan data set to the allocated
    space.

Column Descriptions
-------------------

 Information Area: Name the CA MICS Information Area within
    the timespan.

 File Name: Name the CA MICS file Area within the timespan.

 Cycles: Total iterations (cycles) of this file within the
    timespan.

 Used Space: Space currently used by the file.  Note that the
    space used at the Libname summarization level may not
```

```
   equal the allocated space in the timespan.  This
   difference is attributed to internal SAS usage.

%Used: Percentage of allocated space used by the File within
   the timespan.

Avg Space/Cycle: Average space used by each cycle within the
   timespan.

Def'D Cycles: Number of cycles as defined in
   prefix.MICS.PARMS(DBMODEL).

Estimated Space: Space used by this file area when all
   cycles created.

Estimated %Used: Percentage of the Estimated Maximum used.
```

4.4.4.7 Component Inactive Data Elements Report

```
------------------ Component Inactive Data Elements Report -----------------
Command ===>


Product:           ===>     (Enter component ID)

Print dictionary:  ===>     (Y/N)


Press END to continue.
```

The Component Inactive Data Elements Report provides a listing of data elements that are inactive for a component based on the options specified in the cccGENIN members (where ccc is the component identifier) of sharedprefix.MICS.GENLIB, and complex level database options in sharedprefix.MICS.PARMS(CPLXDEF). Inactive data elements are not stored in the database.

The report shows the names of the inactive data elements by file and these reasons for their inactivity:

- The file is switched off completely by settings in the FILE statement of cccGENIN, so the data elements are not stored in the database.

- The data element is made inactive for all timespans by settings in the timespan mask of the NAME or NAMX statement of cccGENIN.

- The data element is not stored in the database because the ESSENTIAL INITIAL or ONLY option is in effect and this element is not marked as essential.

- The data element is not stored in the database because the DERIVED COMP option is in effect, which allows recalculation of an element from existing elements when necessary.

- An OPTIONS statement in cccGENIN specifies a cluster code that excludes this data element from the database.

- A product option validation, such as file and timespan dependencies, results in the exclusion of this data element from the database.

The report also shows an alias name for the data element, if one is used. This will aid in locating the corresponding CA MICS data dictionary entry for further information.

Optionally, a detailed description of each listed data element as it appears in the CA MICS data dictionary, is printed at the end of the report.

Refer to Chapter 2 of the PIOM for more information on complex level database options and Chapter 4 of the System Administrator Guide for the component, file level and data element options.

To produce the Component Inactive Data Elements Report, choose option 7 from the CA MICS Administrator Utilities menu (MWF 5;4;7).

Fill in a value, as described below, and press END or use the F3 key to submit the inquiry.

Product

Required, no default. Enter one of the following three-character (ccc) component identifiers.

AUM AST BAS CIC DB2 DEX HSM
IDM IMS MQS MTI PWR RMF SMF
SNT SRL TDM TSO VCA VMC WEB

Print dictionary

Required, no default. Enter Y(es) or N(o). If Y(es), the dictionary entries with detailed descriptions of the inactive data elements are printed.

A sample report listing for the Component Inactive Data Elements Report follows.

In the following example, inactive data elements for the Open Edition/MVS Program File of the SMF component are listed. All listed data elements are inactive due to timespan option settings. No data dictionary entry printing was requested.

```
+----------------------------------------------------------------------------------------------------------------------------+
                                            CA MICS COMPONENT INACTIVE DATA ELEMENTS            05:23 Friday, April 13, 2007   1
                                          DATA ELEMENTS TURNED OFF FOR COMPONENT: SMF

  --------------------------------------------------- AREA=BAT FILEID=_OE -------------------------------------------------------

    DATA      FILE IS      OFF DUE TO         OFF DUE TO        OFF DUE TO        OFF DUE TO      DATA DIC.     PRODUCT OPTION
  ELEMENT       OFF     TIMESPAN OPTIONS   ESSENTIAL OPTION   DERIVED OPTION    CLUSTER OPTION     ALIAS        RESET STATUS

  PGMCPHRS      NO            YES                NO                 NO                NO                              NO
  PGMPNM        NO            YES                NO                 NO                NO                              NO
  PGMRKHRS      NO            YES                NO                 NO                NO                              NO
  PGMRVN        NO            YES                NO                 NO                NO                              NO
  PGMTIHRS      NO            YES                NO                 NO                NO                              NO
  PGMTOHRS      NO            YES                NO                 NO                NO                              NO
  PGMUIHRS      NO            YES                NO                 NO                NO                              NO
  PGMVKHRS      NO            YES                NO                 NO                NO                              NO
  PGMCVIO       NO            YES                NO                 NO                NO                              NO

+----------------------------------------------------------------------------------------------------------------------------+
Heading Descriptions
--------------------

 AREA: The name of the CA MICS information area.

 FILEID: The name of the CA MICS file.

Column Descriptions
-------------------

 DATA ELEMENT: The name of the CA MICS data element defined
  in cccGENIN.

 FILE IS OFF: If YES, the file is switched off completely by
  settings in the FILE statement of cccGENIN.  Therefore the
  data element is not stored in the data base.

 OFF DUE TO TIMESPAN OPTIONS: If YES, the data element is
  made inactive for all timespans by settings in the timespan
  mask of the NAME or NAMX statement of cccGENIN, and is not
  stored in the data base.

 OFF DUE TO ESSENTIAL OPTION: If YES, the ESSENTIAL INITIAL
  or ONLY option is in effect and this data element is not
  marked as essential.  Therefore the data element is not
  stored in the data base.

 OFF DUE TO DERIVED OPTION: If YES, the DERIVED COMP option
  is in effect and this data element is marked as derived,
  which allows it to be recalculated from other elements.
  Therefore the data element is not stored in the data base.

 OFF DUE TO CLUSTER OPTION: If YES, an OPTIONS statement in
  cccGENIN specifies a cluster code that excludes this
  data element from the data base.

 DATA DIC. ALIAS: If present, displays an alias name for this
  data element used for the data dictionary entry.

 PRODUCT OPTION RESET STATUS: If YES, a product option
  validation, such as file and timespan dependencies, has
  switched off this data element, which results in the
  exclusion from the data base.  The validation executes part
  of the same code that is used during component generation
  (cccCGEN) and is different for each component.
```

4.4.4.8 SAS Format/Informat Print Utility

The SAS Format and Informat Print Utility is an administrative tool to print SAS informats or formats that reside in the CA MICS sharedprefix or unit level format libraries.

This utility can print informat and formats from both the complex level format library and the unit level library in a single execution. It can also process multiple format catalogs in a single execution.

The SAS Format and Informat Print Utility provides the following options:

o Print all formats within a format catalog or multiple format catalogs.

o Print just the directory listing of a catalog or multiple catalogs.

o Specify individual or multiple formats by name. The utility supports colon (:) and hyphen (-) wildcard characters within the format name specification.

o Report simultaneously from both the complex and one unit.

The following sections provide more details concerning the SAS Format/Informat Print Utility:

1 - Panel Entries
2 - Sample Print Directory MICSLOG Output
3 - Sample Print Directory Report Output
4 - Sample Print Specific Format MICSLOG Output
5 - Sample Print Specific Formats Report Output

4.4.4.8.1 Panel Entries

```
 Command ===>

----------------------  SAS Format/Informat Print Utility  -----------



Complex Level Format Library?  ===> N   (Y/N)

Format Catalog:  USERFMT1  ===> N   MICSFMTS  ===> N   USERFMT2  ===> N
                 Other:    ===> _______________________  (specify name)

Print Entire Catalog?   ===> N      Print Catalog Directory?   ===> N

 - or -
Format Name(s):  ===>  ________________________________________________
_______________________________________________________________________


Unit Level Format Library?  ===> N      (Y/N)       DBID  ===> P

Format Catalog:  USERFMT1  ===> N   MICSFMTS  ===> N   USERFMT2  ===> N
                 Other:    ===> _______________________  (specify name)

Print Entire Catalog?   ===> N      Print Catalog Directory?   ===> N
 - or -
Format Names(s):  ===> ________________________________________________
_______________________________________________________________________
```

The top half of the panel controls the complex level reporting while the bottom half of the panel controls unit level reporting.

Complex Level Format Library? - Required. Defaults to N. Specify Y to print formats from one of the complex level (MCOLIB) format catalogs.

Format Catalog: USERFMT1 ===> MICSFMTS ===> USERFMT2 ===>
Required. Defaults to N. Specify Y for each catalog from which you want to print formats.

Format Catalog: OTHER ===>
Defaults to blank. Must be specified if all of the format catalogs listed above are set to N. Contains the exact name of a user format catalog that resides in either the complex level format library (MCOLIB) or the unit level format library (MUOLIB).

Print Entire Catalog? - Required only if 'Format Name(s)' is left blank. Defaults to N. Specify Y to print all the formats for the format catalogs you have selected for processing.

Print Catalog Directory? - Required. Defaults to N. Specify Y to print just a listing of all the members of the format catalogs you have selected for processing.

Format Names(s): - Required only if 'Print Entire Catalog' is set to N. Defaults to blank. Specify the names of any formats and/or informats you want to print.

Note: You must specify format names exactly as they appear in the catalog using the naming conventions required by SAS.

o Precede names of entries that contain character formats with a dollar sign ($).

o Precede names of entries that contain character informats with an at sign (@) and a dollar sign($), for example, @$entry-name.

o Specify names of entries that contain numeric formats without a prefix.

o Precede names of entries that contain numeric informats with an at sign (@).

You can use the colon (:) and hyphen (-) wildcard characters to specify entries. For example, the following prints all character formats that begin with cplx.

```
$cplx:
```

Unit Level Format Library? - Required. Defaults to N. Specify Y to print formats from one unit's (MUOLIB) format catalogs.

All the remaining fields are the same as those described above for complex level reporting.

4.4.4.8.2 Sample Print Directory MICSLOG Output

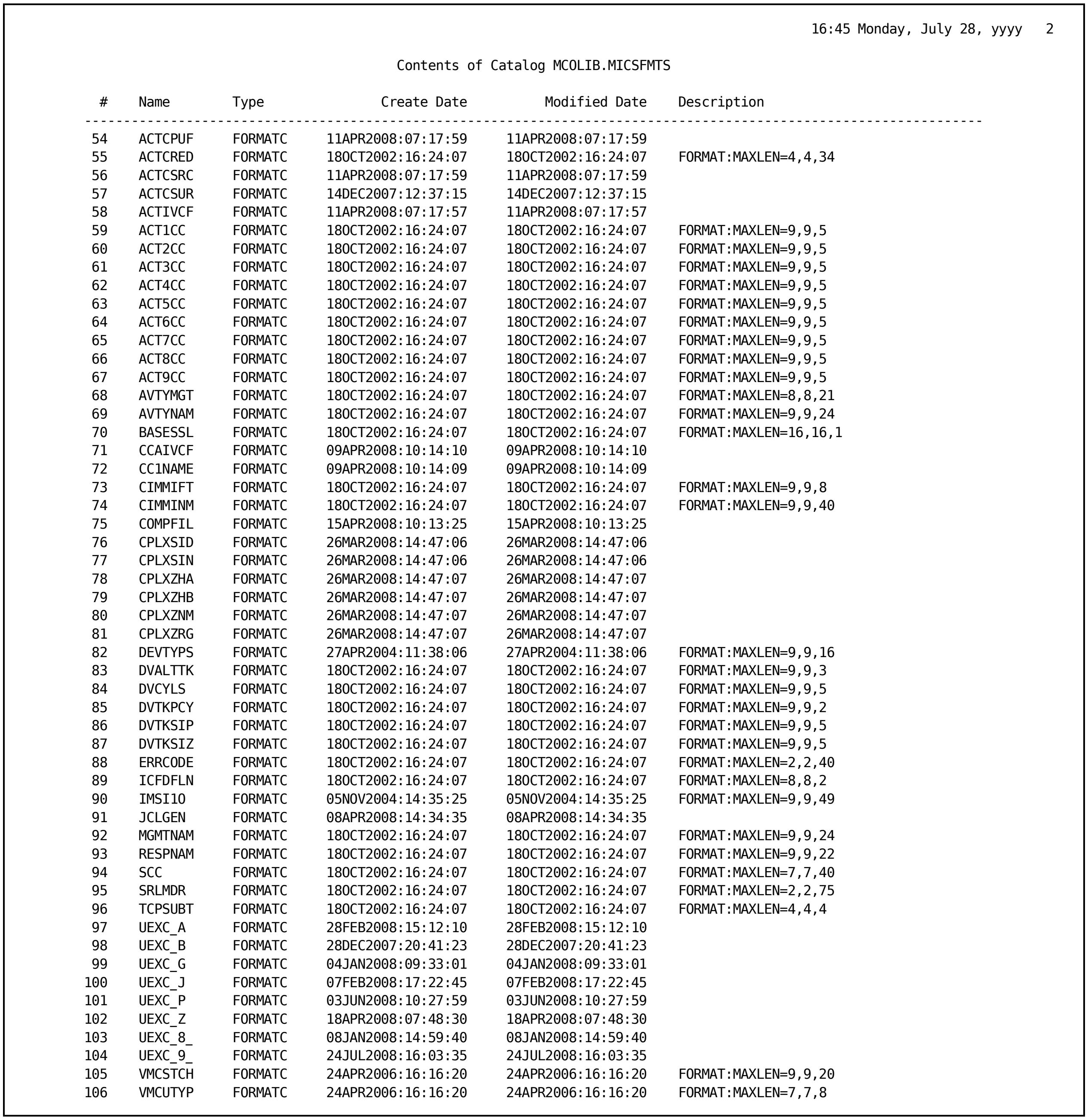
A sample MICSLOG for the SAS Format/Informat Print Utility follows. This sample MICSLOG was generated from the panel option 'Print Catalog Directory?'.

```
Monday, 28 July yyyy---------------------------------------- MICS LOG --------------------------------------------------------------
16.45.18 ICF01098I MICF INQUIRY SAS EXECUTION STARTED.
16.45.18 MWF02000I SAS Print Format Utility Execution Options:
16.45.18 MWF02000I Complex Level Formats: Y
16.45.18 MWF02000I USERFMT1 Catalog: N
16.45.18 MWF02000I MICSFMTS Catalog: Y
16.45.18 MWF02000I USERFMT2 Catalog: N
16.45.18 MWF02000I    OTHER Catalog:
16.45.18 MWF02000I Print Catalog Directory: Y
16.45.18 MWF02000I Print Entire Format Catalog: N
16.45.18 MWF02000I Print Specified Formats: N
16.45.19 ICF01099I MICF INQUIRY SAS EXECUTION COMPLETED.
```

4.4.4.8.3 Sample Print Directory Report Output

A sample report listing for the SAS Format/Informat Print Utility follows. This sample report was generated from the panel option 'Print Catalog Directory?'.

```
                                                                                             16:45 Monday, July 28, yyyy   1

                                        Contents of Catalog MCOLIB.MICSFMTS

   #   Name        Type               Create Date          Modified Date    Description
-------------------------------------------------------------------------------------------------------------------
   1   ACTPRNM     FORMAT      14DEC2007:12:37:18     14DEC2007:12:37:18
   2   CIMYESNO    FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,3
   3   CPLXSFZN    FORMAT      26MAR2008:14:47:07     26MAR2008:14:47:07
   4   CPLXZONE    FORMAT      26MAR2008:14:47:07     26MAR2008:14:47:07
   5   DSORG       FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,3
   6   DVALTTK     FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,3
   7   DVCLASS     FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,8
   8   DVCYLS      FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,5
   9   DVTKPCY     FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,2
  10   DVTKSIP     FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,5
  11   DVTKSIZ     FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,5
  12   DVTYPE      FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,8
  13   FTPFAIL     FORMAT      10NOV2003:14:17:35     10NOV2003:14:17:35
  14   LUSEL       FORMAT      10NOV2003:14:17:35     10NOV2003:14:17:35
  15   MAPCT       FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,9
  16   MASRLCT     FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    SRL     SRLFMT03  SOURCE  01.05 SRL6065
  17   MBYTES      FORMAT      24APR2006:16:16:13     24APR2006:16:16:13
  18   MNTHFMT     FORMAT      03JUN2008:08:47:58     03JUN2008:08:47:58    FORMAT:MAXLEN=16,16,9
  19   MONEY       FORMAT      03JUN2008:08:47:58     03JUN2008:08:47:58    FORMAT:MAXLEN=16,16,17
  20   MONEYB      FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,14
  21   MONEYP      FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,13
  22   MONEYR      FORMAT      03JUN2008:08:47:58     03JUN2008:08:47:58    FORMAT:MAXLEN=16,16,13
  23   MONEYRT     FORMAT      19MAY2005:12:39:39     19MAY2005:12:39:39
  24   MONEYV      FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,8
  25   MONEYX      FORMAT      03JUN2008:08:47:58     03JUN2008:08:47:58    FORMAT:MAXLEN=16,16,19
  26   RECFM       FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,5
  27   SESTYPE     FORMAT      10NOV2003:14:17:35     10NOV2003:14:17:35
  28   SHROPT      FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,2
  29   SOCST       FORMAT      10NOV2003:14:17:35     10NOV2003:14:17:35
  30   SSL         FORMAT      10NOV2003:14:17:35     10NOV2003:14:17:35
  31   TDMDEVTP    FORMAT      03JUN2008:08:51:40     03JUN2008:08:51:40    FORMAT:MAXLEN=16,16,8
  32   TDMDSKTP    FORMAT      03JUN2008:08:51:40     03JUN2008:08:51:40    FORMAT:MAXLEN=16,16,25
  33   TDMEVYTP    FORMAT      03JUN2008:08:51:40     03JUN2008:08:51:40    FORMAT:MAXLEN=16,16,24
  34   TDMEXPTP    FORMAT      03JUN2008:08:51:40     03JUN2008:08:51:40    FORMAT:MAXLEN=16,16,24
  35   TDMMISTP    FORMAT      03JUN2008:08:51:40     03JUN2008:08:51:40    FORMAT:MAXLEN=16,16,24
  36   TDMMLNTP    FORMAT      03JUN2008:08:51:40     03JUN2008:08:51:40    FORMAT:MAXLEN=16,16,24
  37   TDMPRTTP    FORMAT      03JUN2008:08:51:40     03JUN2008:08:51:40    FORMAT:MAXLEN=16,16,24
  38   TDMTAPTP    FORMAT      03JUN2008:08:51:40     03JUN2008:08:51:40    FORMAT:MAXLEN=16,16,24
  39   TDMTRMTP    FORMAT      03JUN2008:08:51:40     03JUN2008:08:51:40    FORMAT:MAXLEN=16,16,24
  40   TDMTR2TP    FORMAT      03JUN2008:08:51:40     03JUN2008:08:51:40    FORMAT:MAXLEN=16,16,24
  41   VMCCWOP     FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,8
  42   VMCPRVA     FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,5
  43   VMCPRVB     FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,5
  44   VMDEVCLS    FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,8
  45   VMDEVTYP    FORMAT      18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,25
  46   VMXDVCLS    FORMAT      24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=16,16,11
  47   VMXDVTYP    FORMAT      24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=16,16,11
  48   XBYTES      FORMAT      24APR2006:16:16:13     24APR2006:16:16:13
  49   YR          FORMAT      24APR2006:16:16:13     24APR2006:16:16:13
  50   ACTCDIS     FORMATC     14DEC2007:12:37:15     14DEC2007:12:37:15
  51   ACTCICF     FORMATC     11APR2008:07:17:59     11APR2008:07:17:59
  52   ACTCMPF     FORMATC     11APR2008:07:17:59     11APR2008:07:17:59
  53   ACTCOMP     FORMATC     11APR2008:07:17:59     11APR2008:07:17:59
```



```
                                                                              16:45 Monday, July 28, yyyy   2

                                          Contents of Catalog MCOLIB.MICSFMTS

      #    Name       Type                Create Date          Modified Date    Description
    ---------------------------------------------------------------------------------------------------------------
     54    ACTCPUF    FORMATC     11APR2008:07:17:59     11APR2008:07:17:59
     55    ACTCRED    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=4,4,34
     56    ACTCSRC    FORMATC     11APR2008:07:17:59     11APR2008:07:17:59
     57    ACTCSUR    FORMATC     14DEC2007:12:37:15     14DEC2007:12:37:15
     58    ACTIVCF    FORMATC     11APR2008:07:17:57     11APR2008:07:17:57
     59    ACT1CC     FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,5
     60    ACT2CC     FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,5
     61    ACT3CC     FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,5
     62    ACT4CC     FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,5
     63    ACT5CC     FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,5
     64    ACT6CC     FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,5
     65    ACT7CC     FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,5
     66    ACT8CC     FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,5
     67    ACT9CC     FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,5
     68    AVTYMGT    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=8,8,21
     69    AVTYNAM    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,24
     70    BASESSL    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=16,16,1
     71    CCAIVCF    FORMATC     09APR2008:10:14:10     09APR2008:10:14:10
     72    CC1NAME    FORMATC     09APR2008:10:14:09     09APR2008:10:14:09
     73    CIMMIFT    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,8
     74    CIMMINM    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,40
     75    COMPFIL    FORMATC     15APR2008:10:13:25     15APR2008:10:13:25
     76    CPLXSID    FORMATC     26MAR2008:14:47:06     26MAR2008:14:47:06
     77    CPLXSIN    FORMATC     26MAR2008:14:47:06     26MAR2008:14:47:06
     78    CPLXZHA    FORMATC     26MAR2008:14:47:07     26MAR2008:14:47:07
     79    CPLXZHB    FORMATC     26MAR2008:14:47:07     26MAR2008:14:47:07
     80    CPLXZNM    FORMATC     26MAR2008:14:47:07     26MAR2008:14:47:07
     81    CPLXZRG    FORMATC     26MAR2008:14:47:07     26MAR2008:14:47:07
     82    DEVTYPS    FORMATC     27APR2004:11:38:06     27APR2004:11:38:06    FORMAT:MAXLEN=9,9,16
     83    DVALTTK    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,3
     84    DVCYLS     FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,5
     85    DVTKPCY    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,2
     86    DVTKSIP    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,5
     87    DVTKSIZ    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,5
     88    ERRCODE    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=2,2,40
     89    ICFDFLN    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=8,8,2
     90    IMSI1O     FORMATC     05NOV2004:14:35:25     05NOV2004:14:35:25    FORMAT:MAXLEN=9,9,49
     91    JCLGEN     FORMATC     08APR2008:14:34:35     08APR2008:14:34:35
     92    MGMTNAM    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,24
     93    RESPNAM    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=9,9,22
     94    SCC        FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=7,7,40
     95    SRLMDR     FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=2,2,75
     96    TCPSUBT    FORMATC     18OCT2002:16:24:07     18OCT2002:16:24:07    FORMAT:MAXLEN=4,4,4
     97    UEXC_A     FORMATC     28FEB2008:15:12:10     28FEB2008:15:12:10
     98    UEXC_B     FORMATC     28DEC2007:20:41:23     28DEC2007:20:41:23
     99    UEXC_G     FORMATC     04JAN2008:09:33:01     04JAN2008:09:33:01
    100    UEXC_J     FORMATC     07FEB2008:17:22:45     07FEB2008:17:22:45
    101    UEXC_P     FORMATC     03JUN2008:10:27:59     03JUN2008:10:27:59
    102    UEXC_Z     FORMATC     18APR2008:07:48:30     18APR2008:07:48:30
    103    UEXC_8_    FORMATC     08JAN2008:14:59:40     08JAN2008:14:59:40
    104    UEXC_9_    FORMATC     24JUL2008:16:03:35     24JUL2008:16:03:35
    105    VMCSTCH    FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=9,9,20
    106    VMCUTYP    FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=7,7,8
```

```
                                                                                     16:45 Monday, July 28, yyyy   3
                                              Contents of Catalog MCOLIB.MICSFMTS

          #    Name        Type              Create Date         Modified Date    Description
        ------------------------------------------------------------------------------------------------------------------
        107    VMDVBYT     FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=9,9,6
        108    VMDVSPC     FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=9,9,7
        109    VMP200X     FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=9,9,3
        110    VMP300X     FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=9,9,3
        111    VMP308X     FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=9,9,4
        112    VMP309X     FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=9,9,4
        113    VMP43XX     FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=9,9,5
        114    VMP438X     FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=9,9,5
        115    VMP902X     FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=9,9,3
        116    VMP912X     FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=9,9,3
        117    VMP922X     FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=9,9,3
        118    VMP967X     FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=9,9,4
        119    VMXDSDT     FORMATC     24APR2006:16:16:20     24APR2006:16:16:20    FORMAT:MAXLEN=9,9,8
        120    XBYTES      INFMT       24APR2006:16:16:13     24APR2006:16:16:13
        121    YR          INFMT       24APR2006:16:16:13     24APR2006:16:16:13
```

4.4.4.8.4 Sample Print Specific Format MICSLOG Output

A sample MICSLOG for the SAS Format/Informat Print Utility follows. This sample MICSLOG was generated from the panel option 'Print Format Name(s)'.

```
Monday, 28 July yyyy--------------------------------------- MICS LOG --------------------------------------------------------------
17.38.33 ICF01098I MICF INQUIRY SAS EXECUTION STARTED.
17.38.33 MWF02000I SAS Print Format Utility Execution Options:
17.38.33 MWF02000I Complex Level Formats: Y
17.38.33 MWF02000I USERFMT1 Catalog: N
17.38.33 MWF02000I MICSFMTS Catalog: Y
17.38.33 MWF02000I USERFMT2 Catalog: N
17.38.33 MWF02000I    OTHER Catalog:
17.38.33 MWF02000I Print Catalog Directory: N
17.38.33 MWF02000I Print Entire Format Catalog: N
17.38.33 MWF02000I Print Specified Formats: Y
17.38.33 ICF01099I MICF INQUIRY SAS EXECUTION COMPLETED.
```

4.4.4.8.5 Sample Print Specific Formats Report Output

A sample report for the SAS Format/Informat Print Utility follows. This sample report was generated from the panel option 'Print Format Name(s)'.

```
                                                                                                    17:38 Monday, July 28, yyyy   1
Print from Library: MCOLIB Catalog: MICSFMTS
                                                                                                    17:38 Monday, July 28, yyyy   2

                            ----------------------------------------------------------------------------
                            |       FORMAT NAME: $CPLXSID LENGTH:   24   NUMBER OF VALUES:    1        |
                            |   MIN LENGTH:  40  MAX LENGTH:  40  DEFAULT LENGTH  40  FUZZ:       0   |
                            |--------------------------------------------------------------------------|
                            |START           |END             |LABEL  (VER. V7|V8   26MAR2008:14:47:06)|
                            |----------------+----------------+----------------------------------------|
                            |OSYS            |OSYS            |LSYS CPU-MODEL . SCP . .                |
                            ----------------------------------------------------------------------------

                           ----------------------------------------------------------------------------
                            |       FORMAT NAME: $CPLXSIN LENGTH:   30   NUMBER OF VALUES:    1        |
                            |   MIN LENGTH:  40  MAX LENGTH:  40  DEFAULT LENGTH  40  FUZZ:       0   |
                            |--------------------------------------------------------------------------|
                            |START           |END             |LABEL  (VER. V7|V8   26MAR2008:14:47:06)|
                            |----------------+----------------+----------------------------------------|
                            |LSYS            |LSYS            |SYSTEM DESCRIPTION                      |
                            ----------------------------------------------------------------------------

                           ----------------------------------------------------------------------------
                            |       FORMAT NAME: $CPLXZHA LENGTH:    1   NUMBER OF VALUES:    1        |
                            |   MIN LENGTH:   8  MAX LENGTH:   8  DEFAULT LENGTH   8  FUZZ:       0   |
                            |--------------------------------------------------------------------------|
                            |START           |END             |LABEL  (VER. V7|V8   26MAR2008:14:47:07)|
                            |----------------+----------------+----------------------------------------|
                            |                |                |                                        |
                            ----------------------------------------------------------------------------

                            ----------------------------------------------------------------------------
                            |       FORMAT NAME: $CPLXZHB LENGTH:    1   NUMBER OF VALUES:    1        |
                            |   MIN LENGTH:   8  MAX LENGTH:   8  DEFAULT LENGTH   8  FUZZ:       0   |
                            |--------------------------------------------------------------------------|
                            |START           |END             |LABEL  (VER. V7|V8   26MAR2008:14:47:07)|
                            |----------------+----------------+----------------------------------------|
                            |                |                |                                        |
                            ----------------------------------------------------------------------------

                            ----------------------------------------------------------------------------
                            |       FORMAT NAME: $CPLXZNM LENGTH:   12   NUMBER OF VALUES:    0        |
                            |   MIN LENGTH:   1  MAX LENGTH:  40  DEFAULT LENGTH  40  FUZZ:       0   |
                            |--------------------------------------------------------------------------|
                            |START           |END             |LABEL  (VER. V7|V8   26MAR2008:14:47:07)|
                            |----------------+----------------+----------------------------------------|
                            ----------------------------------------------------------------------------

                            ----------------------------------------------------------------------------
                            |       FORMAT NAME: $CPLXZRG LENGTH:   12   NUMBER OF VALUES:    0        |
                            |   MIN LENGTH:   6  MAX LENGTH:   6  DEFAULT LENGTH   6  FUZZ:       0   |
                            |--------------------------------------------------------------------------|
                            |START           |END             |LABEL  (VER. V7|V8   26MAR2008:14:47:07)|
                            |----------------+----------------+----------------------------------------|
                            ----------------------------------------------------------------------------
```

4.4.4.9 Physical Hardware Parameter Creation

```
---------------  Physical Hardware Parm Creation  ------- Row 1 of 3
Command ===>                                             Scroll ===> CSR

Line Commands: S Select one of the following Unit IDs
                 to execute RMFPHYCP

Cmd  Unit ID  Description
 -   -------  ----------------------------------------
 _   G        C A
 _   P        CA
 _   8        C A

******************************* Bottom of data ************************
```

Use the Physical Hardware Create Parms (RMFPHYCP) utility to create member RMFPHYDU in your prefix.MICS.PARMS library. This member can then be modified to provide values to be assigned during the creation of the Physical Hardware (HARPHY) file. The RMFPHYCP utility can be run in any unit that has the HARPHY file activated and where data element assignments are required.

Member RMFPHYDU will contain CPC, DEV, and LDV key ID statements representing hardware found in the Individual Processor Activity (HARIPU) and Device Activity (HARDVA) files.

Note that RMFPHYCP does not create the RMFPHYDE member used by RMFPGEN, but instead creates the RMFPHYDU PARM member, leaving an existing RMFPHYDE production member unmodified.

Once RMFPHYDU is updated, it can be copied or renamed to RMFPHYDE, and RMFPGEN can be run. RMFPGEN creates the $RMFPHY format in the unit level MUOLIB. $RMFPHY is subsequently used by the DAY020 daily processing job to assign values to HARPHY data elements.

The first time RMFPHYCP is run in a given unit, the RMFPHYDU PARM member is created exclusively from the HARDVA and HARIPU files, comprised of only CPC, DEV, and LDV statements. Any subsequent user updates to add data element assignments can then be made, and RMFPHYDU is copied or renamed to RMFPHYDE in prefix.MICS.PARMS.

Subsequent runs of the RMFPHYCP utility to discover any new physical CPCs or devices or logical device groups create a new RMFPHYDU prefix.MICS.PARMS member. If an existing

RMFPHYDE member is found in prefix.MICS.PARMS, its information is retained (including any user updates) and any new CPC, DEV, and LDV statements for physical CPCs or devices or logical device groups are added to create the new RMFPHYDU prefix.MICS.PARMS member.

Note: Executing RMFPHYCP overlays the existing RMFPHYDU member in prefix.MICS.PARMS. Be sure to copy or rename RMFPHYDU to RMFPHYDE before rerunning RMFPHYCP.

4.4.4.10 Missing Data Elements Report Utility

```
Note:  This utility only executes at SAS Version 9 or later.

----------------  Missing Data Element Report Utility -----------------

 Command ===>

This utility generates a report across all units, timespans and
online files within a component, and reports on those elements that
contain missing values 100% of the time.

These elements may be candidates for deactivation, but we strongly
urge you to use caution in deactivating elements via cluster codes
assigned to releases that you may not have migrated to yet.

To begin, enter a 3-character component identifier below.  You will
get an opportunity to exclude units on a subsequent panel.


CA MICS Component ===> ___  (ccc)


.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

Select a component, then press ENTER to continue.

A list of units where the component is installed is presented to allow you to EXCLUDE units from the analysis.

```
-------------------  Missing Data Elements Report Utility --- Row 1 to n of n

 Command ===>

Line Commands: X = Exclude unit from the report

Cmd  Unit ID    Name                       Label
 -   -------  --------   ----------------------------------------------
 _   A        UNITA      Label for UNIT A
 _   P        UNITP      Label for UNIT P
 _   X        UNITX      Label for UNIT X
 _   Z        UNITZ      Label for UNIT Z
 _   0        UNIT0      Label for UNIT 0



.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

The Missing Data Elements Report Utility accepts a CA MICS component identifier (for example, RMF) and identifies all units in the CA MICS complex where the component is installed. The database files associated with the identified units are examined to identify any data elements that have missing values in all timespans where the data elements are active.

After the analysis is complete, a report is produced:

o Section 1 shows component files where every data element has a non-missing value in at least one observation.

o Section 2 shows component files with data elements that have missing values in all observations.

To produce the Missing Data Elements Report, choose option 10 from the CA MICS Administrator Utilities menu (MWF 5;4;A).

Enter the CA MICS component as described below and press ENTER to begin report generation.

CA MICS Component   Required - Enter one of the following three-character (ccc) component identifiers.

ACT AUM AST BAS CIC DB2
DEX HSM IDM IMS MQS MTI
PWR RMF SMF SNT SRL TDM
TLM TSO VCA VMC VTS WEB

You are provided with a list of units where the component is installed. By default, the databases of all units are examined. To exclude a unit from consideration, place an "X" next to the unit to be excluded. Then press ENTER to refresh the list of units. When the list is complete, either press PF3 or type the END command to generate the report JCL.

Note: CA MICS data elements are activated at the complex level. In general, you should allow all unit databases to be selected. Only exclude units if they contain data that is not representative of your normal configuration. Consider a situation where the component you have selected is deployed in three units, A, B, and P, but is no longer used in unit B. If the component files in unit B are still in that unit's database, the data may not represent your current cccGENIN configuration and the data values may not represent your current environment. In this example, you would want to exclude unit B from the analysis.

After making any JCL modification (for example, update JOBCARD), enter the SUBMIT command to submit the job.

The report that is generated has two possible sections:

Section 1: Files where every data element has a value in at least one observation.

Section 2: Files where at least one data element has missing values in all timespans and observations.

Note: If no component files meet the section criteria, that report section is not generated. For example, if all files contain at least one data element that has missing values in all timespans and observations, section 1 will not be produced.

If a particular file has zero (0) observations in all timespans, it is not included in the report, but will be noted in the MICSLOG as shown below:

```
  BAS00196I ===> File: HARCCU **NO OBS**
```

If all component files have zero (0) observations in all timespans, MICSLOG will note that the report will not be generated.

Each element listed in Section 2 shows the units and timespans where the data element was present and found to have missing values in every database observation.

As the job is running, the MICSLOG shows progress as first the individual databases are examined for component files, and next as each component file is examined for data elements with missing values.

A sample of REPORT SECTION 1, executed for the CICS component, is shown below. Section 1 shows files where every data element, across all file instances where the elements were active, had at least one observation with a non-missing value. These files are ideally configured.

```
                                             CA MICS Missing Data Element Report
 Complex sharedprefix: PROD.MICS                                                                                        Page 001
 Component Analyzed:   CIC                                                                                     Run Date: 17FEByy

+******************************************************************************************************+
+                                 Files With 100% Populated Data Elements                              +
+       Each Data Element in the Listed Files has at Least One Observation with a Non-Missing Value    +
+                                                                                                      +
+  FILE      FILE LABEL                                                                                +
+  ------    --------------------------------------------                                              +
+  CICCDC    CICS Dictionary File                                                                      +
+  CTGGSA    Gateway Server Activity                                                                   +
+******************************************************************************************************+
```

A sample of the beginning of REPORT SECTION 2, executed for the CICS component, is shown below. Section 2 shows files where at least one data element had missing values in every observation across all file instances where the element is active. These file elements should be reviewed to determine if you should keep them activated.

```
                                             CA MICS Missing Data Element Report
 Complex sharedprefix: PROD.MICS                                                                                        Page 002
 Component Analyzed:   CIC                                                                                     Run Date: 17FEByy

+******************************************************************************************************+
+            Files with at Least One Data Element with Missing Values in ALL Observations              +
+            Every Data Element Listed Below has Missing Values in each File Observation               +
+                                                                                                      +
+******************************************************************************************************+
+  FILE      FILE LABEL                                                                                +
+  ------    --------------------------------------------                                              +
+  CICCIN    CICS Incident File                                                                        +
+                                                                                                      +
+ ELEMENT   DATA ELEMENT LABEL                            XDWMY  UNIT IDs                              +
+ --------  --------------------------------------------  -----  ------------------------------------  +
+ OPERID    Terminal Operator Initials                    X....  Z0                                    +
+ TERMINAL  CICS Terminal ID                              X....  Z0                                    +
+******************************************************************************************************+
+                                                                                                      +
+******************************************************************************************************+
+  FILE      FILE LABEL                                                                                +
+  ------    --------------------------------------------                                              +
+  CICCSU    CICS User Activity File                                                                   +
+                                                                                                      +
+ ELEMENT   DATA ELEMENT LABEL                            XDWMY  UNIT IDs                              +
+ --------  --------------------------------------------  -----  ------------------------------------  +
+ CSUCHMDC  No. CICS Dispatcher Change Modes              XD.M.  A0                                    +
+ CSUCONWT  DB2 Connection Wait Time                      .D.M.  A0                                    +
+ CSUDB2RC  Total No. DB2 Requests                        .D.M.  A0                                    +
+ CSUDB2WT  DB2 Wait Time                                 .D.M.  A0                                    +
+ CSUERSTM  Excessive Response Time Total                 .D.M.  AZ0                                   +
+ CSUETRN   Excessive Transactions Processed              .D.M.  AZ0                                   +
+ CSUIPCNM  TCP/IP Connection Name                        X....  0                                     +
+ CSUMXETM  Max Excessive Response Time                   .D.M.  AZ0                                   +
+ CSURDQWT  DB2 ReadyQ Wait Time                          .D.M.  A0                                    +
+ CSUSUSDT  Unidentified Suspend Wait Time                .D.M.  AZ                                    +
+ CSUTCLAS  Transaction Class at Task Creation            X....  0                                     +
+ CSUTCSTG  TIOA Storage Allocated to Terminal            XD.M.  A0                                    +
+ OPERID    Terminal Operator Initials                    X....  0                                     +
+******************************************************************************************************+
+ Analysis of file: CICCSX                                                                             +
+ ELEMENT   DATA ELEMENT LABEL                            TSPNS  UNITS                                 +
+ --------  --------------------------------------------  -----  --------------------------            +
+ CSYAMXTC  Total Task During Active Max Task             XDWM.  AZ                                    +
+ CSYAVAMX  Average Active Task Limit                     .DWM.  A0                                    +
+ CSYAVLDS  Average Loader Suspend Time                   .DWM.  A0                                    +
+ CSYAVNIU  Average NIU Queue Membership Time             .DWM.  A0                                    +
+ CSYCOST   Processing Charges                            XDWM.  AZ0                                   +
```

4.4.5 Data Entry

Most CA MICS files are derived from measurement data produced by one of the common software monitors such as SMF or CA TSO/MON PM. For some CA MICS files, data can be entered manually, either to update standard measurement data or as the sole source of input to the file.

The MWF SAS With CA MICS Libraries (MSAS) application supports manual data entry for the Operations Information Area of the Batch and Operations Analyzer. See the CA MICS Batch and Operations Analyzer Guide for information on Operations Information Area files and data entry requirements.

For instructions on manual data entry for the following CA MICS products, see the:

o CA MICS Accounting and Chargeback User Guide for the Accounting Information Area of CA MICS Accounting and Chargeback

o CA MICS Capacity Planner Guide for the capacity planning application Database (CAPACITY).

DATA ENTRY FACILITY (SAS/FSP)

The SAS Full Screen Product (FSP) is a tool that you can use for manual data entry. SAS/FSP provides facilities to create, browse, and edit observations contained in a SAS file. SAS/FSP is described in the SAS/FSP User's Guide, available from SAS Institute, Inc.

SAS/FSP procedures that are used for CA MICS data entry are:

```
  PROC FSEDIT, used to enter or edit data in SAS data sets.

  PROC FSBROWSE, used to display observations in a SAS data
               set.
```

CA MICS manual data entry functions are performed using formatted screens which define the data elements contained in each observation. You can use standard SAS/FSP screen formatting capabilities to tailor the screens to suit installation requirements.

OPERATIONS INFORMATION AREA MANUAL DATA ENTRY

```
-----------------------  SAS With CA MICS Libraries (MSAS) ----------------------
Option ===> 4



        0 - MICF Options
            Review/modify MICF parameters and default options.
        1 - SAS Only
            Execute SAS/DMS with just the SAS libraries and CA MICS load.
        2 - SAS with CA MICS Complex Libraries Only
            Execute SAS/DMS with CA MICS sharedprefix libraries allocated.
        3 - SAS with CA MICS Database Unit Selection
            Execute SAS/DMS with one or more unit databases allocated.
        4 - CA MICS Data Entry
            Invoke SAS for CA MICS data entry.
        T - Tutorial
        X - Exit
```

. . . .

Manual data entry facilities for the Operations Information Area files are available through MWF's SAS With CA MICS Libraries (MSAS) application. The MSAS panel (above) is displayed when you select the MSAS option from the primary CA MICS Workstation Facility (MWF) menu.

Select option 4 from the MSAS menu to display the CA MICS Data Entry (SCREENS) panel. A sample appears on the next page of this guide.

See the CA MICS Batch and Operations Analyzer Guide for more information on updating the Operations Information Area files.

```
------------------------------  CA MICS Data Entry ------------------------------

 Command ===>


  SAS/FSP Function ===> b  (B-Browse, D-Data Entry, S-Screen Modification)

  Data Source      ===> 1  (Choose from 1-6)

           Operations   1 - OPSAVL01 - IPL Outage Times and Reason Codes
                        2 - OPSCTF01 - Environmental Change Tracking
           Accounting   3 - ACTSCTcc - Standard Credit
                        4 - ACTRCT01 - Recurring Credit
                        5 - ACTSDTcc - Standard Debit
                        6 - ACTRDT01 - Recurring Debit


  Cycle Number     ===> __ (Required with options 3 and 5)



  Press ENTER to continue.  Press END to terminate.


.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

The CA MICS data entry (SCREENS) panel gives you the choice of browsing the CA MICS file, editing the CA MICS file, or modifying the formatted SAS/FSP screens. Enter B, D, or S in the SAS/FSP Function data entry field.

B - Browse permits read-only access to a file.

D - Data Entry permits you to add observations or update existing observations.

S - Screen Modification permits you to modify the data entry or browse screen format.

Enter 1 (IPL Outage Times and Reason Codes) or 2 (Environmental Change Tracking) in the Data Source data entry field.

When you press ENTER, CA MICS will invoke SAS/FSP and begin Operations Information Area data entry.

If you specified 1 (IPL Outage and Reason Codes) for Data Source, you will also need to select the CA MICS unit

Database (DB) ID for which you want to enter IPL outage and reason code information. The MICF Execution-Time Parameters panel is displayed for this purpose. A sample appears on the next page of this guide. If you are uncertain which unit databases contain the SMF product, enter a ? (question mark) for Database ID and MICF will display a selection list of the just those unit databases that contain the SMF product.

If you specified S (Screen Modification) for SAS/FSP Function, refer to the SAS/FSP manual for details on defining and modifying SAS/FSP screen definitions. Be sure to make backup copies of your screen changes as future CA MICS maintenance may update the Operations Information Area SAS/FSP screen definitions.

After you complete Operations Information Area data entry and press END on the data entry panel, the SAS Display Manager (SAS/DMS) panel will be displayed. Use the BYE command to terminate SAS/DMS and redisplay the CA MICS Data Entry panel.

```
-------------------------  Execution-Time Parameters  -------------------------

 Command ===>

Press END to execute the inquiry.  Enter CANCEL to cancel inquiry execution.
Executing CA MICS Inquiry: MSASAV - MSAS Data Entry - OPSAVL File


-------------------------------------------------------------------------------
  Enter one or more Database IDs for the Database(s) you want to process.
?____________________________________ - _____________________________________

******************************* BOTTOM OF DATA ********************************




.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

```
----------------------------  Database Selection  -----------------------------

Command ===> s p                                               Scroll ===> CSR

Executing CA MICS Inquiry: MSASAV - MSAS Data Entry - OPSAVL File

Line Cmds: S Select

     -Database-
Cmd  ID  Label                    Database Title (long name)
 -   -- --------  ------------------------------------------------------------
 _   P  PRIMARY   CA MICS PRIMARY DATABASE
 _   R  REMOTE    REMOTE DATA CENTER
******************************* BOTTOM OF DATA ********************************


.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

4.4.6 Software Maintenance

The Product Support Program (PSP) process provides an interactive panel-driven application for reviewing, preparing, and applying product changes to the CA MICS system. It provides facilities for reviewing outstanding problems, reviewing available product changes, applying product changes, and reviewing the status of your CA MICS system maintenance. See the How to Use PSP Guide for instructions.

While you should be able to apply CA MICS maintenance and handle most parameter changes without assistance, the following maintenance situations require advice from CA Technical Support at http://ca.com/support.

- Changing the value of sharedprefix

- Changing the name of sharedprefix.MICS.ISPTLIB

- Changing ISPF versions

- Changing the name of user-level data sets allocated by MWF dialogs

4.4.7 Adding Your Load Libraries to CA MICS

The following checklist enables you to concatenate your load libraries either before (STEPLIB1) or after (STEPLIB2) the CA MICS and SAS load libraries. This checklist is optional.

The ability to add load libraries to CA MICS is helpful if your site segregates software product load libraries by product or vendor.

__ 1. Create a load library that contains your load modules.

__ 2. Edit sharedprefix.MICS.PARMS(JCLDEFC) and add a line that identifies your fully-qualified load library name.

If you want your load library concatenated before the CA MICS and SAS load libraries, it should look like this:

```
STEPLIB1 'your_load_library'
```

If you want your load library concatenated after the CA MICS and SAS load libraries, it should look like this:

```
STEPLIB2 'your_load_library'
```

We recommend you use STEPLIB2 whenever possible to lessen the impact to CA MICS.

__ 3. Edit sharedprefix.MICS.PARMS(JCLGENUC). Ensure it has one line that reads:

```
ALLPROCC
```

__ 4. Submit sharedprefix.MICS.CNTL(JCLGENUC) and ensure it completes with a condition code of zero.

__ 5. Submit sharedprefix.MICS.CNTL(PROCJOB1) and ensure it it completes with a condition code of zero.

The MICSCOM, MICSDM, MICSDOC, MICSGT, MICSGU, and MICSRPTS PROCs are now updated to use your load library.

```
*************************************************************
*  Repeat Steps 6 through 10 for each unit database that    *
*  will execute your load modules.                          *
```

```
************************************************************

__  6. Edit prefix.MICS.PARMS(JCLDEF) and add a line that
       identifies your fully-qualified load library name.

       Enter the same change you made to JCLDEFC, either
       STEPLIB1 or STEPLIB2.

__  7. Edit prefix.MICS.PARMS(JCLGENU).  Ensure it has one
       line that reads:

           ALLPROCU RESTORE

__  8. Submit prefix.MICS.CNTL(JCLGENU) and ensure it
       completes with a condition code of zero.

__  9. Submit prefix.MICS.CNTL(PROCJOB2) and ensure it
       completes with a condition code of zero.

       PROCJOB2 moves the MICSNDBx and MICSDUx JCL procedures
       containing a reference to your load library to your
       installation's PROCLIB.

__ 10. Submit prefix.MICS.CNTL(PROCJOB3) and ensure it
       completes with a condition code of zero.

       PROCJOB3 moves the MICSDBx and MICSSHRx JCL procedures
       containing a reference to your load library to your
       installation's PROCLIB.

************************************************************
*  Repeat Steps 6 through 10 for each unit database that    *
*  will execute your load modules.                          *
************************************************************
```

4.4.8 CA MICS Administration in SMS-Controlled Environment

This section provides guidance for administering CA MICS in an SMS environment. We recommend that you share this information with your site's SMS or Data Administrator.

The following topics are covered:

```
    1 - SMS Allocation of SAS Data Sets
    2 - CA MICS Tape Mount Management
    3 - Data Set Naming Conventions
    4 - Identifying CA MICS and SAS in SMS ACS Routines
    5 - System-Determined Block Size
    6 - Empty Data Sets
      7 - Migrating CA MICS Complex-Level Data Sets
```

4.4.8.1 SMS Allocation of SAS Data Sets

SAS data sets are allocated and appear to OS/390 and, consequently, to DFSMS as sequential data sets. These data sets have a proprietary internal construction and require a proprietary SAS access method to be read or written at the record level. They are not, in fact, sequential data sets.

SAS supports multivolume allocations with restrictions in the following areas:

- SAS preformats the new data library with empty blocks. This practice makes multivolume allocation for the temporary WORK data library and others impractical because SAS will repeat this preformat operation in each job step. Space for a multivolume file can be preallocated, or it can be JCL-allocated through either a Storage Class with the GUARANTEED SPACE option or a Data Class with a VOLUME COUNT greater than one. We recommend the latter solution.

- Secondary allocations on additional volumes are not honored. The primary allocation is made and then SAS moves on to the next volume. This is true of all volumes except the first, where you will get the primary and secondary allocations as usual. Because SAS does not support secondary extents for multivolume data sets, you must manage physical reallocation of the databases. The frequency with which you must do this will vary with use. Because SAS does not acquire secondary space for multivolume data sets, you will have large amounts of unused DASD locked up.

- There is no protection or flagging in SAS for data sets that get out of sync when a volume restore is done on one of the units where the data library resides. Such a situation can cause unpredictable errors that are difficult to diagnose.

Because of the restrictions on OS/390 multivolume allocation imposed by SAS at this time, data sets using an SMS Data Class that allocates two or more units can cause SAS processing to fail.

SAS libraries are allocated in different ways depending on usage and operating environment. Because of the differences in allocation, these can be assigned different Data Classes.

- The CA MICS Unit Database, CA MICS Accounting and

Chargeback files, CA MICS Capacity Planner files, and CIMANAGE files are permanently allocated libraries.

- All SAS invocations require a WORK file; this is usually a temporarily allocated library. Use the WORKSMSPARMS parameter in sharedprefix.MICS.PARMS(JCLDEFC) and prefix.MICS.PARMS(JCLDEF) to specify STORCLAS, MGMTCLAS, DATACLAS, etc. for WORK file allocations.

- All CA MICS cataloged procedures allocate temporary sort work data sets (SORTWK01-SORTWK06). Use the SORTSMSPARMS parameter in sharedprefix.MICS.PARMS(JCLDEFC) and prefix.MICS.PARMS(JCLDEF) to specify STORCLAS, MGMTCLAS, DATACLAS, etc. for sort work data set allocations.

- CA MICS components enabled to standardized multiple work file support allocate additional WORKnn DDs. Data set allocation parameters are specified in the product's cccOPS member WORK parameter. This parameter supports the SMS STORCLAS, MGMTCLAS, and DATACLAS parameters in addition to UNIT and VOLSER. See the product guides for more information. Refer also to Section 2.3.1.6, Database Complex JCL Definitions, for an example.

- CA MICS components enabled to internal step restart support dynamically allocate and catalog the restart control data set and multiple intermediate SAS work file data sets. Data sets are allocated and cataloged at the beginning of the database update job step, and are retained if the step abends. The data sets are deleted after the database update completes successfully. Data set allocation parameters, including the SMS STORCLAS, MGMTCLAS, and DATACLAS parameters, are specified in the product's cccOPS member. See the product guides for more information.

- Several SAS data libraries are dynamically allocated by the CA MICS Information Center Facility (MICF). These include data libraries allocated to the following ddnames:

  - BICFUSER
  - CAPACITY
  - PMWFSAS
  - SASPROF
  - TEMPSAS
  - USER
  - WORK

Use the CA MICS Administrator Facility Options dialog to specify allocation parameters for these data sets (see Section 4.4.2.4). MICF users can override the shared defaults allocation parameters through the MICF options dialog. Both options dialogs support SMS parameters in the Additional Allocation Parameters free-form data entry field.

The following SAS Usage Notes describe the issues and circumventions available for manipulating SAS Libraries in an SMS-controlled environment:

- o V6-SYS.SASIO-2013
- o V6-SYS.SASIO-2645
- o V6-SYS.SASIO-4635
- o V6-SYS.SASIO-5674
- o V6-SYS.SASIO-7929
- o V6-SYS.SASIO-7930
- o V6-SYS.SASIO-C279

4.4.8.2 CA MICS Tape Mount Management

CA MICS generates some JCL that uses the UNIT=AFF parameter to stack multiple data sets on the same tape and minimize tape mounts during database backup and restore. Some installations use SMS ACS routines to redirect some tape data sets to DASD. This can cause failures when trying to run the CA MICS jobs that use UNIT=AFF. CA MICS jobs that use the UNIT=AFF parameter are the BACKUP/RESTORE jobs, the WEEKLY job, the MONTHLY job, and MONTHACT.

The problem occurs when one or more of the data sets (generally, the smaller data sets) are redirected to DASD, but the larger data sets are written to tape. In addition, semi-stacking interferes with the file sequence and causes LABEL=(n+1,SL) clauses to fail.

The solution is to direct all of the tape data sets to tape. You may need to change your ACS routines so that these data sets will be written to tape.

Refer to IBM APARs II06908 and OY53625 for more information. See also SAS Usage Note V6-SYS.SASIO-D302.

4.4.8.3 Data Set Naming Conventions

Most ACS routines rely heavily on the data set name when making allocation decisions. This should work well in the CA MICS environment, because CA MICS generally provides adequate flexibility for most data set naming requirements.

Problems can arise when a CA MICS data set name conflicts with another standard already in place. For instance, your Data Class ACS routine may force all *.CLIST data sets to have RECFM=VB and LRECL=255. This would conflict with the attributes expected for the CA MICS CLIST data set, and could cause failures or unexpected results.

You can avoid this type of error by having separate rules for CA MICS data sets (see Section 4.4.8.4, Identifying CA MICS and SAS in SMS ACS Routines), or by avoiding ACS routine testing that is too general.

Refer to Sections 2.3.3.3 and 2.3.3.3.2.3 for information about using JCLNAMES to override default data set names.

4.4.8.4 Identifying CA MICS and SAS in SMS ACS Routines

CA MICS provides a flexible approach to file naming. It will accommodate most data set naming conventions.

After the CA MICS convention is specified at your site, those names are available for query by DFSMS ACS routines. In most cases, a short FILTLIST in an ACS routine can recognize any CA MICS file name. Decisions about tape residency, enforcement of system-determined block size, and so on, can then be based on that FILTLIST.

You can choose to modify your ACS routines to identify the program name under which allocation of CA MICS and SAS data sets occur, so that the allocation of these types of data sets will be handled correctly. However, there may be difficulties with implementing this approach at your site. For example, IKJEFT01 is the batch-TMP interface program provided by ISPF. Applications other than CA MICS can use this program. In addition, other applications besides CA MICS may use SAS. This is why we recommend the use of data set name identification over program name identification. If you decide to identify program names in your ACS routines, test for the following program names:

- For CA MICS:
    - MACTL
    - IKJEFT01
- For SAS:
    - SASHOST
    - SASXAL
    - SASCP

4.4.8.5 System-Determined Block Size

System-determined block size can be a useful feature in providing site-SMS conformance. Using it must be moderated by two considerations:

- o Access method
- o Intent of use

System-determined block size generally assumes that standard access methods are employed according to DSORG. Clearly, this is not always the case. SAS data libraries are a good example. As mentioned in Section 4.4.8.1, SMS Allocation of SAS Data Sets, SAS data libraries are allocated with all the external characteristics of a sequential data set. However their internal format is proprietary and cannot be read or written properly at the record level using standard access methods.

System-determined block size is designed to optimize sequential access of large (that is, multi-track) files. Its "one size fits all" implementation does not tailor blocking or buffering for performance efficiency. SAS FORMAT catalogs provide a good example of this. Because DSORG=PS is specified for these data sets, you may assume that half-track blocking is the best option. But these data sets don't contain data, they contain executables. Very large buffers will be allocated and filled while the actual executable that was requested is likely to be fairly small. Examination of the effect half-track blocking has on processing, in this case, demonstrates that this choice causes very inefficient memory use and increased SRB time.

The default forced block sizes delivered with the CA MICS distribution libraries are already set for optimal operation.

See SAS Usage Notes V6-SYS.FILE-6849 and VS-SYS.FILE-7901 for more information.

4.4.8.6 Empty Data Sets

CA MICS normal operation includes allocation of empty, or nearly empty, files that are sometimes spotted by a Storage Administrator or software storage management agent (such as the SMS Management Class Space Release function). The storage manager's standard response is to delete, reblock, or release the "over-allocated" space. However, CA MICS has preallocated that space for a reason (for example, to hold month-to-date accounting information). Arbitrary space release will probably cause a later CA MICS run to fail.

The best defense is a good naming convention that can be recognized and honored by space management tools.

CA MICS provides the SYSTEMMODEL parameter for both the complex level (sharedprefix.MICS.PARMS(JCLDEFC)) and the unit level (prefix.MICS.PARMS(JCLDEF)) to allow users to specify a system-wide data set control block (DSCB). This DSCB is then used by default during allocation, so that SMS or other utilities do not intervene.

If NOMODEL is used as the SYSTEMMODEL model dataset name, the use of the model dataset in the DCB specification will be bypassed.

Refer to Sections 2.3.1.6.4 and 2.3.3.2.1.5 for more information about the SYSTEMMODEL parameter.

4.4.8.7 Migrating CA MICS Complex-Level Data Sets

DFSMS Management Class controls the migration of data sets from primary DASD, as well as the copying of data sets with the SMS-driven backup facility. CA MICS files that are not used frequently are good candidates for migration. The following is a list of files to consider for migration depending on how often they are used at your site:

- sharedprefix.MICS.ASM
- sharedprefix.MICS.BIN
- sharedprefix.MICS.GENLIB
- sharedprefix.MICS.HOLD.PARMS
- sharedprefix.MICS.HOLD.USER.SOURCE
- sharedprefix.MICS.LOCALMOD.CNTL
- sharedprefix.MICS.MACLIB
- sharedprefix.MICS.OBJ
- sharedprefix.MICS.PSP.*

4.5 Integrating CA MICS Data to NSM

This section describes the integration of CA MICS data to NSM. Sections 4.6 and 4.7 deal with the integration of NSM data to CA MICS.

NSM provides end-to-end enterprise management encompassing areas such as security management, storage management, service/help desk, output management, performance and accounting, network management, database and application management, and workload management.

CA MICS integrates with NSM performance and accounting by providing NSM data extract interfaces and a data transfer utility. The support allows CA MICS data to be extracted in specific CSV (Comma Separated Value) formats that subsequently can be processed by NSM Performance Trend or NSM Chargeback.

The following topics are addressed in this section:

  1 - Integration with NSM Chargeback
  2 - Integration with NSM Performance Trend
  3 - Data Conversion and Transfer

4.5.1 Integration with NSM Chargeback

NSM Chargeback provides a chargeback and resource utilization tracking component that records resource consumption by user for each system utilized. It maintains budget information and can be used to generate invoices or charge reports.

The support for NSM Chargeback allows CA MICS data to be charged through NSM Chargeback as an alternative to CA MICS Accounting and Chargeback. The support is provided through a two step process: an inquiry, TNGAD1, and a consolidation utility, TNGCBCSV. These two programs summarize the CA MICS resource data and create a CSV file in the format required by NSM Chargeback.

The TNGAD1 inquiry is meant to be run on a daily basis, although it can be run less frequently if desired. It extracts the requested data from the CA MICS database, summarizes it by userid and date, and saves it in complex-level SAS data libraries.

The consolidation utility, TNGCBCSV, reads the existing SAS data sets, summarizes them by userid, and creates the CSV files for NSM Chargeback. The utility then deletes the SAS data sets. A TODATE parameter allows the user to include only data up to a certain date.

The TNGAD1 and TNGCBCSV programs work together to ensure that all data for a day is included, even if it is extracted from different units or on different days, and that data that has been summarized into a CSV will not get used again. Please note, however, that the TNGAD1 inquiry does NO DUPLICATE CHECKING. It adds the data it extracts from CA MICS to any data it already has in its SAS databases for that day.
If duplicate data exists in the CA MICS database prior to summarization into new SAS data sets for consolidation, the duplicate data will be read in. It is up to you to make sure that data is not read in twice.

Details on using the inquiry and the utility job are described in the sections that follow:

o The TNGAD1 Inquiry
o TNGCBCSV - The Consolidation Utility

4.5.1.1 The TNGAD1 Inquiry

The TNGAD1 inquiry collects resource elements from a CA MICS unit database and saves them in complex-level SAS data libraries. The inquiry gets the list of elements to be extracted from prefix.MICS.PARMS(TNGCBDEF).

Note: The inquiry does NO DUPLICATE DATA CHECKING. Existing data sets are updated with any requested data that has the same month and day.

- Specifying the unit: The unit identifier is a runtime parameter.

- Specifying the files and elements: Specify the elements to be extracted in prefix.MICS.PARMS(TNGCBDEF). TNGCBDEF is read each time the inquiry is run, so changes to this member take effect automatically. NSM Chargeback recommends that a maximum of 10 elements be extracted per complex.

  The following statements are supported:

  ```
  PRIME z1 <z2 ... z9 >
  TIMESPAN ts
  FILE iiifffcc
  USERID element
  BYSHIFT element <element ... >
  TOTAL element <element ... >
  ```

  The TIMESPAN, FILE, USERID, BYSHIFT and TOTAL statements form a group defining the elements to be extracted from a particular file. Multiple groups may be defined, but all the statements referring to one file must be together.

  PRIME Specifies the zones that define your prime shift. NSM Chargeback allows you to charge different rates for prime and nonprime shifts. Any zones listed in this statement will be considered prime shift. All other zones will be considered nonprime shift. This keyword is only required if you choose to split metrics by shift using the BYSHIFT statement described below. Only one PRIME statement is allowed. Up to 9 zones can be specified, separated by blanks.

  TIMESPAN Specifies the timespan of the file. DETAIL and DAYS are supported. One TIMESPAN statement is

```
                 required and it must precede the first FILE
                 statement.  Additional TIMESPAN statements may
                 precede additional FILE statements.  If only
                 one TIMESPAN statement is provided, the same
                 timespan will be used for all requested files.

      FILE       Specifies the file from which the elements are
                 to be extracted.  It INCLUDES cycle number.
                 One FILE statement is required.  There is no
                 limit on the number of FILE statements
                 included.

      USERID     Specifies the element in the file that should
                 be assigned to USERID.  All charges in NSM
                 Chargeback are summed by SYSID and USERID.

      BYSHIFT    Specifies the elements in the file that should
                 be summed for prime and nonprime usage.
                 Separate elements by blanks.  Use additional
                 BYSHIFT statements if necessary.

      TOTAL      Specifies the elements in the file that should
                 be summed for their total usage, without
                 maintaining a distinction between prime and
                 nonprime use.  Separate elements by blanks.
                 Use additional TOTAL statements if necessary.
```

o Specifying environmental parameters: Review sharedprefix.MICS.SOURCE(BASUCBXT) for a description of the parameters available to define DASD requirements and other environmental details. The parameters are implemented via SAS macro variables and can be overridden as needed.

o Output Data Set Naming Conventions: The data sets created by the TNGAD1 inquiry are named as follows:

```
   Data Library:  sharedprefix.MICS.TNGCBSAS.YRyyyy.@ssss
       Data Set:  MmmDdd

   where:
         yyyy   Is a four digit number that denotes the year
                of the data.

         ssss   Is a four character system identifier
                (SYSID).

         mm     Is the month of the data.
```

```
      dd     Is the day of the data.
```

One data library is created per system, per year. Within each data library, there is one data set per day. Data is added to the data set for that day, if one exists. THERE IS NO CHECK FOR DUPLICATE DATA.

4.5.1.2 TNGCBCSV - The Consolidation Utility

Sharedprefix.MICS.CNTL(TNGCBCSV) creates the CSV files for NSM Chargeback from the SAS data sets created by the TNGAD1 inquiry described in the previous section.

- Specifying the SAS data libraries to include in the consolidation: Edit sharedprefix.MICS.CNTL(TNGCBCSV). After the DSNLIST DD, list the SAS data libraries that the consolidation utility should include in the CSVs. You do not have to run the utility for each SYSID separately, the program will parse the data set names and create one data set for each SYSID and year.

- Specifying a date limit: By default, TNGCBCSV will sum together all the data sets for a SYSID. You may, however, specify a date limit using the TODATE parameter. If the TODATE parameter is specified, only data sets for dates up to and including the TODATE are summed together. You could use this to create a CSV only containing data for the previous month, for example.

  The TODATE parameter is specified in the SYSPARM parameter on the EXEC statement, in mmddyyyy format.

- Specifying environmental parameters: Review sharedprefix.MICS.SOURCE(BASUCBXT) for a description of the parameters available to define DASD requirements and other environmental details. The parameters are implemented via SAS macro variables and can be overridden as needed.

- Output Data Set Naming Conventions: The data sets created by TNGCBCSV are named as follows:

  ```
  Data Set:  sharedprefix.MICS.TNGCB.YRyyyy.@ssss
    Member:  DUmmddCB
  ```

  where:

  yyyy   Is a four digit number denoting the year of the run date or the year of TODATE.

  ssss   Is a four character system identifier (SYSID).

  mm   Is the month of the data.

  dd   Is the day of the data.

One data set is created per system, per year. Within each data set there is one member per day.

Note that the TODATE parameter affects the name of the data set. If the TODATE parameter is not used, the year, month and day used in the data set and member name will be derived from the run date. If the TODATE parameter is used, the year, month and day will be derived from TODATE.

4.5.2 Integration with NSM Performance Trend

NSM Performance Trend supports system management and/or capacity planning. It provides historical performance and usage information in a spreadsheet where it can be manipulated and refined. Mainframe performance information collected via CA MICS can be made accessible for use in NSM Performance Trend.

The support for NSM Performance Trend extracts CA MICS data from the CA MICS database in a CSV (Comma Separated Value) format that can be processed by NSM Performance Trend.

To enable a CA MICS component to NSM Performance Trend, you need to create a MICF inquiry and a member in sharedprefix.MICS.PARMS defining what data should be extracted from the component.

For more details, see the following sections:

1 - Parameter Specification
2 - Performance Trend Inquiry
3 - Output Data Set Naming Convention
4 - User Exits

4.5.2.1 Parameter Specification

To enable a CA MICS component to NSM Performance Trend, you must first add a member in sharedprefix.MICS.PARMS defining what data should be extracted from a CA MICS component:

The following statements are supported:

```
COMP ccc
FILE iiifff <file description>
SASDS saswork
NAME element <element label>
DEPADD element element ...
WORKLOAD element element ...
```

Note: All statements are required unless explicitly labeled as optional.

COMP Specifies the three-letter CA MICS component identifier from which performance data will be extracted. It must be the first statement in the member and can only be specified once. For example, RMF.

FILE Specifies the six-character CA MICS file from which to extract output data, e.g., HARCPU. This file must pertain to the component specified in the COMP statement. Optionally, following the CA MICS file name, you can specify the file description. The description will appear in the Resource Type column in the output. If omitted, the three-letter component identifier followed by the file description defined in cccGENIN will be used.

SASDS Specifies a valid one- to eight-character SAS work data set name. The data set name is assigned using the "CA MICS File Selection" panel in MICF during MICF inquiry creation (Step 2.3).

NAME Specifies a CA MICS element name. The element will be extracted from the SAS data set defined by the SASDS statement. Optionally, you can specify a label for the element following the element name. The label will appear in the Resource Subtype column in the output. If omitted, the label defined for the element in cccGENIN will be used. You can specify one or more NAME statements.

DEPADD (Optional) Specifies one or more dependent elements that were erroneously excluded from cccGENIN for a computed element selected by the NAME statement. You may specify one or more DEPADD statements. Note that this statement should ONLY be used to circumvent the problem when cccGENIN does not list all the dependent elements in the DEPEND statement(s) for the elements that you want to retrieve.

WORKLOAD (Optional) Specifies one or more CA MICS elements that will be used to group the CA MICS data into categories for each unique combination of values from the elements. The values of the WORKLOAD elements will appear in the Resource Instance column in the output file.

* Comments. Lines beginning with asterisks (*) are ignored.

For Example:

```
   COMP RMF
   *
   FILE HARCPU   MVS CPU Activity
   SASDS FILE1
   NAME CPUPCBSY Pct Processor Busy
   NAME CPUAVB   Avg Batch Users
   NAME CPUAVT   Avg TSO Users
   NAME CPUAVS   Avg Started Tasks
   NAME CPUSLIH  Total # of Entries to the I/O SLIH
   NAME CPUTPI   Total # of TPI I/O Events
   *
   FILE SCPPSD   MVS Page/Swap Data Set Activity
   SASDS FILE2
   NAME PSDAVU   Avg Used Slots or Swap Sets
   NAME PSDAVA   Avg Allocated Slots
   NAME PSDPGX   Pages Transferred
   NAME PSDAVSTM Avg Service Time
   WORKLOAD PSDPST
```

Note: When a Performance Trend inquiry runs for the first time, SAS code is generated and stored in a member in sharedprefix.MICS.SOURCE based on the associated parameter file in sharedprefix.MICS.PARMS. The member name in sharedprefix.MICS.SOURCE is the same as the parameter file. The code in SOURCE will be directly invoked by the subsequent runs if it has been built.

The CA MICS data extract program will not regenerate the code again for efficiency reasons. Therefore, if you make a change to a parameter file, you MUST delete the corresponding member in sharedprefix.MICS.SOURCE for the change to take effect.

4.5.2.2 Performance Trend Inquiry

To create a Performance Trend inquiry, complete the following checklist:

____ 1. Add a new member in sharedprefix.MICS.PARMS to specify what to extract. See "Parameter Specification" for details.

____ 2. Add a MICF inquiry through the MICF Inquiries panel of the CA MICS Workstation Facility (MWF): (Or you can copy an existing Performance Trend inquiry such as RMFTD1 or RMFTD2 and modify it.)

a. Choose "Structured Inquiry".

b. Go to "Data Manipulation", "External File Facilities", and choose "External Program Invocation". Enter the following:

```
Program Name            ==> BASUCPXT
CA MICS Library DD NAME ==> SOURCE
```

Press END to save. Press END again until you are back to "Structured Inquiry".

c. Choose "CA MICS File Selection". Select the CA MICS file, timespan, and optionally, cycle(s) and database. Enter the output file name. It must be the same as the one defined in Step 1. Press END to save.

d. Go to "Data Manipulation", "Advanced Data Manipulation", and choose "SAS System Statement". Enter the following statement:

```
%SELECT;
```

Press END to save. Press END again to return to the "Data Manipulation" panel.

e. Select "Data Element Derivation" and create a new element, DATE, with the following attributes for the file defined in Step 2.3:

```
Element Name  ===> DATE
       Label  ===> Date
Element Type  ===> D
Value Length  ===> 7
```

```
Output format ===> DATE8.
Element Derivation:
 ===> DATE = MDY(MONTH,DAY,%YEAR4);
```

Press END to save.

f. Select "Advanced Data Manipulation" and choose "Independent Source Statements". Create a macro %TNGGEN as follows at the "Independent Source Statements" panel:

```
%MACRO TNGGEN;
   %LET UCPIPARM = xxxxxx;
   %INCLUDE SOURCE(BASUCP)/NOSOURCE2;
%MEND TNGGEN;
%TNGGEN;
```

where xxxxxx is the member name you defined in Step 1.

Press END to save.

g. Save the inquiry.

You are now ready to run the inquiry.

4.5.2.3 Output Data Set Naming Convention

Performance Trend inquiries produce three types of output:

1. Daily — Identified by a member name ending with the letters "DL". Daily data contains one day's worth of Performance Trend data from a system.

2. Period — Identified by a member name ending with the letters "WK". Period data contains one month's worth of Performance Trend data per system.

3. Enterprise — Identified by a member name ending with letters "CC". Enterprise data consolidates one day's worth of Performance Trend data from all the systems that are enabled for data extract in a CA MICS complex.

The data sets created by Performance Trend inquiries are named as follows:

```
1.  Data Set:  sharedprefix.MICS.TNGPT.YRyyyy.@ssss
    Member:    STmmddDL
               STmm01WK

    where,
           yyyy   Is a four-digit number that identifies the
                  year the data was generated.

           ssss   Is a four-character system identifier
                  (SYSID).

           mmdd   Is the month and day the data was created,
                  e.g., 0926.

           mm     Is the month the data was created.
```

One data set is created per system, per year. Within each data set, there is one member per date in a year for daily data and one member per month in a year for period data.

```
2.  Data Set:  sharedprefix.MICS.TNGPT.YRyyyy
    Member:    STmmddCC

    where,
           yyyy   Is a four-digit number that denotes the
```

```
                year the data originated.

         mmdd   Is the month and day the data was created,
                e.g., 0926.

One data set is created per year.  Each data set contains
enterprise data per day in a year.
```

4.5.2.4 User Exits

User exits can be defined in sharedprefix.MICS.SOURCE(BASUCPXT). The user exits must be written in SAS macro language. The following SAS macro variables and macros are available for providing site-specific definitions:

```
o  %LET ALUNIT = ________;
   ==> Macro variable ALUNIT defines the unit name for
       Performance Trend data set allocation.  Default is
       SYSDA.

o  %LET SPACE = (unit,(primary,secondary,directory));
   ==> Macro variable SPACE defines the space allocation to
       use for the Performance Trend CSV data sets.
       Default is (TRK,(5,5,5)).  Note that a CSV data set
       must be a partitioned data set (PDS).

o  %LET IDXSPACE = (unit,(primary,secondary));
   ==> Macro variable IDXSPACE defines the space allocation
       to use for the Performance Trend index data set.
       Default is (TRK,(5,5)).  Note that the index data set
       must be a sequential data set.

o  %LET DASDUNIT = ________;
   ==> Macro variable DASDUNIT defines the unit name for work
       data sets.  If not specified, the SAS system default
       is used.

o  %LET DASDSPACE = (unit,(primary,secondary,directory));
   ==> Macro variable DASDSPACE defines the space allocation
       to be used for work data sets.  If not specified, the
       SAS system default is used.

o  %LET TRACE = ________;
   ==> Macro variable TRACE triggers debugging option.  Valid
       values are ON, OFF.  Default is OFF.

o  %LET UCPREFIX = ______________________;
   ==> Macro variable UCPREFIX overrides the default prefix
       of Performance Trend data set names.  Default is
       sharedprefix.MICS.TNGPT.

o  %LET CREATCC = ________;
   ==> Macro variable CREATCC controls creation of the
       Enterprise CSVs (STmmddCC members).  Valid values are
       YES and NO.  The default is YES.
```

o %SELECT Macro

==> The SAS macro, %SELECT, can be used to subset the input data at run time. For example,

```
%MACRO SELECT;
  %IF .&INQINAME EQ .RMFTD1 %THEN %DO;
    IF SYSID EQ 'S008' OR SYSID EQ 'S002';
  %END;
  %ELSE %IF .&INQINAME EQ .RMFTD2 %THEN %DO;
    IF SYSID EQ 'A192';
  %END;
%MEND SELECT;
```

In this example, only data from system S008 and S002 will be processed by Performance Trend Inquiry RMFTD1, and only data from system A192 will be processed by Inquiry RMFTD2.

4.5.3 Data Conversion and Transfer

After CSV data for NSM Performance Trend or NSM Chargeback have been extracted from the CA MICS database, they need to be downloaded to the NSM machine where NSM Systems Performance is installed. Furthermore, CSV data for Performance Trend must be converted into the Performance Cube format before they can be processed.

The RMWSXFR program supports CSV data conversion and data transfer. To install a copy of RMWSXFR on your workstation, complete the following checklist:

___ 1. Download TNGXFRZ from sharedprefix.MICS.BIN.

To use FTP (File Transfer Protocol) to transfer TNGXFRZ to your Microsoft Windows Workstation, perform the following steps in order:

___ a. Open a DOS command window.

___ b. Start the FTP program using the following:

```
FTP <hostname>
```

where <hostname> is the machine name or the IP address of the OS/390 system.

Supply the host userid and password as prompted.

___ c. Set binary transfer mode by entering the following command (in lower case):

```
binary
```

```
*****************************************************
*  NOTE:  This FTP command and those to follow      *
*         must be entered in lower case.            *
*****************************************************
```

___ d. Set the name of the host data set with the following:

```
cd 'sharedprefix.MICS.BIN'
```

___ e. Set the directory name of the workstation directory to hold TNGXFRZ with the following command:

```
lcd <directory>
```

where <directory> is a directory on the workstation which will contain TNGXFRZ. Generally, this is a temporary or working directory (e.g. C:\TEMP).

___ f. Transfer the TNGXFRZ to the workstation with the following command:

```
get TNGXFRZ RMWSXFRZ.EXE
```

This will copy TNGXFRZ to the directory specified in step e and rename it to RMWSXFRZ.EXE.

___ g. Once the transfer is complete, exit FTP by entering:

```
quit
```

While this example uses FTP to perform the actual transfer, other utilities (such as IND$FILE) can be used. Whichever program will be used, it must be able to transfer from the host to the workstation without performing any character translation (i.e. EBCDIC to ASCII).

___ 2. Run RMWSXFRZ.EXE. This starts the installation process. You will be prompted for a destination folder where the program files will be installed.

___ 3. After the installation is complete, read README.TXT in the program directory for detailed instructions on how to use RMWSXFR.

4.6 Integrating NSM Data to CA MICS

NSM provides end-to-end enterprise management encompassing areas such as security management, storage management, service/help desk, output management, performance and accounting, network management, database and application management, and workload management. If you are not a NSM client but would like to learn more, visit the CA website at www.ca.com for more information, including how to reach a NSM representative.

CA MICS processes data from:

o NSM Performance Trend

  NSM Performance Trend is one of a suite of NSM Systems Performance graphical applications that collect performance data from your distributed systems and present it in a spreadsheet format, making it possible to see trends that have occurred over long periods of time and to compare one machine with another.

  The processing of NSM Performance Trend data is supported by MTI (CA MICS NSM Integration). Chapter 8 of the System Modification Guide describes MTI in detail and provides instructions for modifying your installation to implement this powerful utility.

o NT Resource Accounting

  NT Resource Accounting is a service of the NSM Chargeback component of NSM Systems Performance. The service is also called NSM Resource Accounting Data Collection Service for NT. As its name suggests, NT Resource Accounting collects resource accounting data from your NT servers.

  For more information on how CA MICS processes NT Resource Accounting data, refer to Chapter 4 of the CA MICS Accounting and Chargeback Guide.

o NSM Asset Management Option (AMO)

  The NSM Asset Management Option provides hardware and software inventory functions, configuration control functions, software metering functions, and more for various platforms including DOS, Windows 3.1, Windows 95, Windows NT, OS/2, Macintosh, UNIX, NetWare, and VMS.

  For more information on how CA MICS processes NSM AMO data, refer to Chapter 4 of the CA MICS Accounting and Chargeback Guide.

CA MICS Data Transfer is the utility which facilitates the transfer and integration of data from NSM data into CA MICS. Instructions for installing this utility are provided in the Section 4.7 of this manual.

```
+--------------------------------------------------------+
| References:  Section 4.7, CA MICS Data Transfer        |
|              System Modification Guide                 |
|                Chapter 8, CA MICS NSM                  |
|                Integration (MTI)                       |
|              CA MICS Accounting and Chargeback Guide   |
+--------------------------------------------------------+
```

4.7 CA MICS Data Transfer (MICSXFR)

MICSXFR is a CA-supplied utility that transfers NSM Performance Cube, Asset Management or NT/UNIX Resource Accounting data to a data set on the host where CA MICS resides for further processing. MICSXFR allows you to specify the data to transfer, converts it to CSV format (if required), consolidates the data, and transfers it to the CA MICS host using FTP. Once the data resides on the host, it is available to CA MICS for further processing by MTI.

MICSXFR requires:

- Pentium class PC capable of running NSM

- Microsoft Windows NT 4.0 or Windows 2000

- 8M of hard disk space

- TCP/IP and FTP

- NSM Version 2.1 or later with NSM Systems Performance, NT Resource Accounting, or NSM Asset Management installed

To install a copy of MICSXFR on your workstation, complete the following checklist:

___ 1. Download MICSXFRZ from sharedprefix.MICS.BIN.

To use FTP (File Transfer Protocol) to transfer MICSXFRZ to your Microsoft Windows Workstation, perform the following steps in order:

___ a. Open a DOS command window.

___ b. Start the FTP program using the following:

```
FTP <hostname>
```

where <hostname> is the machine name or the IP address of the OS/390 system.

Supply the host userid and password as prompted.

___ c. Set binary transfer mode by entering the following command:

```
binary
```

```
*****************************************************
*  NOTE:  This FTP command and those to follow      *
*         must be entered in lower case.            *
*****************************************************
```

___ d. Set the name of the host data set with the following:

```
cd 'sharedprefix.MICS.BIN'
```

___ e. Set the directory name of the workstation directory to hold MICSXFRZ with the following command:

```
lcd <directory>
```

where <directory> is a directory on the workstation which will contain MICSXFRZ. Generally, this is a temporary or working directory (e.g. C:\TEMP).

___ f. Transfer the MICSXFRZ to the workstation with the following command:

```
get MICSXFRZ MICSXFRZ.EXE
```

This will copy MICSXFRZ to the directory specified in step e and rename it to MICSXFRZ.EXE.

___ g. Once the transfer is complete, exit FTP by entering:

```
quit
```

While this example uses FTP to perform the actual transfer, other utilities (such as IND$FILE) can be used. Whichever program will be used, it must be able to transfer from the host to the workstation without performing any character translation (i.e. EBCDIC to ASCII).

___ 2. Run MICSXFRZ.EXE. This starts the self-extracting process. You will be prompted for a location in which to store the unzipped files.

___ 3. In the directory chosen in Step 2, you will find a sub directory named MICSXFR_Install_Disk. Within this directory you will find the SETUP.EXE program. Run SETUP.EXE to install the CA MICS Data Transfer program.

___ 4. Once you have successfully installed MICSXFR, you can delete the MICSXFR_Install_Disk directory.

```
*************************************************************
*                                                           *
*  NOTE:  CA MICS Data Transfer includes an online help     *
*         system which you should consult for information   *
*         on using the program, once it is installed.       *
*                                                           *
*************************************************************
```

Additional considerations:
----

o Automatic Upload of Data to the Mainframe

Using NSM Workload Management, the transfer of NSM Systems Performance data to the OS/390 platform can be automated.

o Using a Profile

The MICSXFR utility has the ability to create profiles. A profile contains information about which machines, resources and subtypes to transmit to the OS/390 platform. When a profile is created or updated, MICSXFR also creates a batch file (profile.BAT) to perform the transfer.

The BAT file can be used to automatically schedule MICSXFR on a regular basis. NSM Workload management can be used to schedule MICSXFR. Additionally, any program that can execute a BAT file at a given time can be used to automate the transfer of data to the OS/390 host.

```
+--------------------------------------------------------+
| References:  Chapter 8, System Modification Guide,     |
|                CA MICS NSM Integration (MTI)           |
+--------------------------------------------------------+
```

4.8 CA MICS Documentation Conversion

The CA MICS Documentation Conversion utility (DOC2HTML) allows the installation to convert existing CA MICS documentation into HTML. The converted documentation can be viewed using any Web browser that conforms to the HTML-4 specification and supports Javascript.

The converted form of documentation is called the CA MICS Documentation Web View (DocWebVu) and when coupled to a Web server, allows anyone with a requisite Web browser to view the CA MICS documentation.

DocWebVu can only present guides that have been converted using DOC2HTML. This allows the installation to control which guides will be included in DocWebVu.

The following sections describe the CA MICS documentation conversion process:

1 - Prerequisites
2 - CA MICS Documentation Web Viewer HFS Space Usage
3 - CA MICS Documentation Conversion Utilities
4 - CA MICS Documentation Web View (DocWebVu)
5 - IBM HTTP Server Considerations
6 - Exporting DocWebVu
7 - Checklists

Note: Throughout this chapter $MICSROOT is used as a substitute for the name of the CA MICS Root Directory specified on the HFSROOT statement in sharedprefix.MICS.PARM(JCLDEFC).

See section 5.7 for more information about the CA MICS Root Directory definition and usage.

4.8.1 Prerequisites

In order to use DOC2HTML, you must first meet the following prerequisites:

1.) The CA MICS complex must have product change BAS7130 applied.

2.) The SAS version used by CA MICS must be Version 8 (Release 8.0) or higher. See section 5.4 for information on upgrading to a new release of SAS.

See the CA MICS Systems Administrator's Guide for information on SAS releases supported by CA MICS.

3.) The CA MICS Root Directory must have been previously created. See section 5.7 for information on defining and creating the CA MICS root directory.

4.8.2 CA MICS Documentation Web Viewer HFS Space Usage

DOC2HTML stores the DocWebVu files in a subdirectory of the CA MICS Root Directory ($MICSROOT/documents/).

DocWebVu can use as much as 400 megabytes of HFS space depending on the number and size of the guides converted.

4.8.3 CA MICS Documentation Conversion Utilities

CA MICS includes a set of utilities for generating and maintaining the DocWebVu facilities.

The following sections describe these utilities:
1 - HTML Conversion Utility (DOC2HTML)
2 - Generated Documentation Update Utility (DOCUPDTG)

An online facility allows conversion of guides and entire categories of guides. It is invoked from the CA MICS Workstation Facility (MWF) Documentation Access Print Dialog (MWF;1;3). Please refer to the online help panels for usage information.

An additional DocWebVu utility, the CA MICS Documentation Export Job (DOCXPORT) is described in Section 4.8.6.

4.8.3.1 HTML Conversion Utility (DOC2HTML)

The HTML Conversion Utility (DOC2HTML) is used to create documents in HTML format. The HTML output is called CA MICS Documentation Web View.

The job in sharedprefix.MICS.CNTL(DOC2HTML) executes the HTML Conversion Utility.

The following sections describe DOC2HTML:
  1 - JCL and Options
  2 - DOC2HTML Output
  3 - DOC2HTML Examples

4.8.3.1.1 JCL and Options

The following is an example of DOC2HTML JCL:

```
//  . . . . JOB . . . .
//HTMLCVRT EXEC MICSDOCH,
//        SYSPARM='options_string'
//
```

where options_string is the DOC2HTML processing options:

o Guide=guide_list

Specifies which CA MICS guides are to be converted to HTML. guide_list contains either guide name (e.g., PIOM) or the one or two character guide id (e.g., L).

If ALL is specified, then all guides are converted.

o CAT=category_list

Requests that all guides in specified category be printed. The two character category id can be found in the CA MICS Document Guide report as shown in section 4.8.3.1.2.

If ALL is specified, then all guides are converted.

o COMP=component_list

Specifies all CA MICS guides associated with a particular CA MICS component (e.g., ACT) be printed.

If ALL is specified, then all guides in all categories are converted.

o DIC=Y or N

Controls the conversion of CA MICS Data Dictionary entries in the HTML guides. By default, this option is set to 'N' (no conversion).

Note: Activating this option can significantly increase the amount of space required to store converted documents.

Note: Options are case sensitive.

To convert all CA MICS guides, the ALL parameter can be used on the GUIDE=, COMP=, or CAT= options.

However, DOC2HTML can be a very resource intensive operation. Converting all CA MICS guides at one time will consume a substantial amount of CPU time and may fail due to installation constraints on CPU time, output lines, or disk space.

If no processing options or guides are selected, DOC2HTML will produce the CA MICS Document Guides report as shown in section 4.8.3.1.2.

4.8.3.1.2 DOC2HTML Output

DOC2HTML produces a report listing the defined CA MICS guides. Figure 4-7 is a sample report of a DOC2HTML output.

This report can be generated without converting documents by running DOC2HTML with no guides specified.

For example:

```
  // . . . .JOB . . . .
  //HTMLCVRT EXEC MICSDOCH,
  //        SYSPARM=' '
  //
```

Note: Only the guides listed in this report are eligible for conversion.

```
                                                                                22:12 Tuesday, July 31, yyyy   1
                                                   CA MICS Document Guides      22:12 Tuesday, July 31, yyyy   1

Category                                  Volume     Guide Id   Title
----------------------------------------------------------------------------------------------------------------------------
(AC) Management Support Application       ACTC          PK      CA MICS Accounting and Chargeback Option Concepts and
Guides                                                          Overview Guide
                                          ACTU          PL      CA MICS Accounting and Chargeback Option User Guide
                                          CAP           PA      CA MICS Capacity Planner Option Guide
                                          PER           PD      CA MICS Performance Manager Option Guide
                                          STG           PG      CA MICS StorageMate Option Guide

(AD) System Administrator Guides          PIOM          L       CA MICS Planning, Installation, Operation, and Maintenance
                                                                Guide
                                          PSPH          BE      CA MICS How to Use the PSP
                                          SAG           BS      CA MICS System Administrator Guide
                                          SMG           M       CA MICS System Modification Guide

(DB) Data Integration Application Guides  AUM           FM      CA MICS Analyzer Option for MeasureWare Guide
                                          CICS          FE      CA MICS Analyzer Option for CICS Guide
                                          DB2           FN      CA MICS Analyzer Option for DB2 Guide
                                          DEX           FV      CA MICS Analyzer Option for VAX/VMS Guide
                                          IDMS          W       CA MICS Analyzer Option for CA IDMS Guide
                                          IMS           FG      CA MICS Analyzer Option for IMS Guide
                                          MQS           FO      CA MICS Analyzer Option for MQSeries Guide
                                          PWR           FQ      CA MICS Analyzer Option for VSE/Power Guide
                                          RMF           FC      CA MICS Hardware and SCP Analyzer Guide
                                          SMF           FD      CA MICS Batch and Operations Analyzer Guide
                                          SNT           C       CA MICS Network Analyzer Option Guide
                                          SRL           FH      CA MICS System Reliability Analyzer Option Guide
                                          TSO           FB      CA MICS Analyzer for TSO Guide
                                          VCA           FJ      CA MICS Space Analyzer Option Guide
                                          VM            FI      CA MICS Analyzer Option for VM/CMS Guide
                                          WEB           FW      CA MICS Web Analyzer Option Guide
                                          WEB           FW      CA MICS Tape Analyzer Option Guide

(DC) Collector Guides                     LDE           CL      CA MICS IMS Log Data Extractor Option Guide
                                          VCC           CK      CA MICS Space Collector Option Guide
                                          VMT           CM      CA MICS Data Transfer Option for VM/CMS Guide

(US) End-User Guides                      AST           AI      CA MICS CA ASTEX Option Guide
                                          RPT           H       CA MICS Standard Reports Guide
                                          DOC           JD      CA MICS Document Access Guide
                                          MREF          JR      CA MICS MICF Reference Guide
                                          MUSR          JU      CA MICS MICF User Guide
                                          SC            I       CA MICS Database Structure and Content Guide
                                          TDM           AH      CA MICS Tandem Option Guide
```

4.8.3.1.3 DOC2HTML Examples

```
Examples:

   o  Convert the PIOM (Planning, Installation, Operation
      and Maintenance) and the TSO guides:

      // . . . .JOB . . . .
      //HTMLCVRT EXEC MICSDOCH,
      //         SYSPARM='GUIDE=TSO PIOM'

   o  Convert all end-user guides:

      // . . . .JOB . . . .
      //HTMLCVRT EXEC MICSDOCH,
      //         SYSPARM='CAT=US'
      //

   o  Convert all end-user guides and the System
      Modification Guide (SMG):

      // . . . .JOB . . . .
      //HTMLCVRT EXEC MICSDOCH,
      //         SYSPARM='CAT=US GUIDE=SMG'
      //

   o  Convert all Accounting and Chargeback Guides:

      // . . . .JOB . . . .
      //HTMLCVRT EXEC MICSDOCH,
      //         SYSPARM='COMP=ACT'
      //

   o  Convert all guides with CA MICS Data Dictionary
      entries:

      // . . . .JOB . . . .
      //HTMLCVRT EXEC MICSDOCH,
      //         SYSPARM='GUIDE=ALL DIC=Y'
      //
```

4.8.3.2 Generated Documentation Update Utility (DOCUPDTG)

Whenever an xxxCGEN is run, the chapters describing file layouts and contents are updated. DOCUPDTG provides a way of refreshing these pages in DocWebVu without having to reconvert an entire guide.

Note: DOCUPDTG refreshes the generated pages of all CA MICS Data Integration Applications and Field Developed Applications documentation.

Sample JCL:

```
    // . . . .JOB . . . .
    //HTMLCVRT EXEC MICSDOCH,
    //SYSIN    DD DISP=SHR,DSN=sharedprefix.SOURCE(DOCUPDTG)
    //
```

4.8.4 CA MICS Documentation Web View (DocWebVu)

DocWebVu is the HTML presentation of CA MICS documentation and is created using the DOC2HTML Utility.

To start DocWebVu, display the file using the URL of

      $MICSROOT/documents/contents.html

where $MICSROOT is the name of CA MICS Root Directory.

Note: the path of the URL may differ according to how the Web Server is configured. However, the initial page of DocWebVu is always contents.html.

4.8.5 IBM HTTP Server Considerations

CA MICS Documentation Web View has been designed and tested on the IBM HTTP Server running on OS/390. The following sections describe some considerations when implementing the DocWebVu in this environment.

1 - Using the Pass Directive
2 - Access Permissions

4.8.5.1 Using the Pass Directive

The CA MICS Documentation Conversion utility stores its output (DocWebVu) in the $MICSROOT/documents directory (where $MICSROOT is the name of the CA MICS Root Directory).

In order to make DocWebVu available to a web server, one may be inclined to copy the entire documents/directory into the home directory of the web server.

However, there are inherent problems whenever multiple copies of the same files exist (i.e., keeping the files synchronized).

To eliminate these problems, the Pass Directive of the IBM HTTP Server is recommended.

When specified in the HTTP Server Configuration file, (e.g., httpd.config), Pass Directive can redirect a requested Web server directory to any HFS directory.

For example, with the following Pass Directive in httpd.config:

```
  Pass /micsdoc/* /user/MICS/documents/*
```

if a user requests /micsdoc/contents.html, the Web server will give them /user/MICS/documents/contents.html.

Reference: IBM Publication SC31-8690 HTTP Server Planning,
            Installing, and Using

4.8.5.2 Access Permissions

The access permissions of all files and directories must be set properly for the IBM HTTP Server to access CA MICS Documentation Web View. It is recommended that a file access mask of rw-rw-r-- be used (Owner=read/write, GROUP=read/write, Other=read) for all DocWebVu files. The chmod command can be used to set/reset permissions.

It is recommended that a file access mask of rwxr-xr-x be used (Owner=read/write/execute, Group=read/execute, Other read/execute) for all DocWebVu directories. The chmod command can be used to set/reset permissions.

Refer to section 5.7.2.2 for more information on access permissions.

4.8.6 Exporting DocWebVu

CA MICS Documentation Web Viewer has been designed to work with the OS/390 IBM HTTP Server and WebSphere.

However, if your installation prefers to maintain the CA MICS documentation on another web server platform, facilities are provided to package the generated DocWebVu files and directories for transport to an alternative platform.

o The exportdoc command facilitates transporting DocWebVu to an alternative web server platform by creating a UNIX TAR file with optional EBCDIC to ASCII conversion and data compression.

o Once the TAR file is created, you can transport the file to the target platform and then extract the DocWebVu files and directories.

The following sections discuss:

1 - exportdoc Prerequisites
2 - exportdoc Command
3 - exportdoc Command Syntax
4 - CA MICS Documentation Export Job (DOCXPORT)
5 - DOCXPORT Examples
6 - CA MICS Documentation Export trouble shooting
7 - DocWebVu Checklists

4.8.6.1 Exportdoc Prerequisites

In order to use the exportdoc command, either in an USS session or the DOCXPORT job, the following prerequisites must be met:

o The exportdoc command executes on z/OS V2R4 or above.

o The environment variable $MICSROOT must be set to the name of the CA MICS root directory.

This is provided automatically in the DOCXPORT job.

4.8.6.2 Exportdoc Command

The exportdoc command resides in the $MICSROOT directory, and has been provided to package the directory containing DocWebVu into a GZIPed TAR file (e.g. /tmp/docexport.tar.Z).

exportdoc can be executed in a UNIX System Services shell session, under TSO in an OMVS session, or as as a batch job (DOCXPORT).

The file created by exportdoc can then be transmitted to another platform using a utility such as FTP, and uncompressed using utilities such as tar and gzip (UNIX), PowerArchiver, Winzip, and PKZIP (for Microsoft Windows 95/98/NT/2000) and pax (for z/OS USS).

4.8.6.3 exportdoc Command Syntax

```
exportdoc [-opts] [filename]

 [] indicates optional parameters

where

  opts     processing options
             a - Perform EBCDIC to ASCII conversion
             u - do not compress resultant TAR file
             d - debug
             t - trace

 filename  the name of the file that will contain the
           exported file.

           If not specified, the default file name is
           /tmp/docexport.tar.

           Note: if compression is used (the default), then a
           .Z suffix is automatically added to the file name.
```

4.8.6.4 CA MICS Documentation Export Job (DOCXPORT)

The job in sharedprefix.MICS.CNTL(DOCXPORT) can be used to export CA MICS Documentation Web View using exportdoc in a batch environment.

The syntax used for the exportdoc command is the same as described in section 4.8.6.3.

4.8.6.5 DOCXPORT Examples

```
o Export DocWebVu

  // ...JOB...
  //XPORTDOC EXEC PGM=BPXBATCH,                                            00013000
  // PARM='sh $MICSROOT/exportdoc '                                        00014000

  DocWebVu will be exported to /tmp/docexport.tar.Z.

o Export DocWebVu without compression:

  // ...JOB...
  //XPORTDOC EXEC PGM=BPXBATCH,                                            00013000
  // PARM='sh $MICSROOT/exportdoc -u'                                      00014000

  DocWebVu will be exported to /tmp/docexport.tar.

o Export DocWebVu with ASCII conversion:

  // ...JOB...
  //XPORTDOC EXEC PGM=BPXBATCH,                                            00013000
  // PARM='sh $MICSROOT/exportdoc -a'                                      00014000

  DocWebVu will be exported to /tmp/docexport.tar.Z

o Export DocWebVu with ASCII conversion to file
  /tmp/docwebvu.tar.Z

  // ...JOB...
  //XPORTDOC EXEC PGM=BPXBATCH,                                            00013000
  // PARM='sh $MICSROOT/exportdoc -a /tmp/docwebvu.tar'                    00014000

  DocWebVu will be exported to /tmp/docwebvu.tar.Z
```

4.8.6.6 CA MICS Documentation Export Trouble Shooting

o "I used exportdoc to create a tar file and transmitted it to another platform, but the explode utilities can't read it."

Make sure that the utility used to transmit the file sent it as binary, with no character translation. FTP, for example, uses a text translation as its default transfer mode.

Additionally, the utility you are using may not support the compression used by exportdoc.

o "I used exportdoc to create a tar file, successfully transmitted it to another platform, exploded it, then hooked it into my web server. However when I try to view the pages, they are really confusing. I get a lot of Z's, boxes, and other strange characters, but nothing readable."

The HTML pages created by DOC2HTML are in EBCDIC character set, rather than ASCII that other platforms use. Therefore, if the exportdoc file is being moved to a platform using ASCII, any text file must first be converted from EBCDIC to ASCII before it can be used.

exportdoc provides an option (-a) that will perform this necessary conversion. See section 4.8.6.3.

o "The graphical portions of the Web page show as a box with a red 'x' in it. What is going on?"

The CA MICS Document Web Viewer requires directory $MICSROOT/documents/images. If this does not exist it can be copied using the job in sharedprefix.MICS.CNTL(DOCHFSU).

o "I keep running exportdoc and it keeps failing due to lack of space."

There is no solution here except to increase the space available.

Note: The DIC=N will not include CA MICS Data Dictionary in converted documentation. This can significantly reduce the amount of space required for converted guides.

4.8.7 DocWebVu Checklists

The following checklist sections will assist you in implementing and using DocWebVu:

1 - Initialization
2 - Conversion of CA MICS Guides

4.8.7.1 Initialization

___ 1. Ensure the CA MICS Root Directory has been set up and access to it has been obtained.

See section 5.7.2

___ 2. Edit sharedprefix.MICS.PARMS(JCLGENUC) so that it contains the following:

```
DOCHFSU DOC2HTML DOCXPORT DOCUPDTG
```

Submit the job in sharedprefix.MICS.CNTL(JCLGENUC). 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.

___ 3. Submit the job in sharedprefix.MICS.CNTL(DOCHFSU) and ensure that it completes with a condition code of zero.

4.8.7.2 Conversion of CA MICS Guides

___ 1. Make sure that all items in 4.8.7.1 have been completed.

___ 2. Edit the job in sharedprefix.MICS.CNTL(DOC2HTML) and specify guide selection options as described in section 4.8.3.1.1.

Submit the job in sharedprefix.MICS.CNTL(DOC2HTML) and ensure that it completes with a condition code of zero.

Alternatively, invoke CA MICS Workstation Facility (MWF) Documentation Access Print Dialog (MWF;1;3) and follow the instructions to convert documentation to HTML.

___ 3. (Optional) Enable DocWebVu to Web server.

Once DocWebVu has been created via DOC2HTML, it must be made available to the Web Server. Please refer to section 4.8.5 for tips on Web Server configuration.

4.9 CA MICS Service Desk Interface

The CA MICS Service Desk interface allows CA MICS to generate call tickets in Service Desk for specific CA MICS operational job abends and application specific events.

The following sections describe the CA MICS Service Desk interface:

1 - Prerequisites
2 - CA MICS Abend Reporting
3 - Application Specific Alerts
4 - Messages and Return Codes

4.9.1 Prerequisites

In order to use the CA MICS Service Desk interface, the following CA products must be installed:

o Service Desk r11

o CA Common Services for z/OS r11

o Service Desk Integration r11 (part of Common Services)

See the CA Common Services for z/OS: Service Desk Integration Guide for installation instructions.

Note: The CA MICS Service Desk interface does not currently let you specify the URL, userid, or password during implementation. You must define these values in the CAISDI Soap Server.

4.9.2 CA MICS Abend Reporting

CA MICS Abend Reporting produces Service Desk call tickets when CA MICS operational jobs (that is, Daily and Monthly) end with any of the following user abend codes:

User Abend Code	Termination Condition/Reason
100	Database marked non-updatable.
101	Database marked non-updatable; aging in progress when failure occurred.
110	Internal step restart failed; required files are missing or corrupted.
120	Internal step restart failed; saved work files missing or corrupted.
130	Stand-alone AUDIT, HISTW, and/or HISTM archive tape processing job not yet executed.
300	Input data file was empty.
310	The original system identifier in the input data was not defined in the SYSID parameter when the last run of ALLPGEN or BASPGEN occurred.
320	Zone was not defined in ZONE parameter when the last ALLPGEN or BASPGEN job was run.
400	Invalid checkpoint record encountered.
410	Limit of ORGSYSID/COMPONENT entries is exceeded in the Checkpoint File.
420	Invalid FORCE or SELECT range was specified.
430	INPUTRDR end-of-file routine was not entered after the last input record.
440	SAS version is not compatible with CA MICS version.
450	PREFIX in the Checkpoint File does not match the PREFIX used in the update procedure.

User Abend Code	Termination Condition/Reason
460	Checkpoint File initialization failed or the checkpoint routine initial update processing failed.
470	Exit did not return control properly.
500	Execution date less than previous date.
510	YEAR is not in the range of years defined to CA MICS.

To activate this feature, the SD_ACTIVE keyword must be set to YES in prefix.MICS.PARMS(EXECDEF). See Section 2.3.5 for more information on setting EXECDEF options.

4.9.3 Application Specific Alerts

Several CA MICS applications provide the ability to generate Service Desk call tickets for specific events. For example, the CA MICS Performance Manager application can create Service Desk call tickets when the ILM soft capping threshold has been exceeded.

Note that the CA MICS Service Desk Activation option must be set in MWF options to generate Application Specific Alerts. See Section 4.4.2.2 for information on activating this option.

Note: You can save the generated MICF query JCL and code to avoid regenerating JCL. Since the current state of the Service Desk Activation option is also saved, any changes to the Service Desk Activation option will not affect the option in the saved MICF query JCL and code.

4.9.4 Messages and Return Codes

This Section lists the messages generated by the CA MICS Service Desk interface.

Four type codes are used to categorize the messages:

l. Information: A note that documents a CA MICS Service Desk option or potentially important feature in the data.

2. Warning: A condition of either the data or the control statements that does not affect the CA MICS operation, but that can lead to unexpected results.

3. Error: An indication that a problem has been encountered with a control statement that will prevent a successful run of the component. Execution is stopped after all control statements are processed.

4. Termination: An indication that a critical problem has been encountered with the data that requires the component to terminate processing immediately.

```
+-----------------+
| B A S 6 3 0 0 0 |
+-----------------+
```

TEXT: Service Desk Interface successfully created request nnnnnn.

TYPE: Information

REASON: Call ticket nnnnn was create in Service Desk.

ACTION: None required.

```
+-----------------+
| B A S 6 3 0 9 9 |
+-----------------+
```

TEXT: Service Desk Interface failed with return code=rc. Extended Info: x_info

```
TYPE:        Error

REASON:      While trying to open a Service Desk issue,
             CA MICS encountered an error. See below for
             information regarding return code and extend
             information area.

             RC  Description
             --  --------------------------------------------
              4 - SD_ACTIVE not YES

              8 - UWUSVDSK Parameter error. x_info takes the
                  form of:

                  returncode\reasoncode

                  where returncode is 8 and reasoncode is:

                   4 - JMSDETPH not loaded
                   8 - SDactive macro var not defined
                  12 - SD_ACTIVE ne YES
                  20 - (not used)
                  24 - (not used)
                  28 - No summary data
                  32 - No description data

            12 - Error in CAISDI/soap. x_info takes the form
                 of:

                 ap1rc\apirs\soaprc\soaprs\func\msg

                 where:

                 apirc is api return code

                 apirs is api reason code

                 soaprc is the CAISDI/soap return code

                 soaprs is the CAISDI/soap reason code

                 func is the CAISDIP/soap function code

                 msg is additional diagnostic information

            16 - Severe error occurred.

ACTION:     Correct condition causing failure and rerun.
```

4.10 Integration with CA SMF Director

CA SMF Director is a software product for the management of SMF data in a z/OS environment. This product allows you to:

- Automatically Unload SYSx.MANx Files
- Maintain an Inventory of SMF Records by Date, Time, Volume Serial, and system ID.
- Protect Data Sets Containing SMF Records
- Print and/or Extract SMF Records by Date and Time
- Automatically duplex and multiplex SMF Records
- Validate SMF records
- Automatic Magnetic Tape Management
- Integrate with Other Storage Management Products

CA SMF Director can also be directed to write all or a portion of the SMF records being dumped to other output files.

Data management information is provided for the additional output files, called split data sets (files), in the form of an index, referred to as the split index. The index itself is managed and used by CA SMF Director, but a copy of it can be produced by CA SMF Director for use by other applications, such as CA MICS. Throughout this documentation we will refer to the copy of the index as the duplicate split index.

This benefits CA MICS processing in the following ways:

- Enables CA MICS to process exactly the data necessary for a given component

- Eases the preparation of data for incremental processing

The remainder of this chapter covers the following topics:

1 - CA SMF Director Prerequisites
2 - CA MICS Prerequisites
3 - Implementation Scenarios
4 - Checklists
5 - CA SMF Director Usage Notes

4.10.1 CA SMF Director Prerequisites

- CA SMF Director must be managing your data center's SMF dump processing.

- The CA SMF Director dump process must be executed with the SPLIT statements as well as the INDEXDSN and INDEXDUP parameters identifying the primary and duplicate split indices respectively.

- All the split data sets created must be cataloged data sets.

- The CA SMF Director INDEXLIM parameter must be defined to set a limit on the number of entries retained in the index. This number should coincide with the CA MICS SMFDJHLLIMIT value defined in prefix.MICS.PARMS member EXECDEF. See section 2.3.5 for more information on the SMFDJHLLIMIT parameter and its affect on CA MICS input processing.

For information on the SMF record types processed by each CA MICS product, see the "Data Sources" section of the related component guide or the sharedprefix.MICS.PARMS(cccSMFD) member for each CA MICS installed component.

Note: To get the most benefit from CA SMF Director and CA MICS integration, SMF records processed by different components should be recorded to separate CA SMF Director duplicate split indices. For example, if SMF and CICS reside in the same unit, each split statement within the CA SMF Director dump job should refer to a unique CA SMF Director duplicate split index for recording the SMF split data sets. This allows CA MICS to only process the data required by each component step. Otherwise, DAYSMF would still be needed to further split the records to avoid each component step from reading unnecessary SMF data sets.

Below is a sample of a CA SMF Director split statement that demonstrates the use of a discrete duplicate split index for SMF records processed by different components.

```
SPLIT ALL SELECT(70:79) TODD(RMF)
   INDEXDSN(hlq.RMF.SMFDINDX)
   TIME(0000) TIMEGDG(hlq.RMFONLY) TIMEUNIT(3390)
   TIMEVOL(SMFD01) TIMESPACE(CYL,500,200)
   INDEXDUP(INDEXRMF).
```

```
SPLIT ALL SELECT(110) TODD(CICS)
   INDEXDSN(hlq.CIC.SMFDINDX)
   TIME(0000) TIMEGDG(hlq.CIC110) TIMEUNIT(VTAPE)
   INDEXDUP(INDEXCIC).
```

Note: The INDEXDUP keyword is included on each SPLIT statement and each refers to a unique DD statement that, in turn, describes a unique duplicate split index data set. This enables each CA MICS component step to input the duplicate split index into the corresponding component step without processing unneeded record types.

If, in the above example, each split data set index entry was recorded to the same duplicate split index, the duplicate split index should be defined to DAYSMF. Otherwise, each component step would read ALL the eligible input files defined in the duplicate split index, which would result in unnecessary processing. The best practice would be to separate the duplicate split index entries by component to maximize the benefit of CA SMF Director.

For more information on implementation, see section 4.10.3.

4.10.2 CA MICS Prerequisites

1. The CA MICS DAILY or incremental update job must have a SMFDRCTR data definition (DD) statement referring to the CA SMF Director duplicate split index allocated with a disposition of SHR. It should NEVER be allocated with any other disposition to avoid contention with CA SMF Director itself.

   This is the data set referenced in the data definition (DD) statement identified in the INDEXDUP parameter of the SPLIT statement contained in the CA SMF Director SMF DUMP job.

2. If your installation's SMF dump split data sets are on tape, CA MICS will dynamically allocate the tape data sets. Therefore, you must ensure your installation does not prohibit dynamic allocation of tape data sets or you will be unable to use this feature.

3. Field Developed Applications do not have an automated interface to CA SMF Director at this time for identifying eligible input.

   However, the split data sets created by CA SMF Director can still be input directly to a FDA and processed just like any other SMF data.

4.10.3 CA SMF Director Implementation Scenarios

This section describes various scenarios for implementing CA SMF Director. During your migration to CA SMF Director, you may employ various combinations of the scenarios described below, but ultimately to derive the most benefit from CA SMF Director integration, your final implementation should be Scenario 2 - CA SMF Director without DAYSMF in the case of the DAILY job, or Scenario 4 if you are using Incremental Update.

This requires that each CA SMF Director duplicate split index contain only the SMF records required by an individual component, and with no overlap between units.

1 -- CA SMF Director with DAYSMF Only
2 -- CA SMF Director without DAYSMF
3 -- CA SMF Director with DAYSMF and Direct Input for Select Components
4 -- CA SMF Director with Incremental Update
5 -- CA SMF Director with Incremental Update and SPLITSMF

4.10.3.1 CA SMF Director with DAYSMF Only

In this configuration, CA SMF Director data and SMF data sets are input directly to the DAYSMF step only. This is useful to those sites that write records used by multiple components to the same duplicate split index. This should be considered as a transitional approach only while migrating SMF dump processes to CA SMF Director control.

Note: Processing CA SMF Director data through DAYSMF minimizes the benefit to be derived from CA SMF Director. Best practice would be to customize CA SMF Director to record the split data set entries to separate duplicate split indices based on the SMF record types and CA MICS component needs. For more information, see section 4.10.1 of this guide.

In this configuration, the CA SMF Director duplicate split index is not directly input to the component steps. This configuration is an extension of default CA MICS processing. In addition, standard SMF data is also input to DAYSMF.

```
 JCLDEF Options
 ---------------
 COMPONENTS   TSO RMF CIC SMF
 SMFRECORDING CIC
 SMFDIRECTOR  TSO RMF CIC SMF
 DAYSMF FILES PERMANENT

    +--------+       prefix.MICS.PARMS(INPUTRDR)
    | DAYSMF |       //INPUTSMF  DD
    +--------+       //SMFDRCTR  DD
        |
        V
    +--------+       (generated from DAYSMF)
    | DAY010 |       //INPUTSMF DD . . prefix.MICS.TSO.DATA
    +--------+
        |
        V
    +--------+       (generated from DAYSMF)
    | DAY020 |       //INPUTSMF DD . . prefix.MICS.RMF.DATA
    +--------+
        |
        V
    +--------+       (generated from DAYSMF)
    | DAY030 |       //INPUTSMF DD . . prefix.MICS.SMF.DATA
    +--------+
        |
        V
```

```
+--------+       (generated from DAYSMF)
| DAY040 |       //INPUTSMF DD . . prefix.MICS.CIC.DATA
+--------+       prefix.MICS.PARMS(INPUTCIC)
```

4.10.3.2 CA SMF Director without DAYSMF

In this configuration, the DAYSMF step is not used to split SMF data. This is used when SMF data is preprocessed or split independent of CA MICS processing. In the case of CA SMF Director, the data for each component is recorded in a unique CA SMF Director duplicate split index.

When the DAYSMF step is deactivated, the INPUT is defined in the prefix.MICS.PARMS(INPUTccc) members.

In this example:

- o The DAYSMF step will not be generated owing to the 'DAYSMF OFF' statement in prefix.MICS.PARMS(JCLDEF).
- o The SMFDIRECTOR statement in prefix.MICS.PARMS(JCLDEF) is optional as it only affects the DAYSMF step. The presence of the SMFDRCTR DD within the individual component steps automatically triggers processing of CA SMF Director input.
- o The prefix.MICS.PARMS(INPUTccc) (where ccc is the component identifier) is used to define the CA SMF Director duplicate split index via the SMFDRCTR DD statement. If a data set containing standard SMF data is used, the INPUTSMF DD is also specified.
- o The DAY010 step has input data from a standard SMF data set.
- o The DAY020 and DAY040 steps have input data from a standard SMF data set and from CA SMF Director.
- o The DAY030 step has standard SMF input.

```
JCLDEF Options
---------------

COMPONENTS   TSO RMF CIC SMF
SMFDIRECTOR  CIC RMF
SMFRECORDING CIC
DAYSMF       OFF

   +--------+       prefix.MICS.PARMS(INPUTTSO)
   | DAY010 |          //INPUTSMF  DD
   +--------+
       |
       V
```

```
+--------+        prefix.MICS.PARMS(INPUTRMF)
| DAY020 |           //INPUTSMF  DD
+--------+           //SMFDRCTR  DD
    |
    V
+--------+        prefix.MICS.PARMS(INPUTSMF)
| DAY030 |           //INPUTSMF  DD
+--------+
    |
    V
+--------+        prefix.MICS.PARMS(INPUTCIC)
| DAY040 |           //INPUTSMF  DD
+--------+           //SMFDRCTR  DD
```

4.10.3.3 CA SMF Director with DAYSMF and Direct Input for Select Components

In this configuration, standard SMF data is input to the DAYSMF step along with the CA SMF Director duplicate split index, using DAYSMF as a CA SMF Director migration aid.
A subset of the components in the unit have direct CA SMF Director duplicate split index input.

In this example:

- DAYSMF will read the standard SMF data as well as the CA SMF Director duplicate split index. Since only RMF is excluded from the DAYSMF step and CIC is included on the SMFRECORDING statement, any TSO, CIC, and SMF data from either input source will be selected. The INPUTSMF DD statements for the individual TSO, RMF, and CIC steps are automatically generated with a definition to the DAYSMF output files.

- CIC and RMF will input the CA SMF Director duplicate split index in the DAY040 and DAY020 steps. CIC (DAY040) will additionally process CICS data output from the DAYSMF step. The prefix.MICS.PARMS members INPUTCIC and INPUTRMF are used to specify the SMFDRCTR DD statement defining the CA SMF Director duplicate split index.

```
JCLDEF Options
---------------

COMPONENTS   TSO RMF CIC SMF
SMFDIRECTOR  CIC RMF
SMFRECORDING CIC
DAYSMF FILES TEMPORARY
DAYSMF EXCLUDE RMF

   +--------+       prefix.MICS.PARMS(INPUTRDR)
   | DAYSMF |          //INPUTSMF  DD
   +--------+          //SMFDRCTR  DD
       |
       V
   +--------+       (generated from DAYSMF)
   | DAY010 |          //INPUTSMF   DD prefix.MICS.TSO.DATA
   +--------+
       |
       V
   +--------+       prefix.MICS.PARMS(INPUTRMF)
   | DAY020 |          //SMFDRCTR   DD
   +--------+
       |
       V
   +--------+        (generated from DAYSMF)
   | DAY030 |           //INPUTSMF   DD prefix.MICS.SMF.DATA
   +--------+
       |
       v
   +--------+        (generated from DAYSMF)
   | DAY040 |           //INPUTSMF DD prefix.MICS.CIC.DATA
   +--------+        prefix.MICS.PARMS(INPUTCIC)
                        //SMFDRCTR DD
```

4.10.3.4 CA SMF Director with Incremental Update

In this configuration, the CA SMF Director duplicate split index is input directly to the INCRccc job. This implementation assumes the data for this component resides in a separate duplicate split index that avoids reading unneeded SMF data.

In this example:

- INCRUPDATE is in use for the CICS component.

- The SMFDIRECTOR statement in prefix.MICS.PARMS(JCLDEF) identifies the input as CA SMF Director.

- The prefix.MICS.PARMS(INPUTccc) (where ccc is the component identifier) is used to define the CA SMF Director duplicate split index using the SMFDRCTR DD statement. If additional non-CA SMF Director data is also to be input, the INPUTSMF DD should also be specified in this member to define the input.

```
JCLDEF Options
---------------

COMPONENTS   CIC
SMFDIRECTOR  CIC


CICOPS Options
---------------

. . . other statements
INCRUPDATE YES
. . . other statements


   +--------+       prefix.MICS.PARMS(INPUTCIC)
   | CICINCR|          //SMFDRCTR  DD
   +--------+
```

4.10.3.5 CA SMF Director with Incremental Update and SPLITSMF

In this configuration, the CA SMF Director duplicate split index is input to the SPLITSMF job. This configuration should only be used when the CA SMF Director duplicate split index contains entries for other SMF record types. The INCRccc job automatically inputs the output from the SPLITSMF job.

Note: Processing CA SMF Director data through SPLITSMF minimizes the benefit to be derived from CA SMF Director. Best practice would be to customize CA SMF Director to record the split data set entries to separate duplicate split indices based on the SMF record types and CA MICS component needs. For more information, see section 4.10.1.

In this example:

o INCRUPDATE is in use for the CICS component.

o The SMFDIRECTOR statement in prefix.MICS.PARMS(JCLDEF) identifies the input as CA SMF Director.

o The prefix.MICS.PARMS(INPUTRDR) is used to define the CA SMF Director duplicate split index using the SMFDRCTR DD statement. If an additional data set containing standard SMF data is to be input, the INPUTSMF DD should also be specified in this member.

```
 JCLDEF Options
 ---------------

 COMPONENTS   CIC
 SMFDIRECTOR  CIC

 CICOPS Options
 ---------------

 . . . other statements
 INCRUPDATE YES
 INCRSPLIT USE
 . . . other statements


    +--------+       prefix.MICS.PARMS(INPUTRDR)
    |SPLITSMF|          //SMFDRCTR  DD
    +--------+
        |
        V
    +--------+       (generated from SPLITSMF)
    | CICINCR|          //prefix.MICS.ccc.IUSPLTDS
```

```
     +--------+
```

4.10.4 Checklists

This section contains the checklists for the initial implementation of CA SMF Director processing as well as handling potential operational issues.

1 - Implementing CA SMF Director Checklist
2 - Specifying Date and Time Range Selection
3 - Recovering a Failed CA SMF Director Split Data Set

4.10.4.1 Implementing CA SMF Director Checklist

The inclusion of the SMFDRCTR DD statement is controlled by the SMFDIRECTOR keyword in prefix.MICS.PARMS(JCLDEF). However, the successful implementation of CA SMF Director is influenced not only by this keyword but also the combination of several other JCLDEF parameters as well as external factors related to the actual CA SMF Director installation itself.

Prior to processing CA SMF Director managed data, you will need to have collected information regarding the way CA SMF Director is deployed at your installation. To assist you in identifying the optimum CA MICS implementation scenario for your installation, this section provides a worksheet with which you can document this information and use it to tailor CA MICS and CA SMF Director accordingly. This worksheet is not used by CA MICS directly, but is meant as a reference for you. The CA SMF Director Definitions worksheet is illustrated in Figure 4-03.

The following checklist contains both the considerations and practical steps for implementing CA SMF Director in a unit.

Preparation:

__ 1. Review the previous subsections of chapter 4.10 to ensure you are familiar with the requirements of activating this feature. At a minimum, you must review both the prerequisite information in sections 4.10.1 and 4.10.2, as well as the usage notes in section 4.10.5.

__ 2. Consult the Systems Programmer responsible for the SMF data at your installation to obtain the name of the CA SMF Director duplicate split indices defined in the SMF DUMP job and which record types they contain.

__ 3. Use the CA SMF Director Definitions Worksheets in Figures 4-01 through 4-03 to obtain and document the information necessary to implement CA SMF Director.

__ 4. See the sharedprefix.MICS.PARMS(cccSMFD) member if needed to correlate the record types to a CA MICS component to complete the worksheet.

Repeat these steps for every unit database as necessary.

JCLDEF definitions:

__ 5. Insert or update the SMFDIRECTOR statement.

Edit prefix.MICS.PARMS(JCLDEF) and find the SMFDIRECTOR statement and add the components identified in step 3 above.

__ 6. Specify your DAYSMF requirements.

Edit prefix.MICS.PARMS(JCLDEF) and find the DAYSMF keyword and based on your needs identified in step 3 above, either leave it as is, or change it to:

DAYSMF OFF

or

DAYSMF EXCLUDE ccc

ccc is the 3 character component identifier to remove those products that no longer require the additional split processing performed by DAYSMF.

Note: This is not restricted to products using CA SMF Director as input, but may include any SMF eligible component.

__ 7. Review your SMFRECORDING statement.

Edit prefix.MICS.PARMS(JCLDEF) and find the SMFRECORDING statement and ensure it still accurately reflects the component input source. You only need to include an optional SMF component on the SMFRECORDING statement if you wish to process the data in DAYSMF.

INPUTccc and INPUTRDR definitions:

__ 8. For each component defined on the SMFDIRECTOR statement, that is NOT being processed in DAYSMF,

Edit prefix.MICS.PARMS(INPUTccc) and insert the SMFDRCTR DD referring to the CA SMF Director duplicate split index, identified in step 5 above.

If you are also processing standard SMF input for the same component, insert the completed INPUTSMF DD in the same member.

The prefix.MICS.PARMS(INPUTccc) member must exist. If the INPUTccc member does not exist, it must be created or the CA MICS DAILY job will not generate.

Note: If only one component uses SMF input, JCLGEN does not create a DAYSMF step to split the SMF input file. Rather, the SMF File that is defined in prefix.MICS.PARMS(INPUTRDR) goes directly into the DAILY job step for that one product.

If you specified DAYSMF OFF, then all input data, regardless of its origin will be input from each component's corresponding prefix.MICS.PARMS(INPUTccc) member.

__ 9. If you have enabled Incremental Update for a component in this unit and wish to process the CA SMF Director split data sets, you must define the duplicate split index in the corresponding prefix.MICS.PARMS(INPUTccc) member.

If however, you are using the SPLITSMF job for this component, you must instead update the prefix.MICS.PARMS(INPUTRDR) member.

Generation:

__ 10. Edit prefix.MICS.PARMS(JCLGENU) and make sure it has the following line in it:

```
JCLGENU
```

Save prefix.MICS.PARMS(JCLGENU).

Submit prefix.MICS.CNTL(JCLGENU). This job should complete with a return code of zero. Review the MICSLOG and locate message BAS10083. Verify whether this message correctly reflects your definitions for products using SMF and CA SMF Director, as well as the components included in the DAYSMF step. Make any changes necessary and submit the job again until you achieve the desired results.

__ 11. Generate the DAILY and INCRccc jobs as applicable.

Edit prefix.MICS.PARMS(JCLGENU) and add the following line:

DAILY INCRccc

Edit prefix.MICS.CNTL(JCLGENU) and submit the job for execution. This job should complete with a return code of zero.

__ 12. Prior to execution of the DAILY job, review the checklist in section 4.10.4.2 for additional operational considerations.

__ 13. (Optional) Delete SMF work data sets

When the DAYSMF step is present, a SMF work data set, prefix.MICS.ccc.DATA (where ccc is the component identifier), is created for each component that uses SMF data. This data set is either temporary or permanent, depending on the value of JCLDEF keyword DAYSMF FILES.

If you deactivated DAYSMF or excluded a component, you should review these data sets and if they are no longer being used, these data sets may be deleted.

```
                              +-------------------------------------------------------------------------+
                              | CA SMF Director Integration     -   CA SMF Director Definitions         |
                              |                                                                         |
                              +-------------------------------------------------------------------------+
Control Definitions           |                                                                         |
                              | COMPONENTS ____ ____ ____ ____ ____  (RMF, SMF, TSO, etc.)              |
                              | SMFRECORDING    ____ ____ ____       (AST, CIC, IDM, HSM, SNT, VCA, WEB) |
                              |                                                                         |
                              | PARMS Library Member is JCLDEF  -   Reference Sections: 2.3.3.2.1       |
                              +-------------------------------------------------------------------------+
                              |                                                                         |
Execution Option              | DAYSMF      FILES  _________ (TEMPORARY or PERMANENT)                   |
                              | DAYSMF      OFF    _________ (deactivates DAYSMF step in this unit)     |
                              | DAYSMF      EXCLUDE ___ ___ ___ ___  (RMF, SMF, TSO, etc.)              |
                              | SMFDIRECTOR  ____ ____ ____ ___  (RMF, SMF, TSO, etc.)                  |
                              |                                                                         |
                              | PARMS Library Member is JCLDEF  -   Reference Sections: 2.3.3.2.1       |
                              +-------------------------------------------------------------------------+
```

Figure 4-01. CA SMF Director Worksheet Part 1 of 2

```
+-----------------------------------------------------------------------------------------------------------------------+
| CA SMF Director Integration Worksheet  -   Duplicate Split Index Layout                                                |
|                                                                                                                       |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|         |           |      | CA SMF  | CA SMF  | CA SMF  | CA SMF  | CA SMF  | CA SMF  | CA SMF  | CA SMF  | CA SMF  |
|         |           | Std  |Duplicate|Duplicate|Duplicate|Duplicate|Duplicate|Duplicate|Duplicate|Duplicate|Duplicate|
| CA MICS | Unit      | SMF  |  Split  |  Split  |  Split  |  Split  |  Split  |  Split  |  Split  |  Split  |  Split  |
| Product | Identifier| Input| Index 1 | Index 2 | Index 3 | Index 4 | Index 5 | Index 6 | Index 7 | Index 8 | Index 9 |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  AST    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  CIC    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  DB2    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  HSM    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  IDM    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  MQS    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  RMF    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  SMF    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  SNT    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  TSO    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  VCA    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  WEB    |           |      |         |         |         |         |         |         |         |         |         |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|         |           |      |         |         |         |         |         |         |         |         |         |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|         |           |      |         |         |         |         |         |         |         |         |         |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|         |           |      |         |         |         |         |         |         |         |         |         |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|         |           |      |         |         |         |         |         |         |         |         |         |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|         |           |      |         |         |         |         |         |         |         |         |         |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|         |           |      |         |         |         |         |         |         |         |         |         |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|                                                                                                                       |
|                                                                                                                       |
| CA SMF Director Integration     -   Reference Sections: 4.10                                                          |
| Note: The figure ends at CA SMF Duplicate Split Index 9, but there is no restriction on the number of indices you    |
|       may define or input to a CA MICS component step.                                                                |
+-----------------------------------------------------------------------------------------------------------------------+
```

Figure 4-02. CA SMF Director Worksheet Part 2 of 2

The ideal CA SMF Director implementation for your CA MICS environment should base the creation of the SMF split data sets and corresponding duplicate split index entries on your component and unit requirements.

In other words, you must consider the components within a unit and across multiple units when deciding how to record the SMF split data sets into the duplicate split index. Proper planning will save processing time and system resources by enabling CA MICS to select ONLY the necessary SMF data for a component and/or unit.

Important items to consider:

o Is this same component defined in multiple units?

If the answer is yes, then each unit should record their split index entries to separate duplicate split indices. In this way, each unit will read ONLY the records needed for their unit.

o Do I have SMF data from multiple systems all processed within the same CA MICS unit?

If the answer is yes, then each system should record their split index entries to the same duplicate split index. This way, CA MICS can read one index and obtain all the records required for that unit's DAILY update.

Alternatively, each system could record their split index entries to a unique duplicate split index and all the indices for that component could be input to the same component step.

In either case, it is extremely important to ensure that the duplicate split index ONLY contains the entries required for this component and no other. In other words, no other record types should be using the same duplicate split index, regardless of the unit.

o Do I have units with multiple components?

If the answer is yes, the split index entries should also be segregated by component. Each component in the unit should have their split index entries recorded to a separate duplicate split index. This enables the duplicate split index to be input directly to the component step and select only the split index entries necessary to satisfy that components needs.

The Worksheet in Figure 4-02 can be both a planning tool for new CA SMF Director implementations as well as a visual aid for migrating your current CA SMF Director implementation to the optimum setup to suit your CA MICS environment.

Using the worksheet provided in Figure 4-02, enter the Unit Identifier for each component you have installed in a unit.

If the SMF data for a component is contained in a CA SMF duplicate split index, type X in the duplicate split index column for that SMF data type.

For example, if RMF data is recorded to duplicate split index 1, type X in that column.

Once you have identified all the components, their corresponding unit identifier, SMF data source (standard SMF or CA SMF Director) and in which duplicate split index their entries reside, you're prepared to evaluate the contents of the worksheet and tailor CA SMF Director (or CA MICS) accordingly.

Visually, the best implementation is to have only one 'X' per column. This would mean that each component can simply input the CA SMF Director duplicate split index that contains only their required entries, and avoid processing any unnecessary data.

If you have multiple 'X's per row, this would mean you have multiple indices to input to the same component. This too is fine as long as none of the indices are shared between components (again, only one 'X' per column). This would be the equivalent of Implementation Scenario 2, in section 4.3.2 of this guide.

For example, using the sample completed worksheet in Figure 4-03, would result in the following CA MICS implementation:

Unit P contains components AST, RMF, SMF, and TSO. To avoid redundant processing of SMF data, each are recording their split index entries to distinct CA SMF Director duplicate split indices. This allows each to be directly input to their component step.

The SMF component is only in one unit, P, but has entries for split data sets in both duplicate split indices 3 and 9. Both should be defined to the DAY030 step in unit P.

The CIC component resides in two units. Both unit's split index entries are recorded to distinct duplicate split indices, (in this example, indices 5 & 6) enabling each unit to read just the SMF records required for that unit. Since no other SMF data is being recorded to the same duplicate split index we will modify prefix.MICS.PARMS(INPUTCIC) for unit C to include the SMFDRCTR DD defining CA SMF Director duplicate split index 5 for direct input to DAY040. We will also modify prefix.MICS.PARMS(INPUTCIC) for unit 4 to include the SMFDRCTR DD defining CA SMF Director duplicate split index 6 for direct input to DAY040.

The split data sets for the DB2 component in Unit D as well as the VCA component in unit S, are also recorded to a separate CA SMF Director duplicate split index so they too will be input directly to the corresponding component step.

```
+---------------------------------------------------------------------------------------------------------------------+
| CA SMF Director Integration Worksheet  -   Duplicate Split Index Sample Layout                                      |
|                                                                                                                     |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|         |           |      | CA SMF  | CA SMF  | CA SMF  | CA SMF  | CA SMF  | CA SMF  | CA SMF  | CA SMF  | CA SMF  |
|         |           | Std  |Duplicate|Duplicate|Duplicate|Duplicate|Duplicate|Duplicate|Duplicate|Duplicate|Duplicate|
| CA MICS | Unit      | SMF  |  Split  |  Split  |  Split  |  Split  |  Split  |  Split  |  Split  |  Split  |  Split  |
| Product | Identifier| Input| Index 1 | Index 2 | Index 3 | Index 4 | Index 5 | Index 6 | Index 7 | Index 8 | Index 9 |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  AST    |     P     |      |    X    |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  CIC    |     C     |      |         |         |         |         |    X    |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  DB2    |     D     |      |         |         |         |         |         |         |    X    |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  HSM    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  IDM    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  MQS    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  RMF    |     P     |      |         |   X     |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  SMF    |     P     |      |         |         |    X    |         |         |         |         |         |    X    |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  SNT    |           |      |         |         |         |         |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  TSO    |     P     |      |         |         |         |    X    |         |         |         |         |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  VCA    |     S     |      |         |         |         |         |         |         |         |    X    |         |
|---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  WEB    |           |      |         |         |         |         |         |         |         |         |         |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|         |           |      |         |         |         |         |         |         |         |         |         |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|  CIC    |     4     |      |         |         |         |         |         |    X    |         |         |         |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|         |           |      |         |         |         |         |         |         |         |         |         |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|         |           |      |         |         |         |         |         |         |         |         |         |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|         |           |      |         |         |         |         |         |         |         |         |         |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|         |           |      |         |         |         |         |         |         |         |         |         |
+---------+-----------+------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
|                                                                                                                     |
|                                                                                                                     |
| CA SMF Director Integration     -   Reference Sections: 4.10                                                        |
| Note: The figure ends at CA SMF Duplicate Split Index 9, but there is no restriction on the number of indices you  |
|       may define or input to a CA MICS component step.                                                             |
+---------------------------------------------------------------------------------------------------------------------+
```

Figure 4-03. CA SMF Director Sample Worksheet

4.10.4.2 Specifying Date and Time Range Selection

In normal processing, CA MICS scans the CA SMF Director duplicate split indices and merges this information with the Input History Log file (ADMIHL). The ADMIHL file contains the high and low timestamps of data processed by each component for each Original System Id (ORGSYSID). In this manner, only data that has not previously been processed is presented to CA MICS.

In some situations where the ADMIHL file does not exist, everything in the duplicate split index will be presented to CA MICS for processing. An example would be during the first execution of a DAILY job for a new unit.

To restrict the data that will be presented to CA MICS, you may use any of the following options. Options 2 and 3 are mutually exclusive as option 2 takes precedence over option 3 during processing.

1) Remove or comment the SMFDRCTR DD statement from the JCL and manually add the desired data sets to the INPUTSMF DD statement.

2) Alternatively, you can use the SMFD_SELECT statement in the parameter overrides (specified in the PARMOVRD DD statement) to restrict data sets selected from the CA SMF Director duplicate split index to a specific date and time range. This method is desirable when you have only one jobstep (component) in which you need to modify the date and time range selection criterion.

   For additional information on Parameter Override, see section 2.3.6 of the PIOM guide.

3) Use the SELECT statement in prefix.MICS.PARMS(EXECDEF) to restrict the date and time range of data to be processed. The SELECT statement method is used in lieu of the previous two options.

   The SELECT statement selection method is most useful when you have multiple components for which you wish to perform date and time range selection as it will be applied to every component (jobstep) in the unit.

   For additional information on specifying the SELECT statement in prefix.MICS.PARMS(EXECDEF), see section A.3 of the PIOM guide.

```
Note:  The SELECT statement in prefix.MICS.PARMS(EXECDEF)
will remain as specified after the update job completes.
Therefore, you must ensure you remove the SELECT before
execution of the next update process.
```

4.10.4.3 Recovering a Failed CA SMF Director Split Data Set

Should the CA MICS update job fail on the existence of the split file state code of 'X', overt action will need to be taken to recover the split data set. You will need to execute a CA SMF Director EXTRACT job to recover the missing data. You must include the INDEXDSN and INDEXDUP parameters to ensure the CA SMF Director primary and duplicate split index are updated when the job completes. This allows CA MICS to be restarted without any intervention.

```
+ - - - - - -  - - IMPORTANT CONSIDERATIONS - - - - - - - - +
|CA SMF Director will not create any new split data sets for|
|this orgsysid (SID) until the failed split data set has    |
|been recovered.                                            |
|                                                           |
|You must have sufficient security authorization to create  |
|a data set with the same high level qualifier as the failed|
|split data set.                                            |
+ - - - - - -  - - IMPORTANT CONSIDERATIONS - - - - - - - - +
```

1) Review the MICSLOG of the CA MICS step that failed. Locate the BAS64026E error message which indicates a 'X' record has been detected and note the data set name of the failed split data set. The data set name of the duplicate split index can be found in the preceding BAS64022I message.

2) Using ISPF, browse the data set identified in the BAS64022I message. Locate the 'X' record using the data set name from the BAS64026E message. The file state code 'X' is located in column 55.

Make note of the ORGSYSID (SID) which begins in column 147.

Also note the abend code which begins in column 152. The abend code is not needed by the extract job but should be used to determine the cause of the split data set failure. This should assist you in taking corrective action to prevent a recurrence. For example, if you encounter a SB37 (insufficient space) you should modify the TIMESPACE parameter for the failed split data set accordingly.

3) Immediately following the 'X' record will be a 'D' record as identified by the 'D' in column 55. The 'D' record is a diagnostic record which only exists when a 'X' record is generated.

Make note of the following additional fields from the 'D' record:

- the name of the CA SMF Director control data set, referred to as the SCDS, which begins in column 10
- the data set name of the primary split index which begins in column 57
- the start date of the history file, which is located in column 118 in mm/dd/yyyy format
- the start time of the history file, which is located in column 129 in hh:mm:ss format

You will need all these values to complete and execute the CA SMF Director Extract job.

4) To obtain the required date and time parameters for the extract job, you will need to run a CA SMF Director LISTH job that will report on the contents of the SCDS.

See the following example:

```
. . . JOB CARD . . .
//STEP1    EXEC PGM=SMFD,REGION=0M,PARM='READ,PERMIT'
//STEPLIB  DD DISP=SHR,DSN=caiprod.smfdir.r11.cailib
//SCDS     DD DISP=SHR,DSN=scds.name.from.d.record
//SYSPRINT DD SYSOUT=*
//SYSXDIAG DD SYSOUT=*
//XDCCDF   DD DUMMY
//SYSIN    DD *
       LISTH SID(sid) FROM(mmddyy).
/*
//
```

- Modify the SCDS DD to refer to the name of the SCDS found in the 'D' record in the duplicate split index.
- Modify the LISTH parameter and substitute the Orgsysid identified in the 'X' record for the lower case sid.
- To reduce the amount of output produced by the LISTH job, enter the date from the start date field found in the 'D' record into the FROM parameter of the LISTH job.

Note:
- You only need the STEPLIB if the CA SMF Director load library is not in linklist. Modify it as needed.

5) Submit the LISTH job.

6) Using the LISTH job output, locate the FROM date and time entry on the LISTH report that matches the START date and time from the duplicate split index 'D' record for the same orgsysid (SID).

You should find an exact match between the START date and time in the LISTH report and the 'D' record. The TO date and time columns from the LISTH report will reflect the highest date and time for the SMF dump process and should be used to complete the TO parameter of the extract statement for the CA SMF Director Extract job.

Note: If there are multiple LISTH entries subsequent to the START date and time from the D record, you will use the START date and time from the 'D' record and the END date and time from the last LISTH entry, to identify the missing date and time range.

7) Below is an example of a CA SMF Director Extract job. Lower case characters represent fields that require modification or review.

```
//. . JOB CARD . .  .                                <==change
//EXTRACT1 EXEC PGM=SMFD,PARM=READ
//STEPLIB  DD  DSN=hlq.smfdir.r11.CAILIB,DISP=SHR   <==change
//SMFOUT   DD  DSN=dsn.from.BAS64026E.msg,          <==change
//             DISP=(,CATLG,DELETE),
//             UNIT=sysda,STORCLAS=storclas,        <==change
//             SPACE=(CYL,(nn,nn),RLSE),            <==change
//             DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760)
//SCDS     DD  DISP=SHR,DSN=scds.from.d.record      <==change
//DUP      DD  DISP=SHR,DSN=dsn.from.BAS64022i.msg  <==change
//SYSPRINT DD  SYSOUT=*
//SYSXDIAG DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
 EXTRACT
  FROM(mmddyy hhmm) TO(mmddyy hhmm)                  <==change
   SID(orgsysid) SELECT(rectype)                     <==change
    INDEXDSN(hlq.smfd.index) INDEXDUP(DUP)           <==change
     TODD(SMFOUT) .
/*
//
```

- Modify the SMFOUT DD to refer to the name displayed in the BAS64026E and/or the 'X' record. You MUST create the recovered file under the exact same data set name which originally failed, including the generation number.

- Modify the SCDS DD to refer to the name of the SCDS found in the 'D' record in the duplicate split index.

- Modify the DUP DD to refer to the name of the duplicate split index identified in the BAS64022I message.

- Modify the EXTRACT parameter with the following:

  - FROM should be the START date and time from the LISTH output for this ORGSYSID (SID) that matched the 'D' record start date and time.
  - TO should be the END date and time obtained from the LISTH report output analysis as described in the preceding note in step 6.
  - SID(ORGSYSID) should be the orgsysid identified in the 'X' record.
  - SELECT(rectype) should specify the record types needed for this CA MICS component step. See the sharedprefix.MICS.PARMS(cccSMFD) member for the list of record types used by this component, ccc equals the three character CA MICS Component ID, if needed.
  - INDEXDSN is the data set name following the name of the SCDS in the 'D' record of the duplicate split index.
  - INDEXDUP should refer to the completed DUP DD statement.

Note:

- You only need the STEPLIB if the CA SMF Director load library is not in linklist. Modify it as needed.
- The CA SMF Director duplicate split index is ALWAYS allocated with a disposition of SHR only. This avoids contention with CA SMF Director.
- the period terminates the extract statement and is required.

8) Submit the Extract job and ensure it completes successfully. Upon completion, the duplicate split index should contain a 'C' index entry for the SMF split data set that had previously failed.

9) Restart the failed CA MICS update job, be it the DAILY or the incremental update job.

4.10.5 CA SMF Director Usage Notes

If so defined, CA MICS components read the CA SMF Director duplicate split indices and selects ALL previously unseen CA SMF split data sets (files) as input into the DAILY update process.

The success of this processing is based on the following:

o CA SMF Director is being executed on a continual basis.

o When interval data is being managed by a TIME-based SPLIT file, you may need to adjust the times that an SMF switch is issued to ensure that the interval data is available in a timely fashion. When SMF Director writes TIME-based SPLIT files, it will not indicate that a file is ready for processing until a record for that split file appears in the next interval that the TIME statement indicates. In order to ensure that the data is immediately available, make sure that the scheduled time for SMF switch operations are set to the times defined in the TIME operand plus the value of the INTVAL in the active SMFPRMxx member, plus one minute. By doing this, there should be data that falls into the next interval.

  For example, suppose a SPLIT statement has the following TIME operand: TIME(0000,0800,1600), indicating that there should be three iterations of the SPLIT file per day:

  - one with data from midnight to 8 AM
  - one with data from 8 AM to 4 PM
  - one with data from 4 PM until the following midnight

  Also suppose the active SMFPRMxx member has INTVAL(15) defined. This would cause all interval data to be generated every 15 minutes.

  Using this example, you would set automated SMF Switch commands (I SMF) to be issued at 00:16, 08:16 and 16:16 every day. This way, once the dump is completed from the switch, the data from the previous portion of the day would be immediately available.

o CA SMF split data sets are cataloged.

o If the split data sets reside on tape, your installation allows dynamic allocation of tape data sets.

o You are recording different SMF data types to unique CA

SMF Director duplicate split indices, or plan to use DAYSMF.

For example, if you are writing SMF and CICS data to unique split data sets, but recording their creation in the same CA SMF duplicate split index, CA MICS will read all the eligible files in the index. That is, those data sets that satisfy the date and time range criterion of not previously processed, and will read all the records regardless of the component step unless it is DAYSMF, which will further split the data accordingly.

Therefore, it is imperative to complete the worksheet in section 4.10.4.1 and review the implementation scenarios in section 4.10.3 to find the implementation that best describes the way CA MICS operates at your installation and tailor CA SMF Director accordingly.

Additionally, should the split data set creation fail for any reason, the CA SMF Director DUMP job will complete with a return code 4; records diagnostic messages to SYSPRINT; and updates the index and duplicate split index with the data set name, date and time range, etc. of the SMF data processed including a split file state code of 'X'.

If CA MICS encounters a never-before-seen split file state code 'X', it will check further to see if the data for that same date and time range has been recovered. If so, processing will continue. If not, CA MICS will write an error message to the MICSLOG indicating that CA SMF Director failed to create the requested split data set, and terminate the job.

For details on recovering a failed SMF split data set, see section 4.10.4.3.

4.11 Integration with CA Examine Auditing

CA Examine Auditing is a comprehensive software package that can perform an automated technical review of the system, hardware and software environment and identify integrity exposures. A real-time view of the computer system is provided without affecting system performance. It is solely an inspection tool and does not permit users to change the system in any way.

One of the features of CA Examine is the ability to locate specific products in program libraries or SMP/E CSIs. This feature makes it useful as a software inventory tool.

CA MICS integrates with CA Examine Auditing by providing a Program Definition Module (PDM) that identifies CA MICS to CA Examine Auditing.

The following checklist can be used to copy the CA MICS PDM to the CA Common Services for z/OS library.

1.) Edit sharedprefix.MICS.PARMS(JCLDEFC) and add/modify the CAILIB keyword to the name of the CA Common Services for z/OS library

2.) Edit sharedprefix.MICS.PARMS(JCLGENUC) so that it contains the following:

```
 PDMCOPY
```

Submit the job in sharedprefix.MICS.CNTL(JCLGENUC). 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.

3.) Execute sharedprefix.MICS.CNTL(PDMCOPY) to copy the CA MICS PDM module into the CA Common Services for z/OS library.

Note: installation security may restrict write access to CA Common services library.

Chapter 5: Maintenance

CA MICS offers different categories of maintenance that you can apply:

- o Product Service Pack (PSP) maintenance

  You receive product changes once a year or more frequently, if necessary. The PSP comprises both technical enhancements and scheduled maintenance for CA MICS products. For more information on applying PSP maintenance, see How to Use the PSP.

- o Product Information Bulletins (PIBs)

  Product Information Bulletins contain official news about a specific CA MICS product. PIBs are normally distributed with maintenance, but can be distributed immediately, if appropriate.

- o Program Temporary Fixes (PTFs)

  Program Temporary Fixes (also referred to as APARs) are developed to enhance product functionality or correct a software error.

  - Published PTFs

    PTFs are corrections to problems or product enhancements that are important enough to make available immediately. Published PTFs have been certified by QA and will be integrated into the next PSP. Published PTFs are available for download at http://ca.com/support, or can be requested by contacting CA Technical Support.

  - Test PTFs

    Occasionally, clients report a problem where an official fix has not yet been developed. In this case, a test PTF is developed to address the reported problem. Test PTFs have not been certified by QA and are only available by contacting CA Technical Support.

    Note: The urgency of a PIB or a PTF is indicated by its HYPER classification as follows:

HYPER NO	The PIB or PTF is of normal importance.

HYPER YES The PIB or PTF is of high importance.

HYPER FLASH The PIB or PTF is of high importance and has critical impact. These bulletins and fixes are automatically mailed to you.

All of the hyper categories indicated above are available at http://ca.com/support, or can be requested by contacting CA Technical Support. If you register for hyper notification (Hyper Solution Delivery) using the CA website at http://ca.com/support, you will automatically receive any PIBs or PTFs classified as HYPER YES or HYPER FLASH by email.

o User Modifications

User Modifications (or USERMODs) are changes to CA MICS products that you develop to suit your unique requirements. Technical Support does not distribute or support USERMODs, but we are ready to help you meet your site's needs. Remember that CA MICS PSP maintenance and/or PTFs can invalidate your modifications. See the System Modification Guide for policies and procedures for developing, applying, and maintaining USERMODs.

Note: For information about logging your modifications, see the System Administrator Guide.

The following sections describe several methods for solving common administrative tasks, as well as other activities associated with maintaining CA MICS:

This section contains the following topics:

5.1 Create an Alternate CA MICS Environment

This section describes a method for creating alternate CA MICS complex or unit environments from an existing complex or unit. The method explains how to create a completely independent copy of a production CA MICS system.

Users often need an alternate method for applying maintenance or product upgrades that minimally impact their production CA MICS system. These alternate requirements arise from factors such as:

- A variety of change control methods mandated by a formal change control program in the organization.
- Operational and maintenance philosophies that may either demand an extended test period or allow only a very narrow window for testing before propagating changes to production.
- The need to compare the before-and-after images of the CA MICS system changes prior to actual production implementation.

This methodology is based on the concept of establishing a separate CA MICS complex and one or more alternate units to which changes can then be applied and tested without impacting the production CA MICS system. The alternate system is created from a snapshot of the production system. Once created, the alternate system has no connection with the production system, i.e., the resulting alternate system cannot be placed directly into production.

Each day during generation of the alternate complex, create a backup copy of the sharedprefix and prefix libraries for the production complex to protect against accidental updates to production.

Important: Read this entire section before you attempt to build an alternate complex. Follow the instructions carefully.

Use the following checklist to create an alternate complex from a production CA MICS complex. The alternate complex will be a totally independent CA MICS system and, as such, will have different sharedprefix, prefix, and tapeprefix names. Your alternate complex will also have a separate MWF environment. It is important to ensure that this isolation is maintained during the generation and use of the alternate complex. Check your records for any user modifications to CA MICS that contain hard-coded assumptions about data set name information.

As you follow these checklists, be sure to review the JCL you are submitting. To avoid production system problems, verify that the correct data set and PROC names are contained in the job streams you are running. Also, check your output listings for positive indications that updates are actually being applied to the alternate rather than the production data sets.

Note: Ensure that any PROCLIB statements maintained in job submission panels under MWF refer to the correct complex name before submitting jobs from the online dialogues.

Checklists for creating alternate CA MICS environments are listed in the following sections. The first checklist should be followed once. The second checklist should be followed once for each unit created in the alternate environment.

1 - Create a CA MICS Complex from an Existing Complex
2 - Create a CA MICS Unit from an Existing Unit

5.1.1 Create a CA MICS Complex from an Existing Complex

```
************************* Important *************************
*                                                           *
*  Read this entire section before you attempt to build an  *
*  alternate complex from an existing complex.  Follow the  *
*  instructions carefully.                                  *
*                                                           *
*  Ensure that each job submitted throughout this           *
*  checklist completes with a condition code of zero.       *
*  If MICSLOG is present, ensure that it does not contain   *
*  any error messages.                                      *
*                                                           *
*  If you intend to maintain this alternate complex, it is  *
*  highly recommended that you complete any CA MICS         *
*  product changes that are in the PSP Online 'applying'    *
*  state before proceeding.                                 *
*                                                           *
*************************************************************

__ 1.  Duplicate Data Sets and HFS Directory

       Duplicate each of the following data sets and the HFS
       directory using different sharedprefix and
       $NeuMICSROOT names:


       HFS DIRECTORIES AND FILES

       $NeuMICSROOT/*

       INSTALLATION AND MAINTENANCE FILES

       sharedprefix.MICS.TAPELOAD.CNTL
       sharedprefix.MICS.TAPELOAD.LOAD

       COMPLEX-LEVEL DATA SETS

       sharedprefix.MICS.ASM
       sharedprefix.MICS.BIN
       sharedprefix.MICS.CAPACITY (if CAP is installed)
       sharedprefix.MICS.CIMANAGE (if CIM is installed)
       sharedprefix.MICS.CLIST
       sharedprefix.MICS.CNTL
       sharedprefix.MICS.DIC.TEXT
       sharedprefix.MICS.DOC.TEXT
       sharedprefix.MICS.DTF.INDEX (if DTF is used)
       sharedprefix.MICS.GENLIB
       sharedprefix.MICS.HOLD.PARMS
       sharedprefix.MICS.HOLD.USER.SOURCE
       sharedprefix.MICS.INC.TEXT
       sharedprefix.MICS.INCLLIB
       sharedprefix.MICS.ISPHLIB
       sharedprefix.MICS.ISPMLIB
       sharedprefix.MICS.ISPPLIB
       sharedprefix.MICS.ISPSLIB
       sharedprefix.MICS.ISPTLIB
       sharedprefix.MICS.LOAD
       sharedprefix.MICS.LOCALMOD.CNTL
       sharedprefix.MICS.MACAUTOS
       sharedprefix.MICS.MACLIB
       sharedprefix.MICS.MCOLIB
       sharedprefix.MICS.MWFPDS.DATA
       sharedprefix.MICS.MWFSAS.DATA
       sharedprefix.MICS.OBJ
       sharedprefix.MICS.PARMS
       sharedprefix.MICS.PROCLIB (if user PROCLIB employed)
       sharedprefix.MICS.PROTOLIB
       sharedprefix.MICS.PSP.ASM
       sharedprefix.MICS.PSP.BIN
       sharedprefix.MICS.PSP.CLIST
       sharedprefix.MICS.PSP.CNTL
       sharedprefix.MICS.PSP.DIC.TEXT
       sharedprefix.MICS.PSP.DOC.TEXT
       sharedprefix.MICS.PSP.GENLIB
       sharedprefix.MICS.PSP.INC.TEXT
       sharedprefix.MICS.PSP.INCLLIB
       sharedprefix.MICS.PSP.ISPHLIB
       sharedprefix.MICS.PSP.ISPMLIB
       sharedprefix.MICS.PSP.ISPPLIB
       sharedprefix.MICS.PSP.ISPSLIB
       sharedprefix.MICS.PSP.ISPTLIB
       sharedprefix.MICS.PSP.LOAD
       sharedprefix.MICS.PSP.MACAUTOS
       sharedprefix.MICS.PSP.MACLIB
       sharedprefix.MICS.PSP.OBJ
       sharedprefix.MICS.PSP.PARMS
       sharedprefix.MICS.PSP.PC.TEXT
       sharedprefix.MICS.PSP.PROTOLIB
       sharedprefix.MICS.PSP.SASFLS
       sharedprefix.MICS.PSP.SOURCE
       sharedprefix.MICS.PSP.USOURCE
       sharedprefix.MICS.SASFLS
       sharedprefix.MICS.SCREENS
       sharedprefix.MICS.SOURCE
       sharedprefix.MICS.TABLES
       sharedprefix.MICS.TEST.INCLLIB
       sharedprefix.MICS.TEST.LOAD
```

sharedprefix.MICS.TEST.SOURCE
sharedprefix.MICS.USER.SOURCE

Note: $NeuMICSROOT is the name of the CA MICS root directory, as specified in the sharedprefix.MICS.PARMS(JCLDEFC) keyword HFSROOT.

__ 2. Duplicate Data Set with Alternate Names

To create an alternate complex with SAS data libraries from the data in the production complex SAS data libraries, use SAS PROC COPY to duplicate the following data sets with the alternate sharedprefix names:

COMPLEX-LEVEL DATA SETS

sharedprefix.MICS.MCOLIB
sharedprefix.MICS.MWFSAS.DATA
sharedprefix.MICS.TABLES
sharedprefix.MICS.SCREENS
sharedprefix.MICS.CIMANAGE
sharedprefix.MICS.CAPACITY

NOTE: For the rest of this checklist, any references to sharedprefix libraries are directing you to the alternate sharedprefix libraries.

__ 3. Delete ISPTLIB and PARMS Members

Use ISPF Option 3.1, Library Utility, to delete these members from sharedprefix.MICS.ISPTLIB:

```
ICFDBTDB  ICFDBTDD  ICFDSI   MAF$DB   MAFDBCx  MAFIHTx
MAFOJTx   MAFRST    MAFRSTx  MWF$PLB  MWF$SITE
```

(where x is the unit database ID for each installed unit database).

__ 4. Create an MWF Logon Procedure (Alternate Complex)

Create a MWF CLIST or TSO logon procedure for the alternate complex using the CLIST or logon procedure from the production complex as an example. Be sure to change data set names to reflect the alternate sharedprefix libraries.

__ 5. Invoke MWF CLIST (Alternate Complex)

Supply initialization parameter information by invoking the CA MICS Workstation Facility (MWF) CLIST for the alternate complex. The first time you enter MWF in the alternate complex, you must select and supply parameters for each of the first four initialization options listed on the CA MICS Administrator Facility (MAF) Parameters menu. The first three options are straightforward and require the standard CA MICS complex related information. The fourth menu option, which enables you to customize the user data set names allocated by MWF, MSAS, and MICF, is discussed further.

If you need additional information about any of the parameters, use the HELP command for each panel.

__ Select Option 4 from the CA MICS Administrator Facility Parameters menu (5;0;4 from the CA MICS Workstation Facility (MWF) main menu) and specify a unique data set suffix for the user table library allocated for each MWF user (the TABLES entry).

For example, you may include the word ALT1 in the alternate sharedprefix user table library name to distinguish it from the production userid user table library names:

production complex	MICS.PROD.USER.TABLES
alternate complex	MICS.ALT1.USER.TABLES

Caution: If you do not change the suffix of the user table library name, you will corrupt the production version of the user table library.

__ Also from Option 4, specify a unique suffix for all of the MICF user data sets. The following example is a partial listing of the data sets available.

CAPACITY	MICS.ALT1.USER.CAPACITY
DTFINDEX	MICS.ALT1.USER.DTF.INDEX
DTFOUT	MICS.ALT1.USER.DTFOUT
LOG	MICS.ALT1.USER.LOG
OUTPUT	MICS.ALT1.USER.OUTPUT
REPORT	MICS.ALT1.USER.REPORT
USER	MICS.ALT1.USER.DATA

Caution: If you do not change the suffix of a MICF user data set, you will corrupt the

```
                   production version of that MICF user data
                   set.

       __ (OPTIONAL) When you exit from the MAF Options
          dialog, MICF will ask you to confirm allocation of
          the alternate and unique
          userid.MICS.ALT1.USER.TABLES data set you defined.
          Press the ENTER key to allocate
          userid.MICS.ALT1.USER.TABLES.

__ 6.  Edit sharedprefix.MICS.PARMS(JCLDEFC)

       Change the keywords to reflect the alternate complex.
       In particular check:

        o The SHAREDPREFIX keyword value to reflect the
          alternate sharedprefix for your alternate complex.

        o If you have a user PROCLIB facility, change the
          PROCLIB parameter to specify the PROCLIB data set
          for the alternate complex.  You may also want to
          change the value of the NOTIFY and JOBNAME
          keywords.

        o If specified, change HFSROOT keyword value to
          reflect the alternate CA MICS Root Directory.

          +---------------------------------------------------+
          |Reference(s): Section 5.7 (Root Directory)         |
          +---------------------------------------------------+

__   7. Edit sharedprefix.MICS.PARMS(USERJCL)

       Change the old sharedprefix references to the
       alternate sharedprefix name.  If you have a user
       PROCLIB facility, code the appropriate PROCLIB
       statement.

__   8. Edit sharedprefix.MICS.PARMS(JCLNAMES)

       Change the old sharedprefix references to the
       alternate sharedprefix name.

       +------------------------------------------------------+
       | Reference(s): Section 2.3.3.3.2.3 (JCLNAMES)         |
       +------------------------------------------------------+

       CA MICS lets you supply a unique name for CA MICS JCL
       procedures.  This capability is beneficial for sites
```

sharing a single PROCLIB data set. If the CA MICS PROCs for the production complex must coexist with those from the alternate complex, you must rename the PROC names for the alternate complex to prevent accidental replacement.

If you do not have a user PROCLIB facility, you must use the alternate sharedprefix.MICS.PARMS(JCLNAMES) to define unique procedure names. The following is an example of statements that appear in sharedprefix.MICS.PARMS(JCLNAMES):

```
MICSASM  MICTASM
MICSCOM  MICTCOM
MICSDB   MICTDB&PROCIDENTIFIER
MICSDM   MICTDM
MICSDOC  MICTDOC
MICSDOCH MICTDOCH
MICSDU   MICTDU&PROCIDENTIFIER
MICSGT   MICTGT
MICSGU   MICTGU
MICSNDB  MICTNDB&PROCIDENTIFIER
MICSRPTS MICTRPTS
MICSSHR  MICTSHR&PROCIDENTIFIER
```

```
*********************** WARNING ****************************
*                                                          *
*  Ensure that any PROCLIB references maintained in job    *
*  submission panels under MWF refer to the correct complex *
*  name before submitting jobs from the online dialogs.    *
*                                                          *
*  USING THE INCORRECT PROCEDURE UNDER THESE CONDITIONS    *
*  COULD DAMAGE YOUR PRODUCTION COMPLEX.                   *
*                                                          *
************************************************************
```

__ 9. Edit sharedprefix.MICS.GENLIB(DBTABLE)

Delete all existing entries and insert a single line beginning in column 1 that reads:

```
*::
```

__ 10. Edit and Execute JCLGEN0

__ Edit sharedprefix.MICS.CNTL(JCLGEN0) and change the old sharedprefix references to the alternate sharedprefix name.

__ Submit the following job for the alternate complex:

'sharedprefix.MICS.CNTL(JCLGEN0)'

__ 11. Execute PROCJOB1

Submit the following job:

'sharedprefix.MICS.CNTL(PROCJOB1)'

__ 12. (CONDITIONAL) Execute JCLGEN1

If Accounting and Chargeback is installed, submit the following job:

'sharedprefix.MICS.CNTL(JCLGEN1)'

__ 13. (CONDITIONAL) Protect Production Complex.

If the production complex has any Production Reporting definitions, remove all of them from the alternate complex in order to protect the production complex. After executing the job below, any and all Production Reporting definitions must be re-entered on the alternate complex, using the unique high-level qualifiers for the data set names.

Add a job card, specify the correct PROCLIB library, and submit the following JCL. If you changed the name of the MICSDM procedure in Step 7 of this checklist, be sure to make the same name change in this JCL.

```
//EXEC MICSDM
//SYSTSIN DD *
 ISPSTART PGM(MACTL) NEWAPPL(MWF) +
  PARM(MADA LF=MWFBATCH,ADDLOGIC=MWFMFT,+
     PANEL=ICFCOCL0,NONDISPL,DISPLAY,+
     IF<?RC,EQ,0>,EXIT,ERROR)
```

Caution: DO NOT run this job against your production complex or you will lose your MICF production reporting definitions and will not be able to recover without restoring sharedprefix.MICS.ISPTLIB.

__ 14. (OPTIONAL) Back Up All Libraries (Alternate Complex)

The code and text for the products on the PSP distribution medium are copied directly to your sharedprefix.MICS and sharedprefix.MICS.PSP libraries. To avoid potential space problems, compress your sharedprefix.MICS and PSP libraries before loading the new product.

__ 15. Review Security Requirements (Alternate Complex)

Review your security requirements for the alternate complex. You may want to consider data set security, as well as, authorization levels in MWF.

```
+-----------------------------------------------------+
|Reference: Chapter 4 (Authorization Levels in MWF)  |
+-----------------------------------------------------+
```

5.1.2 Create a CA MICS Unit from an Existing Unit

```
************************* Important *************************
*                                                           *
*  Read this entire section before you attempt to build an  *
*  alternate unit from an existing unit.  Follow the        *
*  instructions carefully.                                  *
*                                                           *
*  Ensure that each job submitted throughout this           *
*  checklist completes with a condition code of zero.       *
*  If MICSLOG is present, ensure that it does not contain   *
*  any error messages.                                      *
*                                                           *
*************************************************************

__  1. Execute COPYLIBS (Build Alternate Unit Libraries)

       Submit the following job:

        'sharedprefix.MICS.CNTL(COPYLIBS)'

      +-----------------------------------------------------+
      | Reference(s): Section 3.5.2.1                       |
      +-----------------------------------------------------+

__  2. Copy Data Sets (Alternate Prefix)

       Copy the following data sets using the alternate
       prefix names:

       UNIT-LEVEL DATA SETS

       prefix.MICS.PARMS   (Use ISPF)
       prefix.MICS.MUOLIB  (Use SAS PROC COPY)
       prefix.MICS.USER.SOURCE

__  3. (CONDITIONAL) Create Alternate Unit

       (CONDITIONAL) Choose to create an alternate unit from
       an existing unit with data or without data by
       performing one of the following:

       __ (WITH DATA) Use SAS PROC COPY to create an
          alternate unit with SAS data libraries from the
          data in the production unit SAS data libraries.

          These data libraries are:

           prefix.MICS.CHECKPT.DATA
```

prefix.MICS.DETAILn
prefix.MICS.DAYSn
prefix.MICS.MONTHSn
prefix.MICS.WEEKSn
prefix.MICS.YEARSn
prefix.MICS.ccc.IUDETAIL
prefix.MICS.ccc.IUDAYS

Edit the alternate prefix.MICS.CHECKPT.DATA by changing the prefix name defined on the first line in column 164 through column 185.

OR

__ (WITHOUT DATA) Although the prefix libraries listed above may be copied from the production unit, you may choose to generate an alternate unit with empty data libraries. If you do so, you must submit the ALLOCATE job later in this checklist to create the SAS data libraries.

NOTE: For the rest of this checklist, any references to prefix libraries are directing you to the alternate prefix libraries.

__ 4. Edit prefix.MICS.PARMS(JCLDEF)

Change the PREFIX, TAPEPREFIX and UNIT ID parameters to reflect the alternate prefix and tapeprefix for this unit database.

If a user PROCLIB contains the alternate complex PROCs, change the PROCLIB parameter in JCLDEF to specify the correct PROCLIB library.

__ 5. Edit Other prefix.MICS.PARMS Members

Edit the members USERJCL, INPUTccc, and IMSSPND (if IMS is installed) in the alternate prefix.MICS.PARMS and change old prefix references to the alternate prefix name.

If a user PROCLIB contains the alternate complex PROCs, change the PROCLIB parameter in USERJCL to specify the correct PROCLIB library.

__ 6. Edit prefix.MICS.PARMS(JCLNAMES)

Change the old prefix references to the alternate prefix name.

__ 7. Delete prefix.MICS.USER.SOURCE(PREFIX)

__ 8. Edit and execute JCLGENA

Point to the correct complex and unit-level data sets and submit the following job:

```
  'prefix.MICS.CNTL(JCLGENA)'
```

__ 9. Review JCL in PROCJOB2

Check the JCL generated by JCLGENA by examining the following members in prefix.MICS.CNTL:

```
  PROCJOB2   MICSNDB   MICSDU
```

If the JCL and JES parameters do not meet your installation's standards, update the prefix.MICS.PARMS members JCLDEF or USERJCL and resubmit JCLGENA.

```
+---------------------------------------------------+
| Reference(s): Section 3.3.3                       |
+---------------------------------------------------+
```

__ 10. Execute PROCJOB2

Move the first group of CA MICS cataloged JCL procedures to the PROC library of the alternate complex by submitting the following job:

```
  'prefix.MICS.CNTL(PROCJOB2)'
```

```
+---------------------------------------------------+
| Reference(s): Section 3.5.4                       |
+---------------------------------------------------+
```

__ 11. (CONDITIONAL) Execute JCLGENF

If this is a SPECIAL unit database, complete this step.

Generate the special database installation jobs. For background execution, submit the following job:

```
  'prefix.MICS.CNTL(JCLGENF)'
```

__ 12. Review UNITGEN Options

Review sharedprefix.MICS.PARMS(UNITGEN) for the options in effect.

```
+-----------------------------------------------------+
|  Reference(s):  Section 5.6.1                       |
+-----------------------------------------------------+
```

Also, in the alternate complex, delete member sharedprefix.MICS.PARMS(UGLIST)

__ 13. Execute UNITGEN

Submit the following job:

```
  'prefix.MICS.CNTL(UNITGEN)'
```

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

If the UNITGEN NOAUTOSUBMIT, or UNITGEN AUTOSUBMIT and NOOPJCL parameters have been specified, UNITGEN will complete with a normal return code of 4. Otherwise, UNITGEN and UNITGEN1 should complete with a return code of 0.

__ 14. (CONDITIONAL) Execute UNITGEN1

Perform this step only if your installation has specified

UNITGEN NOAUTOSUBMIT

You must manually submit the UNITGEN1 job to perform the required JCL and parameter generations.

After the UNITGEN job from the previous step completes successfully, submit the following job:

```
  'prefix.MICS.CNTL(UNITGEN1)'
```

__ 15. Verify UNITGEN1 Results

Verify that the UNITGEN1 job executed successfully.

The UNITGEN1 job contains multiple steps that perform COPYLIBU, CYCLEGEN, JCLGEN, cccPGEN, and GDGSGEN processing or part of it.

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

All steps should complete with a return code of zero.

If errors are encountered during UNITGEN1 job execution, correct the errors and rerun UNITGEN and UNITGEN1 jobs respectively.

```
+---------------------------------------------------+
|  Reference(s):  Sections 5.6 and 5.6.1            |
+---------------------------------------------------+
```

__ 16. Execute PROCJOB3

Move CA MICS database set-up cataloged JCL procedures to the PROC library of the alternate complex by submitting the following job:

```
  'prefix.MICS.CNTL(PROCJOB3)'
```

```
+-----------------------------------------------------+
| Reference(s): Section 3.5.4                          |
+-----------------------------------------------------+
```

__ 17. (CONDITIONAL) Execute JCLGEND

Perform this step only if your installation has specified

UNITGEN NOOPJCL

If NOOPJCL is specified, UNITGEN1 did not generate the CA MICS operational jobs.

To generate these jobs, submit the following job:

```
  'prefix.MICS.CNTL(JCLGEND)'
```

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

All steps should complete with a return code of zero.

```
+-----------------------------------------------------+
       |  Reference(s):  Section 5.6.1                     |
       +---------------------------------------------------+

__ 18. (CONDITIONAL) Execute cccGDGGN

       If Detail Tape Processing is active in this unit,
       complete the following steps to define the tape GDG:

       __ Edit prefix.MICS.PARMS(JCLGENU) so that it reads:

          cccGDGGN

          If Incremental Update is active, add this:

          INCRccc

       __ Submit prefix.MICS.CNTL(JCLGENU) to regenerate
          the JCL for cccGDGGN and INCRccc (if applicable).

       __ Submit prefix.MICS.CNTL(cccGDGGN) to define
          the tape GDG.

       +---------------------------------------------------+
       | Reference(s): Chapter 7, ccc Analyzer Guide(s)    |
       +---------------------------------------------------+

__ 19. (CONDITIONAL) Execute ALLOCATE (WITHOUT DATA)

        Allocate the Online Database and associated files.

        __ Edit prefix.MICS.CNTL(ALLOCATE) and fill in the
           SPACE, VOLSER, and UNIT keyword values.

        __ Submit the following job:

             'prefix.MICS.CNTL(ALLOCATE)'

        +---------------------------------------------------+
        | Reference(s): Section 3.5.6.2                     |
        +---------------------------------------------------+

__ 20. (CONDITIONAL) Execute CKPTINIT and cccIUALC

       Submit the following jobs:

        prefix.MICS.CNTL(CKPTINIT) (if WITHOUT DATA)
        prefix.MICS.CNTL(cccIUALC) (if WITHOUT DATA)

__ 21. (CONDITIONAL) Execute ACTINITA
```

If Accounting and Chargeback is installed in this unit, submit the following job:

```
  'prefix.MICS.CNTL(ACTINITA)'
```

Note: Review component specific features and implementation for additional requirements in the appropriate analyzer guide, i.e., TAPECSU, TAPEDSU, etc. Special considerations for accounting must also be reviewed if the cloned unit had accounting installed and active.

__ 22. Test/Execute DAILY

Test the DAILY by submitting the following job:

```
  'prefix.MICS.CNTL(DAILY)'
```

```
+-----------------------------------------------------+
| Reference(s): Section 3.5.7 and Chapter 4           |
+-----------------------------------------------------+
```

__ 23. (OPTIONAL) Back Up All Libraries (Alternate Unit)

The code and text for the products on the PSP distribution medium are copied directly to your sharedprefix.MICS and sharedprefix.MICS.PSP libraries. To avoid potential space problems, compress your sharedprefix.MICS and PSP libraries before loading the new product.

5.2 Change ISPF Library Names

This section provides a procedure for updating CA MICS after your ISPF library names have changed. Typically, the library names change in order to comply with data set naming standards or when a new version of ISPF has been installed at your site.

Before changing any library names, review Chapter 6 of the CA MICS System Administrator Guide to ensure that the version of ISPF to which you want to switch is supported by CA MICS. Then use the following checklist.

__ 1. Edit sharedprefix.MICS.PARMS(JCLDEFC)

Update the following parameters with the new ISPF library names:

```
ISPLLIB1  ISPMLIB1  ISPTLIB1
ISPLLIB2  ISPMLIB2  ISPTLIB2
```

```
+-----------------------------------------------------+
| Reference(s): Section 2.3.1.6.2 (ISPF Parameters)  |
+-----------------------------------------------------+
```

__ 2. Edit sharedprefix.MICS.CNTL(JCLGEN0)

Update the following parameters on the EXEC statement:

```
ISPTLB1   ISPMLB1
ISPTLB2   ISPMLB2
```

__ 3. Execute JCLGEN0

Regenerate PROCJOB1 by submitting the following job:

```
'sharedprefix.MICS.CNTL(JCLGEN0)'
```

Ensure that there are no error messages in MICSLOG and that the last message is the normal termination message, BAS10999I.

__ 4. Execute PROCJOB1

The following JCL procedures use ISPF libraries and were changed in sharedprefix.MICS.CNTL members:

```
PROCJOB1  MICSDM  MICSGT  MICSGU  MICSRPTS
```

To install these procedures, submit the following job:

```
  'sharedprefix.MICS.CNTL(PROCJOB1)'
```

Ensure that the job completes with a condition code of zero.

__ 5. Change ISPF Library Names

Change the names of the ISPF libraries used in the execution of MWF. See the references for a discussion of options used to invoke MWF.

```
+-----------------------------------------------------+
| Reference(s): Sections 3.3.6.2 and 3.3.6.3          |
+-----------------------------------------------------+
```

__ 6. (CONDITIONAL) Update Panels and Tutorial Libraries

If your CA MICS installation has used the ISPF ISPPREP service to preprocess CA MICS panels and tutorials, continue with this step; otherwise skip to the next step.

Update your preprocessed panel and tutorial libraries using the new version of ISPF.

```
+-----------------------------------------------------+
| Reference(s): Section 3.3.6.4 (ISPPREP)             |
+-----------------------------------------------------+
```

NOTE: Use of ISPPREP is optional if you are migrating from an older release to a newer release of ISPF. It is mandatory if you are migrating from a newer release to an older release because preprocessed panels are not downward compatible.

__ 7. Edit prefix.MICS.PARMS(JCLDEF)

Update the following parameters with the new ISPF library names:

```
     ISPLLIB1  ISPMLIB1  ISPTLIB1
     ISPLLIB2  ISPMLIB2  ISPTLIB2
```

Note: If you are taking advantage of JCLGEN parameter sharing, then these parameters might not be specified in prefix.MICS.PARMS(JCLDEF).

```
       +-----------------------------------------------------+
       | Reference(s): Section 2.3.1.6.7                     |
       +-----------------------------------------------------+

__  8. Execute JCLGENA

       Point to the correct complex and unit-level data sets
       and submit the following job:

         'prefix.MICS.CNTL(JCLGENA)'

__  9. Execute PROCJOB2

       Move the first group of CA MICS cataloged JCL
       procedures to the production PROC library by
       submitting the following job:

         'prefix.MICS.CNTL(PROCJOB2)'

       +----------------------------------------------------+
       | Reference(s): Section 3.5.4                        |
       +----------------------------------------------------+

__  10. Edit/Execute JCLGENU

       __ Edit prefix.MICS.PARMS(JCLGENU) so that it contains
          a single line that reads:

            RESTORE

       __ Submit the following job:

            'prefix.MICS.CNTL(JCLGENU)'

       Make sure that there are no error messages in the
       MICSLOG; that the last message is the normal
       termination message, BAS10999I; and that the job
       completes with a condition code of zero.

*************************************************************
* Repeat Steps 7 through 10 for each unit database.         *
*************************************************************
```

5.3 Update CA MICS for SAS System Changes

```
************************************************************
* NOTE:  If you are upgrading your CA MICS system to       *
*        a new version of SAS, follow the instructions     *
*        outlined in section 5.4.                          *
*                                                          *
*        This checklist is ONLY used for changing the      *
*        SAS System Library names, typically associated    *
*        with application of SAS System maintenance.       *
*                                                          *
************************************************************
```

This section provides a procedure for updating CA MICS when you change SAS System library names. Typically, library names change to comply with data set naming standards or when a new service pack (TSLEVEL) of SAS has been installed at your site.

__ 1. (CONDITIONAL) Stop the Q&R Mainframe Server

If you are using the CA MICS Q&R Mainframe Server with this complex, you must stop the server.

See the CA MICS Q&R Workstation Administrator Guide for general information about stopping the Q&R Mainframe Server.

__ 2. (OPTIONAL) Create CA MICS SAS Libraries

Create CA MICS SAS libraries. You can reference your site SAS libraries, and skip this step.

```
+-----------------------------------------------------+
| Reference(s): Section 3.3.1 (SAS Libraries)         |
+-----------------------------------------------------+
```

Modify the sample job shown below to populate the CA MICS SAS libraries from your new SAS installation libraries by doing the following:

__ If you are using SAS 9.2 or above, use the ISASCNTL and OSASCNTL DD statements with the CONFIG data set by uncommenting them and commenting the ones with the CNTL data set.

__ Change sasprefix to the data set qualifier of the new SAS libraries.

Note that with SAS 9.1.3 or above, sasprefix may include language and encoding abbreviation (e.g. W0 and ENW0).

__ Change micssasprefix to the data set qualifiers that are valid for your CA MICS installation.

__ Add a JOB statement to the JCL and any installation specific JCL.

```
//COPY1    EXEC PGM=IEBCOPY,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT2   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//ISASAUTO DD DISP=SHR,DSN=sasprefix.AUTOLIB
//ISASCNTL DD DISP=SHR,DSN=sasprefix.CNTL
//*ISASCNTL DD DISP=SHR,DSN=sasprefix.CONFIG
//ISASLIB  DD DISP=SHR,DSN=sasprefix.LIBRARY
//ISASMSG  DD DISP=SHR,DSN=sasprefix.SASMSG
//*
//OSASAUTO DD DISP=OLD,DSN=micssasprefix.AUTOLIB
//OSASCNTL DD DISP=OLD,DSN=micssasprefix.CNTL
//*OSASCNTL DD DISP=OLD,DSN=micssasprefix.CONFIG
//OSASLIB  DD DISP=OLD,DSN=micssasprefix.LIBRARY
//OSASMSG  DD DISP=OLD,DSN=micssasprefix.SASMSG
//SYSIN    DD *
  COPY INDD=((ISASCNTL,R)),OUTDD=OSASCNTL
  COPY INDD=((ISASLIB,R)),OUTDD=OSASLIB
  COPY INDD=((ISASMSG,R)),OUTDD=OSASMSG
  COPY INDD=((ISASAUTO,R)),OUTDD=OSASAUTO
/*
//COPY2    EXEC PGM=IEBGENER,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DISP=SHR,DSN=sasprefix.SASHELP
//SYSUT2   DD DISP=OLD,DSN=micssasprefix.SASHELP
//SYSIN    DD DUMMY
```

__ Submit the job and ensure that it finishes with a return code of 0.

__ 3. Update JCLDEFC

Review the reference for information about defining JCLGEN parameters for SAS. Edit sharedprefix.MICS.PARMS(JCLDEFC) and update the following parameters with information that is applicable to the new SAS release or library name:

```
               SASAUTOS  SASCONFIGx  SASHELP  SASLIB
               SASMSG    SASNAME     SASVER

            ******************************************************
            * Reference(s): Section 2.3.1.6.2 (JCLGEN)           *
            ******************************************************

      __   4. Rename Libraries

            Edit sharedprefix.MICS.CNTL(JCLGEN0), changing the
            SASPFX= value on the EXEC statement for the MICSDM
            in-stream cataloged procedure.

      __   5. Execute JCLGEN0

            Submit the following job:

              'sharedprefix.MICS.CNTL(JCLGEN0)'

            Ensure that there are no error messages in SYSTSPRT or
            in MICSLOG and that the last MICSLOG message is the
            normal termination message, BAS10999.

      __   6. Execute PROCJOB1

            The following JCL procedures changed:

              MICSCOM  MICSDM  MICSDOC  MICSDOCH MICSGT  MICSGU
              MICSRPTS MICSSVR MICSASM

            Move them to your CA MICS PROCLIB by submitting the
            following job:

              'sharedprefix.MICS.CNTL(PROCJOB1)'

            Ensure that there are no error messages and that the
            job completes with a condition code of zero.  If you
            maintain PROCs in a production PROCLIB, copy the
            regenerated procedures listed above to your production
            PROCLIB.

      __   7. (CONDITIONAL) Execute JCLGEN1

            Perform this step only if you have the CA MICS
            Accounting and Chargeback product installed.

            Submit the following job:

              'sharedprefix.MICS.CNTL(JCLGEN1)'
```

```
       Ensure that there are no error messages in SYSTSPRT or
       in MICSLOG and that the last MICSLOG message is the
       normal termination message, BAS10999.

*************************************************************
*  Repeat the following steps for each database unit.       *
*************************************************************

__  8. Suspend Production Operations

       NOTE: If you are upgrading from SAS Version 8 to
       SAS 9.1.3 or above and you enabled incremental
       update for one or more products in any of the unit
       databases, make sure the final DAILY job was run.

       Suspend production operations for the unit database by
       using the Operational Status and Tracking SUSPEND
       command in MWF to deactivate production scheduling.

__  9. Edit JCLDEF

       Edit prefix.MICS.PARMS(JCLDEF) and update the
       following parameters with information that is
       applicable to the new SAS release or library name:

         SASAUTOS  SASCONFIGx  SASHELP  SASLIB
         SASMSG    SASNAME     SASVER

       ******************************************************
       * Note:  If you are taking advantage of JCLGEN       *
       *        parameter sharing, then these parameters    *
       *        might not be specified in                   *
       *        prefix.MICS.PARMS(JCLDEF).                  *
       *                                                    *
       *        If SASVER is the only parameter that        *
       *        required a change, proceed to the final     *
       *        step in this checklist.                     *
       ******************************************************

       ******************************************************
       * Reference(s): Section 2.3.3.2 (JCLGEN)             *
       *               Section 2.3.1.6.7 (Parameter Sharing)*
       ******************************************************

       Suspend production operations for the unit database by
       using the Operational Status and Tracking SUSPEND
       command in MWF to deactivate production scheduling.
```

__ 10. Execute JCLGENA

Regenerate the CNTL members by submitting the following job:

```
  'prefix.MICS.CNTL(JCLGENA)'
```

Ensure that there are no error messages in SYSTSPRT or in MICSLOG and that the last MICSLOG message is the normal termination message, BAS10999.

__ 11. (CONDITIONAL) Execute PROCJOB2

The following JCL procedures changed:

```
  MICSDUx  MICSNDBx
```

Move them to your CA MICS PROCLIB by submitting the following job:

```
  'prefix.MICS.CNTL(PROCJOB2)'
```

Ensure that there are no error messages and that the job completes with a condition code of zero.

If you maintain PROCs in a production PROCLIB, copy the regenerated procedures listed above to your production PROCLIB.

__ 12. Execute JCLGENC

Regenerate the CNTL members by submitting the following job:

```
  'prefix.MICS.CNTL(JCLGENC)'
```

Ensure that there are no error messages in SYSTSPRT or in MICSLOG and that the last MICSLOG message is the normal termination message, BAS10999.

__ 13. (CONDITIONAL) Execute PROCJOB3

The following JCL procedures changed:

```
  MICSDBx  MICSSHRx
```

Move them to your CA MICS PROCLIB by submitting the following job:

```
                'prefix.MICS.CNTL(PROCJOB3)'

              Ensure that there are no error messages and that the
              job completes with a condition code of zero.  If you
              maintain PROCs in a production PROCLIB, copy the
              regenerated procedures listed above to your production
              PROCLIB.

       __ 14. Execute JCLGEND

              Submit the following job:

                'prefix.MICS.CNTL(JCLGEND)'

              Ensure that there are no error messages in MICSLOG;
              that the last message is the normal termination
              message, BAS10999; and that the job completes with a
              condition code of zero.

       __ 15. Resume Daily Operations

              Resume the CA MICS Daily operation of the current
              database unit.

       **************************************************************
       *  Repeat the above steps for each database unit.            *
       **************************************************************

       __ 16. (CONDITIONAL) Start the Q&R Mainframe Server

              If you are using the CA MICS Q&R Mainframe Server
              with this complex, restart the server.

       __ 17. (CONDITIONAL) Activate CLIST

              If the name of the CA MICS SAS load library has
              changed, review the reference on the CA MICS
              Workstation Facility (MWF) startup CLIST.

              *******************************************************
              * Reference(s): Section 3.3.6.2                       *
              *******************************************************

       __ 18. Review Instructions

              Review the instructions on updating MWF with the new
              SAS version's parameter and data set name information.

              *******************************************************
```

```
* Reference(s): Section 4.4.2.3                       *
******************************************************
```

5.4 Convert CA MICS to a New SAS Version

This section describes a procedure for converting an existing CA MICS complex to a new version of SAS. This procedure must be completed even if your SAS System Library names are the same as your previous SAS System Library names. Also, you cannot operate your CA MICS complex with mixed versions of SAS.

Considerations for converting to a new version of SAS and the associated checklist for doing the conversion are contained in the following sections:

1 - Considerations for Converting to a New SAS Version
2 - Converting to a New SAS Version Checklist

5.4.1 Considerations for Converting to a New SAS version

This section provides information about converting to the latest version of SAS, as it applies to your CA MICS environment. CA MICS should currently be running a supported release of SAS. Refer to the System Administrator Guide for information on SAS releases supported by CA MICS. Note that the SAS web site contains SAS articles, presentations, and documentation regarding the latest SAS version.

The conversion checklist can only be used to convert from an older to a newer version of SAS (for example, 8.2 to 9.1.3). This checklist does not support conversion from a newer to an older release of SAS (for example, 9.1.3 to 8.2). If a conversion to an older version of SAS is necessary, you must restore all CA MICS data sets to the state they were in before the initial conversion. After restoration is complete, all data from the newer release will be lost.

```
CA MICS SAS Configuration Options Members
------------------------------------------
```

During the SAS conversion, the following members in sharedprefix.MICS.PARMS will be replaced with version-specific members and any customization will be lost:

```
MABATCH  MATSO  MQRBATCH  ICFBATCH
```

To avoid losing local modifications made to these members, the same customization should be made in these corresponding SAS-version-specific members in sharedprefix.MICS.PARMS:

```
MABATCHx  MATSOx  MQRBTCHx  ICFBTCHx
```

where x is 7 for SAS version 8, or 9 for SAS version 9.

```
Library Conversion
------------------
```

Note that once a SAS data library has been converted to SAS version 9, the data library is no longer accessible by SAS versions prior to 9.

```
SAS/TOOLKIT Software
--------------------
```

SAS/TOOLKIT Software is a separate licensed product that permits the writing of SAS procedures, informats, formats, functions, CALL routines, and I/O engines. If your site has written modules using SAS/TOOLKIT, these will have to be recompiled under SAS version 9. While CA MICS uses SAS/TOOLKIT Software to create a number of custom informats, formats, and CALL routines, SAS/TOOLKIT Software is not required to execute them. For more information on SAS/TOOLKIT Software, contact the SAS Institute.

SAS Execution

Unlike previous releases of SAS, version 9 has only one entry point named SAS. The following members must be examined for possible SASNAME SASCFG updates: sharedprefix.MICS.PARMS (JCLDEF and JCLDEFC) and prefix.MICS.PARMS(JCLDEF). The MWF SAS Execution Options need to specify this entry point.

SAS National Language Support

One of the significant features of SAS version 9 is its SAS National Language Support (NLS). NLS provides support for language- specific messages, documentation, sorting, and characters. As a result, the default SAS configuration and the autocall data set names have embedded the language and encoding abbreviations. The IBM EBCDIC encoding (W0) is required for CA MICS. CA MICS has been tested successfully with the language code EN (English). Any other encoding/language combination can cause unexpected results.

5.4.2 Converting to a New SAS Version Checklist

```
*************************************************************
*                                                           *
* WARNING!  This procedure converts your entire CA MICS     *
*           complex and all database units to a new version *
*           of SAS.  You CANNOT operate your CA MICS        *
*           complex with mixed versions of SAS.             *
*                                                           *
*           If you are using CA MICS Query and Reporting,   *
*           (Q&R) you must execute a separate instance of   *
*           the CA MICS Mainframe Server for each unique    *
*           SAS Version.                                    *
*                                                           *
*           This procedure is designed to convert your      *
*           CA MICS environment to SAS Version 9.1.3 or     *
*           higher.                                         *
*                                                           *
*           If you have any questions or concerns, contact  *
*           Technical Support before beginning this         *
*           procedure.                                      *
*                                                           *
*************************************************************
```

This section provides a procedure for updating CA MICS when you upgrade to a new SAS version. This procedure must be completed even if your SAS System Library names are the same as your previous SAS System Library names.

```
*************************************************************
*                                                           *
* NOTE:  This procedure only applies to a new SAS version,  *
*        for example, for upgrading from SAS Version 8 to   *
*        SAS 9.1.3, or from SAS 9.1.3 to SAS 9.2.           *
*                                                           *
*        Do NOT use this procedure to upgrade to a new      *
*        maintenance release of your SAS system, for        *
*        example, upgrading from SAS 9.1.3 TS-1 to TS-2.    *
*        For such maintenance upgrades, see section 5.3.    *
*                                                           *
*************************************************************
```

__ 1. (CONDITIONAL) Stop the Q&R Mainframe Server

If you are using the CA MICS Q&R Mainframe Server with this complex, you must stop the server.

See the CA MICS Q&R Workstation Administration Guide for general information about stopping the Q&R Mainframe Server.

__ 2. Apply CA MICS Product Changes

Apply the CA MICS product changes that provide support for the new SAS version.

For assistance, contact Technical Support.

__ 3. Back up Existing CA MICS SAS Libraries

Back up your existing CA MICS SAS libraries. If you plan to use the same libraries for the new SAS version that you use for the current SAS version, we recommend that you make a backup copy of your current SAS libraries before proceeding.

__ 4. (OPTIONAL) Create CA MICS SAS Libraries

Create CA MICS SAS libraries. You can reference your site SAS libraries, and skip this step.

```
+------------------------------------------------------+
| Reference(s): Section 3.3.1 (SAS Libraries)          |
+------------------------------------------------------+
```

Modify the sample job shown below to populate the CA MICS SAS libraries from your new SAS installation libraries by doing the following:

__ If you are using SAS 9.2 or above, use the ISASCNTL and OSASCNTL DD statements with the CONFIG data set by uncommenting them and commenting the ones with the CNTL data set.

__ Change sasprefix to the data set qualifier of the new SAS libraries.

Note that with SAS 9.1.3 or above, sasprefix may include language and encoding abbreviation (e.g. W0 and ENW0).

__ Change micssasprefix to the data set qualifiers that are valid for your CA MICS installation.

__ Add a JOB statement to the JCL and any installation specific JCL.

```
//COPY1    EXEC PGM=IEBCOPY,REGION=2M
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT2   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//ISASAUTO DD DISP=SHR,DSN=sasprefix.AUTOLIB
//ISASCNTL DD DISP=SHR,DSN=sasprefix.CNTL
//*ISASCNTL DD DISP=SHR,DSN=sasprefix.CONFIG
//ISASLIB  DD DISP=SHR,DSN=sasprefix.LIBRARY
//ISASMSG  DD DISP=SHR,DSN=sasprefix.SASMSG
//*
//OSASAUTO DD DISP=OLD,DSN=micssasprefix.AUTOLIB
//OSASCNTL DD DISP=OLD,DSN=micssasprefix.CNTL
//*OSASCNTL DD DISP=OLD,DSN=micssasprefix.CONFIG
//OSASLIB  DD DISP=OLD,DSN=micssasprefix.LIBRARY
//OSASMSG  DD DISP=OLD,DSN=micssasprefix.SASMSG
//SYSIN    DD *
  COPY INDD=((ISASCNTL,R)),OUTDD=OSASCNTL
  COPY INDD=((ISASLIB,R)),OUTDD=OSASLIB
  COPY INDD=((ISASMSG,R)),OUTDD=OSASMSG
  COPY INDD=((ISASAUTO,R)),OUTDD=OSASAUTO
/*
//COPY2    EXEC PGM=IEBGENER,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DISP=SHR,DSN=sasprefix.SASHELP
//SYSUT2   DD DISP=OLD,DSN=micssasprefix.SASHELP
//SYSIN    DD DUMMY
```

__ Submit the job and ensure that it finishes with a return code of 0.

__ 5. (CONDITIONAL) Conversion to SAS 9.1.3 ONLY

If you are converting to SAS 9.1.3, the SAS Autocall library (sasprefix.W0.AUTOLIB) must be converted to a fixed block data set to address issues documented in SAS Usage Note SN-013990. Review this SAS Usage Note before proceeding with the rest of this conversion checklist.

This data set should be used only for CA MICS, and is only necessary for SAS 9.1.3. The fixed-block version of the SAS Autocall library must be created with an LRECL of 80 and a BLKSIZE that is a multiple of 80.

Note that standard IBM utilities, such as IEBCOPY, will not copy variable blocked data sets to fixed block.

The following is a sample job using the SAS provided VBGT72 macro, referenced in SN-013990 to copy the SAS Autocall library. VBGT72 must be downloaded from the SAS institute support site before it can be used. In this example, VBGT72 was downloaded and then saved as member VBGT72 in sasprefix.VBGT72.LIBRARY:

```
//    ....JOB....
//VBGT72   EXEC SAS
//VBGT72   DD DISP=SHR,DSN=sasprefix.VBGT72.LIBRARY
//ISASAUTO DD DISP=SHR,DSN=sasprefix.W0.AUTOLIB
//AUTOOUT  DD DISP=(NEW,CATLG,DELETE),
//    DSN=micssasprefix.FB.W0.AUTOLIB,
//    UNIT=unit,VOL=SER=volume,
//    DCB=(LRECL=80,RECFM=FB,BLKSIZE=6160),
//    SPACE=(CYL,(1,1,45))
//SYSIN    DD *
 %INCLUDE VBGT72(VBGT72);
 %VBGT72(indd=ISASAUTO);
//
```

The JCL will need to be modified before submission to suit your site requirements.

Alternatively, the copy utility provided with ISPF (Option 3.3) can be used.

```
**************************************************************
*                                                            *
* NOTE:  If you are upgrading from SAS Version 8 to          *
*        SAS 9.1.3 or above, make sure all CA MICS           *
*        production jobs are completed at this point         *
*        and do not run until this checklist is finished.    *
*        In particular, if you enabled incremental update    *
*        for one or more products in any of the unit         *
*        databases, make sure the final DAILY job was run.   *
*                                                            *
**************************************************************
```

__ 6. Execute MICSLS1

Activate the LOAD and PARMS libraries' version-dependent modules for the SAS version in use and convert the sharedprefix.MICS.MCOLIB library to the new version of SAS.

Upon successful completion of this job, your previous SAS sharedprefix.MICS.MCOLIB library will be renamed to sharedprefix.MICS.MCOLIB.SASOLD.

__ Edit sharedprefix.MICS.CNTL(MICSLS1) and fill in the parameters on the EXEC statement. Do not enter parameters on the PROC statement.

```
******************************************************
*  NOTE: For SAS 9.1.3 only, override the PROC       *
*  variable AUTOLIB to point to the fixed block      *
*  version of the SAS Autocall library created       *
*  in a previous step.                               *
*                                                    *
*  If you are running SAS Version 8 or above, the    *
*  MICSLS1 job dynamically allocates the new         *
*  sharedprefix.MICS.MCOLIB data set using the       *
*  allocation parameters of the existing             *
*  sharedprefix.MICS.MCOLIB data set.  If you        *
*  encounter problems with this allocation, for      *
*  example, due to site-specific allocation          *
*  and/or security requirements, you can uncomment   *
*  the NEWSASX DD of the COPYMCO step and define it  *
*  according to your requirements.                   *
******************************************************
```

__ Submit the following job.

'sharedprefix.MICS.CNTL(MICSLS1)'

```
+------------------------------------------------+
| Reference(s): Section 3.2.5                    |
+------------------------------------------------+
```

All job steps must complete with a condition code of zero.

If the job fails for some reason in the COPYMCO step, after allocating the new sharedprefix.MICS.MCOLIB.SAS9 library or in the RENAME step; you must delete sharedprefix.MICS.MCOLIB.SASOLD and/or sharedprefix.MICS.MCOLIB.SAS9 before resubmitting the above job.

__ 7. Edit JCLDEFC

Edit sharedprefix.MICS.PARMS(JCLDEFC) and update the following parameters with information that is applicable to the new SAS version and library names.

Note that SAS version 9.2 and above require multiple SASCONFIGx parameters to be specified.

```
SASAUTOS  SASCONFIGx  SASHELP   SASLIB
SASMSG    SASNAME     SASVER
```

```
+----------------------------------------------------+
|Reference(s): Section 2.3.1.6.2 (Complex Lib Def's))|
+----------------------------------------------------+
```

__ 8. Edit JCLGEN0

Edit sharedprefix.MICS.CNTL(JCLGEN0), changing the parameters on the EXEC statement for the MICSDM in-stream cataloged procedure to reflect the MICS/SAS library names. See section 3.3.2 for a description of the parameters.

__ 9. Execute JCLGEN0

Submit the following job:

'sharedprefix.MICS.CNTL(JCLGEN0)'

__ 10. Execute PROCJOB1

The following JCL procedures changed:

```
MICSCOM  MICSDM  MICSDOC  MICSDOCH MICSGT  MICSGU
MICSRPTS MICSSVR MICSASM
```

Move them to your CA MICS PROCLIB by submitting the following job:

'sharedprefix.MICS.CNTL(PROCJOB1)'

If you maintain PROCs in a production PROCLIB, copy the regenerated procedures listed above to your production PROCLIB.

__ 11. Execute MDBGEN

Generate the meta database by submitting the following job:

'sharedprefix.MICS.CNTL(MDBGEN)'

__ 12. (CONDITIONAL) Execute JCLGEN1

Perform this step only if the CA MICS Accounting and Chargeback Option product is installed.

Submit the following job:

'sharedprefix.MICS.CNTL(JCLGEN1)'

__ 13. Execute MICSCNVT

Convert your CA MICS SAS libraries to the new SAS version.

NOTE: This step is not necessary when upgrading from SAS version 9.1.3 to 9.2.

You can run this job to convert complex-level SAS libraries to the new SAS version. You may or may not be using these libraries at your installation.

(Optional) sharedprefix.MICS.SCREENS
(Optional) sharedprefix.MICS.MWFSAS.DATA
(Optional) sharedprefix.MICS.CAPACITY
(Optional) sharedprefix.MICS.CIMANAGE
(Optional) sharedprefix.MICS.TABLES

Upon successful completion of this job, your original SAS library will be renamed to any.data.set.name.SASOLD, where any.data.set.name refers to any of the data sets listed above.

__ Edit sharedprefix.MICS.CNTL(MICSCNVT) and fill in the parameters on the EXEC statement. Do not enter parameters on the PROC statement. The DATASET= parameter specifies the fully qualified data set name of the data set to be converted.

If you are running SAS Version 8, 9, or above, the MICSCNVT job dynamically allocates the new any.data.set.name data set using the allocation parameters of the existing any.data.set.name data set. If you encounter problems with this allocation, for example, due to site-specific allocation and/or security requirements, you can uncomment the NEWSASX DD of the COPYMCO step and define it according to your requirements.

```
       __ Submit the following job:

          'sharedprefix.MICS.CNTL(MICSCNVT)'

       +----------------------------------------------------+
       | Reference(s): Section 3.2.5                        |
       +----------------------------------------------------+

       All job steps must complete with a condition code of
       zero. However, there is a potential to get a return
       code of U0999 with the following error message:

       ERROR: Cannot open data library DDIN because it is
              uninitialized.

       The above error is caused by attempting to convert an
       empty data set.

       If the MICSCNVT job fails in the COPYMCO step, after
       allocating the new any.data.set.name.sas9 library, or
       in the RENAME step, you must delete
       any.data.set.name.SASOLD and/or any.data.set.name.SAS9
       before resubmitting the above job.

*************************************************************
*  Repeat Steps 14 through 26 for each database unit.      *
*************************************************************

__ 14. Suspend Production Operations

       NOTE: If you are upgrading from SAS Version 8 to
       9.1.3 or above and you enabled incremental
       update for one or more products in any of the unit
       databases, make sure the final DAILY job was run.

       Suspend production operations for the unit database by
       using the Operational Status and Tracking SUSPEND
       command in MWF to deactivate production scheduling.

__ 15. Execute MICSCNVT

       Convert the prefix.MICS.MUOLIB formats library.

       NOTE: This step is not necessary when upgrading from
       SAS version 9.1.3 to 9.2.

       Upon successful completion of this job, your OLD SAS
       formats library will be renamed to
```

prefix.MICS.MUOLIB.SASOLD.

__ Edit sharedprefix.MICS.CNTL(MICSCNVT) and fill in the parameters on the EXEC statement. Do not enter parameters on the PROC statement. The DATASET= parameter specifies the fully qualified data set name of your prefix.MICS.MUOLIB data set to be converted.

If you are running SAS Version 8, 9 or above, the MICSCNVT job dynamically allocates the new any.data.set.name data set using the allocation parameters of the existing prefix.MICS.MUOLIB data set. If you encounter problems with this allocation, for example, due to site-specific allocation and/or security requirements, you can uncomment the NEWSASX DD of the COPYMCO step and define it according to your requirements.

__ Submit the following job:

'sharedprefix.MICS.CNTL(MICSCNVT)'

```
+---------------------------------------------------+
| Reference(s): Section 3.2.5                       |
+---------------------------------------------------+
```

All job steps must complete with a condition code of zero.

If the job fails for some reason in the COPYMCO step, after allocating the new prefix.MICS.MUOLIB.SAS9 library or in the RENAME step, then you must delete the prefix.MICS.MUOLIB.SASOLD and/or the prefix.MICS.MUOLIB.SAS9 library before resubmitting the above job.

__ 16. Edit JCLDEF

Edit prefix.MICS.PARMS(JCLDEF) and update the following parameters with information that is applicable to the new SAS version or library name:

Note that SAS version 9.2 and above require multiple SASCONFIGx parameters to be specified.

```
SASAUTOS  SASCONFIGx  SASHELP  SASLIB
```

SASMSG   SASNAME   SASVER

NOTE: If you are taking advantage of JCLGEN parameter sharing, then these parameters might not be specified in prefix.MICS.PARMS(JCLDEF).

```
+---------------------------------------------------+
|Reference(s): Section 2.3.3.2 (Complex Lib Def's)) |
|              Section 2.3.1.6.7 (Parameter Sharing)|
+---------------------------------------------------+
```

__ 17. Copy JCLGENA

Copy sharedprefix.MICS.CNTL(JCLGENA) to prefix.MICS.CNTL(JCLGENA).

__ 18. Edit JCLGENA

Edit prefix.MICS.CNTL(JCLGENA) to point to the correct complex-level and unit-level data sets.

__ 19. Execute JCLGENA

Regenerate the CNTL members by submitting the following job:

'prefix.MICS.CNTL(JCLGENA)'

__ 20. Execute PROCJOB2

The following JCL procedures changed:

MICSDUx MICSNDBx

Move them to your CA MICS PROCLIB by submitting the following job:

'prefix.MICS.CNTL(PROCJOB2)'

If you maintain PROCs in a production PROCLIB, copy the regenerated procedures listed above to your production PROCLIB.

__ 21. Execute JCLGENC

Regenerate the CNTL members by submitting the following job:

```
          'prefix.MICS.CNTL(JCLGENC)'

   __ 22. Execute PROCJOB3

          The following JCL procedures changed:

          MICSDBx  MICSSHRx

          Move them to your CA MICS PROCLIB by submitting the
          following job:

          'prefix.MICS.CNTL(PROCJOB3)'

          If you maintain PROCs in a production PROCLIB, copy
          the regenerated procedures listed above to your
          production PROCLIB.

   __ 23. Execute JCLGEND

          Submit the following job:

          'prefix.MICS.CNTL(JCLGEND)'

   __ 24. Execute BASPGEN

          Submit the following job:

          'prefix.MICS.CNTL(BASPGEN)'

   __ 25. (Optional) Convert Database Files

          Convert Database Files - DETAIL DAYS WEEKS MONTHS
          YEARS

          NOTE: This step is not necessary when upgrading from
          SAS version 9.1.3 to 9.2.

          NOTE: You may want to wait to convert your CA MICS
          database files to the new version of SAS until all
          CA MICS users have converted their job streams.

          __ Submit the following job to back up all your unit-
             level database files.

             'prefix.MICS.CNTL(BACKUP)'

          __ Delete all your unit-level database files.

          __ (CONDITIONAL) If you enabled incremental update for
             one or more products in this unit database, then
             delete the incremental update database and
             checkpoint files.

             prefix.MICS.ccc.IUCKPT
             prefix.MICS.ccc.IUDETAIL
             prefix.MICS.ccc.IUDAYS

          __ Submit the following job to allocate all your unit-
             level database files.

             'prefix.MICS.CNTL(ALLOCATE)'

          __ (CONDITIONAL) If you enabled incremental update for
             one or more products in this unit database, then
             submit the following job to allocate and initialize
             the incremental update database and checkpoint
             files:

             'prefix.MICS.CNTL(cccIUALC)'

          __ Submit the following job to restore your backed-up
             database files:

             'prefix.MICS.CNTL(RESTORE)'

   __ 26. Resume Daily Operations

   *************************************************************
   *  Repeat Steps 14 through 26 for each database unit.       *
   *************************************************************

   __ 27. (CONDITIONAL) Start the Q&R Mainframe Server

          If you are using the CA MICS Q&R Mainframe Server
          with this complex, restart the server.

   __ 28. (CONDITIONAL) Review References to MWF CLIST

          If the name of the CA MICS SAS load library has
          changed, review the reference that discusses the
          CA MICS Workstation Facility (MWF) startup CLIST.

          +----------------------------------------------------+
          | Reference(s): Section 3.3.6.2                      |
          +----------------------------------------------------+

   __ 29. Review Instructions on Updating MWF
```

Review the instructions on updating MWF with the new SAS version's parameter and data set name information.

NOTE: It is critical that you specify the correct SAS version on the SAS Global Parameters for ISPF Applications (MWF;5;0;3) panel.

```
+------------------------------------------------------+
| Reference(s): Section 4.4.2.3                        |
+------------------------------------------------------+
```

__ 30. Execute MICSCPVT

Inform all your MICF users to convert their private SAS catalogs by following the instructions given below.

NOTE: This step is not necessary when upgrading from SAS version 9.1.3 to 9.2.

Each MICF user may have one or more of the following private SAS catalogs to be converted.

(Optional) userid.MICS6.USER.DATA
(Optional) userid.MICS6.USER.SAS.PROFILE
(Optional) userid.MICS6.USER.CAPACITY

NOTE: You will need to make the sharedprefix.MICS.CNTL(MICSCPVT) job stream available to your users so that they can execute it to convert their private SAS catalogs.

Upon successful completion of this job, the original SAS catalogs will be renamed by adding a .SASOLD suffix to the original data set name.

__ Edit sharedprefix.MICS.CNTL(MICSCPVT) and, optionally, fill in the values for SAS macro variables USERID and PREFIX following the //SYSIN DD statement in the first job step, BLDEXEC.

- o If you leave the USERID parameter blank, the userid associated with the batch job will be used to construct the user data set names. The default data set names are like the following:

  &USERID..MICS.USER.TABLES

- o If you specify the PREFIX parameter, this value will appear in front of the USERID in generated data set names. For example:

```
&PREFIX..&USERID..MICS.USER.TABLES
```

The job dynamically allocates the user's new private data sets and copies the original private data sets to them. If you encounter problems with this allocation, for example, due to site-specific allocation and/or security requirements, you can uncomment the NEWSASX DD of the COPYPVT step and define it according to your requirements.

__ Submit the following job:

```
'sharedprefix.MICS.CNTL(MICSCPVT)'
```

```
+---------------------------------------------------+
| Reference(s): Section 3.2.5                       |
+---------------------------------------------------+
```

All job steps except the COPYPVT step must complete with a condition code of zero. The COPYPVT step may complete with a condition code of zero or 4.

If the job fails for some reason in the COPYPVT step, after allocating the new private data sets, or in the RENAME step; you must delete the new private data sets before resubmitting the above job.

5.5 Implement Complex Level JCLGEN Parameters

This following optional checklist shows you how to activate JCLGEN parameter sharing for all units in the CA MICS Complex.

```
+----------------------------------------------------------+
| Reference(s): Section 2.3.1.6 (JCLDEFC)                  |
|                       2.3.1.6.1 (JCLDEFC Control Def)    |
|                       2.3.1.6.7 (JCLGEN Parm Sharing)    |
|                       2.3.3.2.1 (JCLDEF)                 |
|                       2.3.3.2.1.1 (JCLDEF Control Def)   |
+----------------------------------------------------------+
```

This checklist assumes that sharable unit level parameters are the same as the complex level parameter for all units.

__ 1. Modify JCLDEFC

Modify sharedprefix.MICS.PARMS(JCLDEFC) and add the following as the FIRST statement:

```
COMPLEXPARMS YES
```

__ 2. Edit JCLGENUC

Edit sharedprefix.MICS.PARMS(JCLGENUC) so that it contains the following:

```
PROCJOB1 ALLPROCC
```

Submit the following job:

```
'sharedprefix.MICS.CNTL(JCLGENUC)'
```

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.

__ 3. Execute PROCJOB1

Submit the following job:

```
  'sharedprefix.MICS.CNTL(PROCJOB1)'
```

__ 4. (CONDITIONAL) Copy PROCLIB

All complex-level JCL procedures have changed.

If you maintain PROCs in a production PROCLIB, copy the regenerated procedures to your production PROCLIB.

```
***********************************************************
*  Repeat steps 5 through 9 for each database unit in the *
*  CA MICS complex.                                       *
***********************************************************
```

__ 5. (CONDITIONAL) Edit JCLDEF

Review and edit prefix.MICS.PARMS(JCLDEF). If JCLDEF contains COMPLEXPARMS NO, then remove it. Comment out any JCLDEF parameter that will be cascaded from the complex level to the unit level.

__ 6. Execute JCLGENU

__ Edit prefix.MICS.PARMS(JCLGENU) so that it contains a single line that reads:

```
PROCJOB2 PROCJOB3 RESTORE JCLGENB
```

__ Submit the following job:

```
'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. Execute PROCJOB2

Move the first group of CA MICS cataloged JCL procedures to the production PROC library by submitting the following job:

```
'prefix.MICS.CNTL(PROCJOB2)'
```

```
+----------------------------------------------------+
| Reference(s): Section 3.5.4                        |
+----------------------------------------------------+
```

__ 8. Execute PROCJOB3

Move CA MICS database set-up PROCS to the installation PROCLIB by submitting the following job:

```
         'prefix.MICS.CNTL(PROCJOB3)'

       +---------------------------------------------------+
       | Reference(s): Section 3.5.4                       |
       +---------------------------------------------------+

__  9. (CONDITIONAL) Copy PROCLIB

       All JCL procedures have changed for this unit.

       If you maintain PROCs in a production PROCLIB, copy
       the regenerated procedure to your production PROCLIB.

  ************************************************************
  *  Repeat the above steps for each unit database in the    *
  *  CA MICS Complex.                                        *
  ************************************************************
```

5.6 Consolidated Unit Generation (UNITGEN)

The optional consolidated unit generation (UNITGEN) process is designed primarily to reduce the complexities and number of job executions normally associated with PSP maintenance. UNITGEN may also be used for administrative and other maintenance activities.

Generally, the System Modification section of the Product Change text (PSP.PC.TEXT members) will instruct you to execute a series of one or more unit-level generation jobs in order to implement the Product Change. Often, you will need to repeat these steps in multiple database units, for example, if the product is active in multiple units, or for CA MICS Base Platform (BAS) product changes.

The optional consolidated unit generation process replaces the multiple unit-level generation job executions with a single job: UNITGEN. UNITGEN performs the same functions as JCLGENB, JCLGENC, JCLGEND, ALLPGEN, COPYLIBU, CYCLEGEN, and GDGSGEN.

Usually, the UNITGEN process performs all the functions mentioned above. If you choose to generate and print a composite UNITGEN checklist during the online PSP maintenance process, a tailored UNITGEN is generated that only performs the functions necessary for the set of selected product changes and the specific unit. You must complete such a checklist, in particular the UNITGEN jobs, before you can generate and print another composite UNITGEN checklist.

If a UNITGEN process is performed without a composite UNITGEN checklist, or after a composite UNITGEN checklist is completed, it contains all the functions mentioned above.

- The UNITGEN process actually consists of two jobs: UNITGEN and UNITGEN1.

- The UNITGEN job generates the UNITGEN1 job based on the products that are installed in the unit and the generation processes that are required. Executing the UNITGEN1 job performs the actual processing: JCL generation, cccPGEN, and so on.

- If you specify the UNITGEN AUTOSUBMIT statement in sharedprefix.MICS.PARMS(UNITGEN), then UNITGEN will automatically submit the generated UNITGEN1 job for execution. Otherwise, you will need to manually submit the UNITGEN1 job for execution after UNITGEN completes successfully.

- Note that the AUTOSUBMIT YES statement in JCLDEF (or in JCLDEFC if this unit accepts complex-level defaults) has no effect on the UNITGEN process.

- Note also that UNITGEN1 may complete with a CC=8 when manually submitted. The occurrence of the CC=8 is only on the first run in a new unit.

The following sections document the UNITGEN processing options and provide a checklist for UNITGEN usage.

5.6.1 UNITGEN Processing Options

Parameters controlling UNITGEN processing are specified in sharedprefix.MICS.PARMS(UNITGEN).

The UNITGEN statement has the following syntax:

```
UNITGEN parameter...
```

where parameter can be:

- AUTOSUBMIT/ NOAUTOSUBMIT -- Automatically submit the UNITGEN1 job.

  Note: UNITGEN AUTOSUBMIT processing is independent of the JCLDEF(C) AUTOSUBMIT statement.

  Defaults to NOAUTOSUBMIT.

- OPJCL/NOOPJCL -- Optionally generate the CA MICS operational jobs (e.g., DAILY, WEEKLY, etc.). These are the jobs normally generated by the JCLGEND job:

  ```
  AUDIT, BACKUP, BACKUPM,
  DAILY, DAILYRPT, DAYSMFR,
  EXCFULL, HISTM, HISTW,
  INCRccc, MONTHLY, MONTHRPT,
  RESTORE, SCHEDULE, WEEKLY,
  WEEKRPT, YEARACT, YEARLY
  ```

  Note: If OPJCL is in effect and a composite UNITGEN checklist is requested, the UNITGEN job is tailored to generate only the operational jobs required by the set of selected product changes, if any.

  Defaults to OPJCL.

- COMPRESS/ NOCOMPRESS -- Optionally compress the prefix.MICS.PARMS, prefix.MICS.CNTL, and prefix.MICS.USER.SOURCE data sets during execution of the UNITGEN1 job.

  Defaults to COMPRESS.

Multiple UNITGEN statements can be specified, and each UNITGEN statement can accept more than one parameter. If a parameter is specified more than once, the last parameter specification is accepted.

Here are valid examples of the UNITGEN statement:

```
UNITGEN AUTOSUBMIT NOOPJCL
UNITGEN COMPRESS
```

5.6.2 UNITGEN Checklist

The instructions in the following checklist guide your usage of the UNITGEN process for regenerating unit-level parameters, JCL, and SAS macros.

__ 1. Review/Modify UNITGEN Options

Review sharedprefix.MICS.PARMS(UNITGEN) for the options in effect.

```
+---------------------------------------------------+
|  Reference(s):  Section 5.6.1                     |
+---------------------------------------------------+
```

__ 2. Execute UNITGEN

Submit the following job:

```
  'prefix.MICS.CNTL(UNITGEN)'
```

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

If the UNITGEN NOAUTOSUBMIT, or UNITGEN AUTOSUBMIT and NOOPJCL parameters have been specified, UNITGEN will complete with a normal return code of 4. Otherwise, UNITGEN and UNITGEN1 should complete with a return code of 0.

__ 3. (CONDITIONAL) Execute UNITGEN1

Perform this step only if your installation has specified

UNITGEN NOAUTOSUBMIT

You must manually submit the UNITGEN1 job to perform the required JCL and parameter generations.

After the UNITGEN job from the previous step completes successfully, submit the following job:

```
  'prefix.MICS.CNTL(UNITGEN1)'
```

__ 4. Verify UNITGEN1 Results

Verify that the UNITGEN1 job executed successfully.

The UNITGEN1 job contains multiple steps that perform COPYLIBU, CYCLEGEN, JCLGEN, cccPGEN, and GDGSGEN processing or part of it.

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

All steps should complete with a return code of zero.

If errors are encountered during UNITGEN1 job execution, correct the errors and rerun UNITGEN and UNITGEN1 jobs respectively.

```
+----------------------------------------------------+
|  Reference(s):  Sections 5.6 and 5.6.1             |
+----------------------------------------------------+
```

__ 5. (OPTIONAL) Execute JCLGEND

Perform this step only if your installation has specified

UNITGEN NOOPJCL

If NOOPJCL is specified, UNITGEN1 did not generate the CA MICS operational jobs.

To generate these jobs, submit the following job:

```
  'prefix.MICS.CNTL(JCLGEND)'
```

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

All steps should complete with a return code of zero.

```
+-----------------------------------------------------+
|  Reference(s):  Section 5.6.1                       |
+-----------------------------------------------------+
```

__ 6. (OPTIONAL) Update the CA MICS Jobs

If JCLGEND was executed, then all the CA MICS operational jobs have changed for this unit. If a tailored UNITGEN requested by a composite UNITGEN checklist was executed, then some of the operational jobs may have changed. This includes the following jobs in prefix.MICS.CNTL:

```
AUDIT     BACKUP    BACKUPM   DAILY     DAILYRPT
DAYSMFR   EXCFULL   HISTM     HISTW     INCRccc
MONTHLY   MONTHRPT  RESTORE   SCHEDULE  WEEKLY
WEEKRPT   YEARACT   YEARLY
```

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.

__ 7. (OPTIONAL) Update the CA MICS PROCs

Depending on the JCLDEF parameters, the following CA MICS PROCs may have changed:

```
MICSSHRx  MICSDBx  MICSNDBx  MICSDUx
```

where x is the CA MICS database unit id.

If you maintain PROCs in a production PROCLIB, copy the regenerated procedures listed above to your production PROCLIB.

5.7 Activating CA MICS USS and HFS Support

With the introduction of OS/390 UNIX System Services (USS), a new file system was introduced. The Hierarchical File System (HFS) is based on the directory structure used by most UNIX systems.

Certain CA MICS features and products utilize USS services and create HFS files. For example, CA MICS Documentation Conversion feature and the CA MICS Web Analyzer product require access to USS services and HFS files. It is very likely that your installation is already running applications using USS services and HFS files, and this section explains how to enable CA MICS to these system services.

The following sections describe the processes required to define the CA MICS Root Directory and enable CA MICS to the USS and HFS services.

5.7.1 The CA MICS Root Directory

The CA MICS Root Directory is the name of the HFS sub-directory to be used as the default location for storing HFS files for the CA MICS Platform, CA MICS Data Integration Applications, and CA MICS Management Applications.

In general, CA MICS HFS files are created in sub-directories of the CA MICS Root Directory. You may think of the CA MICS Root Directory as the HFS corollary to the CA MICS sharedprefix -- it becomes the "prefix" for common CA MICS HFS files.

You should also note that CA MICS applications may give you the option to override the CA MICS Root Directory in order to secure HFS files for a specific application area as required by your installation. In essence, the HFS files are equivalent to the MICF Production Reporting job stream definition where you have the option to store job stream definitions and report outputs in files that do not begin with the CA MICS sharedprefix string.

While the CA MICS Root Directory is not required, certain CA MICS features and products do require access to USS facilities, definition of a CA MICS Root Directory, and authorization to create HFS files.

For example, CA MICS Documentation Conversion and the CA MICS Web Analyzer require access to USS services, HFS files, and the CA MICS Root Directory. If you choose not to take advantage of these features and products, then you do not need to define the CA MICS Root Directory. However, it is very likely that your installation is already running applications using USS services and HFS files, and you will presumably want to enable CA MICS to these system services.

5.7.2 Prerequisites

In order to create and access the CA MICS Root Directory there are prerequisites that must be met.

The prerequisites may require intervention from several installation departments such as DASD Management and Security.

The following sections describe the prerequisites in each of the following areas:

1 - Security
2 - Access permissions
3 - HFS Space

5.7.2.1 Security

Prior to using USS services or accessing HFS files, a user must be authorized by the data center's security system (for example, CA Top Secret, CA ACF2, or RACF) by having the OMVS extension added to the user ID.

The CA MICS administrator should have authority to create, read, or delete anything within the CA MICS Root Directory structure.

Additionally, the CA MICS administrator should also have access to a UNIX System Services (USS) session either through a Telnet connection, OMVS using TSO, or/and ISHELL (ISPF USS Shell).

CA MICS "end-users" may require similar authorization, though their access may be limited to "read-only".

5.7.2.2 Access Permissions

Whenever an HFS file or directory is created, security flags called access permissions are set.

These permissions control the level of access a user has for a given directory or file.

There are three levels of access:

- o Owner (e.g., the user creating file)
- o Group (any user in the same "group" as the Owner)
- o Other

Each level has three permissions: read (r), write (w), and execute (x).

Access permissions are usually represented as three three-byte strings indicating the permissions for each level of access.

The following shows a sample directory entry:

```
 access      owner  group    size  create date  filename
-------------------------------------------------------------
-rw-rw-r-- 1 USER06 CA MICSG 10037 Jul 28 23:50 contents.html
```

The owner (USER06) has read/write permission (rw-), anyone in group CA MICSG also has read/write permission (rw-), and the rest have read permission (r--).

CA MICS normally creates HFS files with permissions of rw-rw-r--.

CA MICS also creates HFS subdirectories with permissions of rwxr-xr-x.

Important: However, these permissions may be altered by your installation's USS UMASK definition (default access permissions) or by your security system.

You can use UNIX chmod command to change access permissions for any file or group of files. Here is an example:

```
   chmod -R u+rw,g+rw,o+rw documents/*
```

This example resets access permissions to rw-rw-rw- for all files in the documents directory and any sub-directories of the documents directory.

You can use the oid and gid commands to change the owner and group IDs respectively.

Important: Security packages such as ACF2, TopSecret, and RACF may change access permissions based upon existing security policies.

5.7.2.3 HFS Space

The installation must provide space within the HFS to store CA MICS files. This is called the CA MICS Root Directory.

Each CA MICS complex requires a unique CA MICS Root Directory. Do not attempt to share CA MICS Root Directories between multiple CA MICS complexes (e.g., between a production and test complex), as this action may result in data corruption.

The name of the CA MICS Root Directory is specified on the HFSROOT statement in sharedprefix.MICS.PARMS(JCLDEFC).

Note: The name of the root directory must be no longer than 64 characters. If it is, DOC2HTML will abend.

By default, HFSROOT has a value of /MICS/sharedprefix where sharedprefix is the name specified on the SHAREDPREFIX statement in sharedprefix.MICS.PARMS(JCLDEFC). Note the MICSLEVEL/NOMICSLEVEL option does not apply to the default HFSROOT value.

For more information, see the Section 2.3.1.6.1 for the HFSROOT keyword and Section 2.3.1.6 for the JCLDEFC parameter definition member.

5.7.3 Activation Checklist

This checklist describes the process of enabling the CA MICS Root Directory.

__ 1. Complete the CA MICS Root Directory Prerequisites

Make sure that all necessary permissions and HFS directories are created before completing this checklist.

You can use the UNIX mkdir command to create the CA MICS Root Directory in your HFS.

For example:

```
     mkdir -p /u/users/MICS/complex1
```

```
+----------------------------------------------------+
| Reference(s): Section 5.7.2                        |
+----------------------------------------------------+
```

__ 2. Define the HFS Directory to CA MICS

Edit sharedprefix.MICS.PARMS(JCLDEFC) and add or modify the HFSROOT keyword with the name of the CA MICS Root Directory.

For example:

```
     HFSROOT '/u/users/MICS/complex1'
```

```
+----------------------------------------------------+
| Reference(s): Section 2.3.1.6.2 (HFSROOT)          |
+----------------------------------------------------+
```

__ 3. Edit/Execute JCLGENUC

__ Edit sharedprefix.MICS.PARMS(JCLGENUC) so that it contains the following:

```
     BASHFSU  PROCJOB1  ALLPROCC
```

__ Submit the following job:

```
     'sharedprefix.MICS.CNTL(JCLGENUC)'
```

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.

__ 4. Execute PROCJOB1

Submit the following job:

```
'sharedprefix.MICS.CNTL(PROCJOB1)'
```

__ 5. Copy Regenerated MICSDOCH

The MICSDOCH JCL procedure has changed.

If you maintain PROCs in a production PROCLIB, copy the regenerated procedure MICSDOCH to your production PROCLIB.

__ 6. Execute BASHFSU

Unload CA MICS HFS files to the CA MICS Root Directory by submitting the following job:

```
'sharedprefix.MICS.CNTL(BASHFSU)'
```

Ensure that there are no error messages in MICSLOG or SYSTSPRT, and that the job completes with a condition code of zero.

5.7.4 BASHFSLD Utility

The provided BASHFSLD utility is used to reload all delivered files to the $MICSROOT directory for all products and components.

BASHFSLD is a REXX exec that unloads various sharedprefix.MICS.BIN members into the CA MICS Root Directory.

Since BASHFSLD can overlay any existing files, it should be run with caution because any user modifications made to these files will be lost.

To run this utility, execute the BASHFSLD job in sharedprefix.MICS.CNTL. Sample JCL is listed below, where ccc is the component identifier:

```
//BASHFSLD EXEC IKJEFT01,
//         PARM='%BASHFSLD ccc ( options'
//
```

By default, BASHFSLD will unload HFS files for all sharedprefix.MICS.BIN members defined in cccLHFSV members. However, precise cccLHFSV members can be selected by specifying them on the %BASHFSLD invocation.

BASHFSLD Options

TRACE   Activates REXX tracing options.

DEBUG   Displays additional messages for debugging.

LIST   Lists members to be written, but does not copy to the $MICSROOT directory.

BYPASS   Missing members in cccLHFSV are ignored and a warning message is issued.

NOOVERWRITE Does not overwrite existing members.

5.8 Add New System ID or LPAR to an Existing Database

The following checklist assists you in adding a new SYSID or LPAR for PR/SM to a CA MICS database unit. It assumes that the operating system measurements have not changed.

__ 1. Review References to SYSID or ORGSYSID

Review option statements in the following members for any reference to SYSID or ORGSYSID. Determine whether logic for the new SYSID or LPAR should be added to the account code derivation logic routines, and update as required.

```
'sharedprefix.MICS.PARMS(ACCTRTE)'
'sharedprefix.MICS.PARMS(cccACRT)'
  where ccc is the component identifier.
```

__ 2. (OPTIONAL) Execute the RMFCGEN

The CPUID parameter in 'sharedprefix.MICS.GENLIB(RMFGENIN)' may need to be modified. If changes are made, submit the following job to accept your updates:

```
'sharedprefix.MICS.CNTL(RMFCGEN)'
```

Ensure that there are no error messages in ISPLOG and MICSLOG, and that the job completes with a condition code of zero.

__ 3. (OPTIONAL) Review References to SYSID or ORGSYSID in EXITS

Review the #cccEXIT members in 'sharedprefix.MICS.SOURCE,' where 'ccc' is the component identifier or architecture (BAS). Determine whether the user exits in these members contain any reference to SYSID or ORGSYSID. Update each exit as required.

__ 4. Review Installation Customization for SYSID References

Review user-defined SAS code and formats for SYSID specific references.

__ 5. Verify Checkpoint File Size

Review Checkpoint File size to ensure that there are enough entries for the systems/subsystems that are being added.

```
+-----------------------------------------------------+
| Reference(s):  Section 2.3.2.4                      |
+-----------------------------------------------------+
```

__ 6. (OPTIONAL) Specify Complex Computing System Parameter Definitions

Specify the complex-level computing system parameter definitions. Complex-level computing system definitions are located in sharedprefix.MICS.PARMS(CPLXSID). Carefully read the referenced section listed below before specifying the option values.

```
+-----------------------------------------------------+
| Reference(s):  Section 2.3.1.9                      |
+-----------------------------------------------------+
```

__ 7. (OPTIONAL) Execute CPLXGEN

Run the CPLXGEN job to generate complex option definition global switches, system and time zone definitions, default YEARS time-span, week start, and fiscal calendar parameters. Submit the following job:

```
  'sharedprefix.MICS.CNTL(CPLXGEN)'
```

```
+-----------------------------------------------------+
| Reference(s):  Section 2.3.1.8 (CPLXDEF)            |
|                Section 2.3.1.9 (CPLXSID)            |
|                Section 2.3.1.10(CPLXZONE)           |
|                Section 3.4.2 (cccGENIN)             |
+-----------------------------------------------------+
```

```
*************************************************************
*  Repeat the above steps for each database unit that will  *
*  process data from the new SYSID or LPAR.                 *
*************************************************************
```

__ 8. Suspend Operation

Suspend production operations for the unit database by using the Operational Status and Tracking SUSPEND command in MWF to deactivate production scheduling.

__ 9. Review and Update Parameters

Define the required PARMS members.

__ CA MICS base platform (BAS):

'prefix.MICS.PARMS(SYSID)'
'prefix.MICS.PARMS(ZONE)'
'prefix.MICS.PARMS(SITE)'

Update the members in prefix.MICS.PARMS for the appropriate products.

__ CA MICS Analyzer Option for MeasureWare (AUM):

'prefix.MICS.PARMS(INPUTAUM)'
'prefix.MICS.PARMS(AUMAURT)'
'prefix.MICS.PARMS(AUMOPS)'

__ CA MICS Analyzer Option for VAX/VMS (DEX):

'prefix.MICS.PARMS(INPUTDEX)'
'prefix.MICS.PARMS(DEXOPS)'

__ CA MICS Analyzer Option for IMS (IMS):

'prefix.MICS.PARMS(INPUTIMS)'
'prefix.MICS.PARMS(IMSAURT)'
'prefix.MICS.PARMS(IMSOPS)'
'prefix.MICS.PARMS(IMSRLRT)'
'prefix.MICS.PARMS(IMSSPND)'

__ CA MICS Analyzer Option for VSE/Power (PWR):

'prefix.MICS.PARMS(INPUTPWR)'
'prefix.MICS.PARMS(PWRCONFG)'
'prefix.MICS.PARMS(PWRGROUP)'
'prefix.MICS.PARMS(PWRGPRTE)'
'prefix.MICS.PARMS(PWROPS)'
'prefix.MICS.PARMS(PWRTNRTE)'

__ CA MICS System Reliability Analyzer Option (SRL):

'prefix.MICS.PARMS(INPUTSRL)'
'prefix.MICS.PARMS(SRLDEVS)'
'prefix.MICS.PARMS(SRLOPS)'

__ CA MICS Tandem Option (TDM):

```
    'prefix.MICS.PARMS(INPUTTDM)'
    'prefix.MICS.PARMS(TDMOPS)'

__ CA MICS Tape Analyzer Option, TLM Component:

    'prefix.MICS.PARMS(INPUTTLM)'
    'prefix.MICS.PARMS(TLMOPS)'
    'prefix.MICS.PARMS(VOLGPRTE)'

__ Field Developed Applications (FDAs) for MSI:

    'prefix.MICS.PARMS(cccOPS)'

__ Field Developed Applications (FDAs) for MTI:

    'prefix.MICS.PARMS(cccOPS)'
    'prefix.MICS.PARMS(INPUTccc)'
    'prefix.MICS.PARMS(cccSYSID)'

__ CA MICS Analyzer Option for VM/CMS (VMC):

    'prefix.MICS.PARMS(INPUTVMC)'
    'prefix.MICS.PARMS(VMCOPS)'

****************************************************
* --------------INPUT SOURCE NOTE------------------*
*                                                  *
* The following components may use SMF data as an  *
* input source:                                    *
*        AST, CIC, DB2, HSM, IDM, MQS, RMF,        *
*        SMF, SNT, VTS, TSO, VCA, WEB, and         *
*        FDAs (if applicable)                      *
*                                                  *
* For the components listed above, you must choose *
* one of the alternative methods shown below.      *
* These specifications identify the data set names *
* containing the raw data that will be processed   *
* by the unit's DAILY update job.                  *
*                                                  *
*   (1) If DAYSMF ON is specified in               *
*       prefix.MICS.PARMS(JCLDEF), then you must   *
*       identify the input data set names in       *
*       prefix.MICS.PARMS(INPUTRDR).               *
*                                                  *
*   (2) If DAYSMF OFF is specified in              *
*       prefix.MICS.PARMS(JCLDEF) you must         *
*       identify the input data set names in       *
*       prefix.MICS.PARMS(INPUTccc) for each       *
*       component in the unit.                     *
*                                                   *
*   (3) If this component is in a unit by itself,  *
*       then you must identify the input data set  *
*       names in prefix.MICS.PARMS(INPUTRDR).      *
*                                                   *
*****************************************************

__ CA MICS CA ASTEX Option (AST):

    'prefix.MICS.PARMS(INPUTAST)'
    'prefix.MICS.PARMS(ASTAURT)'
    'prefix.MICS.PARMS(ASTOPS)'

__ CA MICS Analyzer Option for CICS (CIC):

    'prefix.MICS.PARMS(INPUTCIC)'
    'prefix.MICS.PARMS(CICALONE)'
    'prefix.MICS.PARMS(CICAURT)'
    'prefix.MICS.PARMS(CICMSAC)'
    'prefix.MICS.PARMS(CICOPS)'
    'prefix.MICS.PARMS(CICRLRT)'
    'prefix.MICS.PARMS(CICTHRSH)'

__ CA MICS Analyzer Option for DB2 (DB2):

    'prefix.MICS.PARMS(INPUTDB2)'
    'prefix.MICS.PARMS(DB2OPS)'

__ CA MICS Analyzer Option for CA IDMS (IDM):

    'prefix.MICS.PARMS(INPUTIDM)'
    'prefix.MICS.PARMS(IDMAURT)'
    'prefix.MICS.PARMS(IDMOPS)'
    'prefix.MICS.PARMS(IDMRLRT)'
    'prefix.MICS.PARMS(IDMRPTS)'

__ CA MICS Analyzer Option for MQSeries (MQS):

    'prefix.MICS.PARMS(INPUTMQS)'
    'prefix.MICS.PARMS(MQSOPS)'

__ CA MICS Hardware and SCP Analyzer (RMF):

    'prefix.MICS.PARMS(INPUTRMF)'
    'prefix.MICS.PARMS(RMFOPS)'

__ CA MICS Batch and Operations Analyzer (SMF):

    'prefix.MICS.PARMS(INPUTSMF)'
    'prefix.MICS.PARMS(JOBGROUP)'
    'prefix.MICS.PARMS(JOBGPRTE)'
    'prefix.MICS.PARMS(PRINTDEF)'
    'prefix.MICS.PARMS(SMFOPS)'
    'prefix.MICS.PARMS(SMFUOPTS)'
    'prefix.MICS.PARMS(TURNRTE)'

__ CA MICS Tape Analyzer Option, VTS Component:

    'prefix.MICS.PARMS(INPUTVTS)'
    'prefix.MICS.PARMS(VTSOPS)'
    'prefix.MICS.PARMS(VTSCMPID)'

__ CA MICS Network Analyzer Option (SNT):

    'prefix.MICS.PARMS(INPUTSNT)'
    'prefix.MICS.PARMS(NETGROUP)'
    'prefix.MICS.PARMS(NETGPRT)'
    'prefix.MICS.PARMS(NETAPPL)'
    'prefix.MICS.PARMS(SNTCIDRT)'
    'prefix.MICS.PARMS(SNTNIDRT)'
    'prefix.MICS.PARMS(SNTOPS)'
    'prefix.MICS.PARMS(SNTPLURT)'
    'prefix.MICS.PARMS(SNTRPTD)'
    'prefix.MICS.PARMS(SNTRPTM)'

__ CA MICS Analyzer for TSO:

    'prefix.MICS.PARMS(INPUTTSO)'
    'prefix.MICS.PARMS(TSOAURT)'
    'prefix.MICS.PARMS(TSOOPS)'

__ CA MICS Space Analyzer Option (VCA):

    'prefix.MICS.PARMS(INPUTHSM)'
    'prefix.MICS.PARMS(INPUTVCA)'
    'prefix.MICS.PARMS(HSMOPS)'
    'prefix.MICS.PARMS(VCAOPS)'

__ CA MICS Web Analyzer Option (WEB):

    'prefix.MICS.PARMS(INPUTWEB)'
    'prefix.MICS.PARMS(WEBOPS)'
    'prefix.MICS.PARMS(WEBSLIST)'
    'prefix.MICS.PARMS(WLECONF)'
    'prefix.MICS.PARMS(WLESCRPT)'

 __ Field Developed Applications (FDAs) using SMF data
    as input source:
```

```
    'prefix.MICS.PARMS(INPUTccc)'

***IMPORTANT****IMPORTANT***IMPORTANT****IMPORTANT***
*                                                   *
*  FDAs may or may not use SMF data as an input     *
*  source.                                          *
*                                                   *
*  CA MICS does NOT provide an automated interface  *
*  to CA SMF Director from within an FDA.  This     *
*  means that CA MICS will NOT be able to read the  *
*  CA SMF Director duplicate split index and obtain *
*  the necessary input files.  However, any SMF data*
*  sets output from CA SMF Director may still be    *
*  directly input as standard SMF data.             *
*                                                   *
***IMPORTANT****IMPORTANT***IMPORTANT****IMPORTANT***

+---------------------------------------------------+
|  References:  Section 2.3.2, all subsections      |
|               Section 2.3.3.2, all subsections    |
|               Chapter 7, ccc Analyzer Guide(s)    |
|               Section 7.6, System Modification    |
|                 Guide (for MSI)                   |
|               Section 8.7, System Modification    |
|                 Guide (for MTI)                   |
|               Section 1.5.2.3, CA ASTEX Option    |
|                 Guide                             |
|               Section 7.2.2, Application          |
|                 Extension Option for Tandem Guide |
+---------------------------------------------------+
```

__ 10. Review EXECDEF

Review the default values for the dynamic execution options in prefix.MICS.PARMS(EXECDEF) and update as needed. Carefully review the entries affecting data retention limits.

```
+---------------------------------------------------+
| References: Section 2.3.5                         |
|             Section A.3                           |
|             Section 7.3.6 Batch and Operations    |
|              Analyzer Guide                       |
+---------------------------------------------------+
```

__ 11. Review/Modify UNITGEN Options

Review sharedprefix.MICS.PARMS(UNITGEN) for the options in effect.

```
+----------------------------------------------------+
|  Reference(s):  Section 5.6.1                      |
+----------------------------------------------------+
```

__ 12. Execute UNITGEN

Submit the following job:

```
  'prefix.MICS.CNTL(UNITGEN)'
```

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

If the UNITGEN NOAUTOSUBMIT, or UNITGEN AUTOSUBMIT and NOOPJCL parameters have been specified, UNITGEN will complete with a normal return code of 4. Otherwise, UNITGEN and UNITGEN1 should complete with a return code of 0.

__ 13. (CONDITIONAL) Execute UNITGEN1

Perform this step only if your installation has specified

UNITGEN NOAUTOSUBMIT

You must manually submit the UNITGEN1 job to perform the required JCL and parameter generations.

After the UNITGEN job from the previous step completes successfully, submit the following job:

```
  'prefix.MICS.CNTL(UNITGEN1)'
```

__ 14. Verify UNITGEN1 Results

Verify that the UNITGEN1 job executed successfully.

The UNITGEN1 job contains multiple steps that perform COPYLIBU, CYCLEGEN, JCLGEN, cccPGEN, and GDGSGEN processing or part of it.

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

All steps should complete with a return code of zero.

If errors are encountered during UNITGEN1 job execution, correct the errors and rerun UNITGEN and UNITGEN1 jobs respectively.

```
+---------------------------------------------------+
|  Reference(s):  Sections 5.6 and 5.6.1            |
+---------------------------------------------------+
```

__ 15. Execute JCLGEND

Perform this step only if your installation has specified

UNITGEN NOOPJCL

If NOOPJCL is specified, UNITGEN1 did not generate the CA MICS operational jobs.

To generate these jobs, submit the following job:

```
  'prefix.MICS.CNTL(JCLGEND)'
```

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

All steps should complete with a return code of zero.

```
+----------------------------------------------------+
|  Reference(s):  Section 5.6.1                      |
+----------------------------------------------------+
```

__ 16. (CONDITIONAL) Define Accounting Rates

If Accounting and Chargeback is installed, use the CA MICS Workstation Facility (MWF) to complete the RATETBL panels. Select option 4;2;3;6 from the MWF primary options panel. The JCL for the MAGRUN will be displayed automatically if any of the RATETBL specifications are altered.

NOTE: If the logical SYSID is checked before assigning a comp code value, review the ACTFnnnn members in sharedprefix.MICS.SOURCE, where 'nnnn' is the number of an accounting comp code. If the CPU processor or processor speed is used in determining your site's accounting logic, then logic for the new CPU or LPAR will need to be added.

__ 17. Final Review

The system is fully generated and ready for check-out. Make final review of all definitions and JCL, applying any final user modifications at this time. Also review the prefix.MICS.USER.SOURCE(#BASEXIT) member. Determine if the _USRSEL exit contains any references to ORGSYSID. Update the exit if required. Optionally, review the #cccEXIT members in prefix.MICS.USER.SOURCE, where 'ccc' is the component identifier. Determine whether the user exits in these members contain any reference to ORGSYSID or SYSID. Update each exit as required.

__ 18. Resume Operations

Enable CA MICS operational jobs for this unit database. Use the Operational Status and Tracking RESUME command in MWF to enable CA MICS operational jobs.

__ 19. Resume Daily Processing

Use the Operational Status and Tracking DAILY command in MWF to resume daily processing.

```
+----------------------------------------------------+
| Reference(s): Section 3.5.7                        |
+----------------------------------------------------+
```

```
*************************************************************
*  Repeat the above steps for each database unit that will  *
*  process data from the new SYSID or LPAR.                 *
*************************************************************
```

__ 20. (OPTIONAL) Execute EXTFILE

If your site uses external files for processing, select option 4;2;3;7 from the MWF primary option panels. Examine the algorithms and rates for external files. If any external file specifications are added or changed, submit the following:

```
Submit 'sharedprefix.MICS.CNTL(EXTFILE)'
```

__ 21. Notify Affected Users

Notify the affected users that a new SYSID or LPAR has been added to CA MICS so that they can account for the difference in CPU trend reports, chargeback, or database inquiries.

5.9 Moving CA MICS Products to a Different Unit

Several processes MUST be performed in order to transfer an installed and operational CA MICS product and its data files to another unit. The following checklists provide directions for accomplishing the transfer:

1 - Make Advanced Preparations
2 - Install the Product in the Target Unit
3 - Transfer Data Files and Checkpoint Information to the Target Unit
4 - Transfer Tape Files to the Target Unit
5 - Finalize the Process

The following conventions will be used throughout these checklists:

original_unit = The unit from which the CA MICS product will be moved

original_prefix = The high-level qualifier of the original unit

original_tapeprefix = The high-level qualifier of the tape files in the original unit

target_unit = Destination for the product. This can be a new or an existing unit

target_prefix = The high-level qualifier of the destination unit

target_tapeprefix = The high-level qualifier of the tape files in the destination unit

CONSIDERATIONS

This chapter considers Date Integration Applications only. It does not apply to the Management Support Application products.

All associated Accounting journal files will also need to be moved with the appropriate Date Integration Application.

If you are going to transfer products from this unit to another unit then these reporting mechanisms must be updated to reflect the identifier of the new unit. These changes include PROC names (e.g., MICSSHRu) and DD name macros (e.g.,

```
&uBATX for the detail time-span), where "u" is the identifier
of the unit database.
```

5.9.1 Make Advanced Preparations

__ 1. Establish Time Frame

Establish a schedule and time frame for accomplishing the transfer. We strongly recommend the following:

__ Wait until monthly processing has been completed in the affected units before executing the transfer (particularly if the CA MICS product has an accounting interface)

__ Execute a DAILY as the first operational job in both units after the transfer has been completed, since week-to-date and month-to-date files are built during the daily process.

__ 2. Notify Users

Alert all CA MICS users that you will be moving the product well in advance of the actual day.

__ Inform all affected personnel that the original_unit will be undergoing changes on a certain date and may be unavailable for use.

__ Notify your users that the database identifier and the data set names that are used in their JCL, report code, user-level MICF inquiries, and database macros must be changed for the product.

__ 3. Determine DASD Amount

Immediately, prior to installing the product in the target_unit, determine the amount of DASD space allocated to the product in the original_unit, as follows:

__ Print the prefix.MICS.PARMS(DBMODEL) member.

__ Review prefix.MICS.PARMS(DBSPLIT) to determine if any of the product's information area files are included in a split timespan.

__ Create a SAS PROC DATASETS job in the original_unit to determine the total number of tracks used by the CA MICS product's files in each timespan (including any split timespans). Use the printout of the DBMODEL member to assist you in identifying the information areas and file names for the product.

Use the following job as a example:

```
//DATASETS EXEC MICSSHRx
//SYSIN    DD *
  PROC DATASETS DD=timespan;
  PROC DATASETS DD=timespan;
  PROC DATASETS DD=timespan;
```

where x is the database identifier of the original_unit, and timespan equals DETAIL, DAYS, WEEKS, MONTHS, YEARS (if active) or the DD name of any split timespan (e.g., DETAIL2).

__ Add a valid job card (and PROCLIB card, if required) and submit the job. Ensure that the job completes with a condition code of zero.

__ Print and save the output of the DATASETS job for use in later steps of this checklist.

__ 4. Install the Products

Complete the "Install the Products in the Target_Unit" checklist in section 5.9.2.

5.9.2 Install the Product in the Target Unit

__ 1. (CONDITIONAL) Add a New Product

Complete the "Add a New Product to an Existing Database Unit" checklist, Section 3.8.2

OR

Complete the "Add a New Product to a New Database Unit" checklist, Section 3.8.3.

__ 2. Incorporate Parameters

Incorporate parameters from the original_unit as appropriate. Review each of the product parameter members for changes during the installation process.

__ 3. (CONDITIONAL) Reenter Journal Files' Information

If this product interfaces with the CA MICS Accounting and Chargeback product, then you must REENTER the product's journal file rates, algorithms, and qualifications in the target_unit. The existing accounting specifications CANNOT be copied from the original_unit.

```
+-------------------------------------------------------------+
|  Reference(s):  Section 7.6, Accounting and Chargeback      |
|                              User Guide                     |
+-------------------------------------------------------------+
```

__ 4. Verify Timespans

Use the printout from the PROC DATASETS in the Advanced Preparation Checklist to verify that each timespan in the target_unit has enough space to accommodate the product that you are moving. Reallocate space as required.

__ 5. Review User Modifications

Review any unit-level user modifications that apply to the product. These may include (but are not limited to) exception processing code, accounting journal file exits, or user-defined formats in prefix.MICS.USER.LOAD. Make changes as required and incorporate your modifications into the structure of the target_unit.

__ 6. Update Administrator-written Report

Change any existing administrator-written report programs, to specify the target_prefix, target_tapeprefix, or database ID, as required.

__ 7. Regenerate MICF Production Inquiries

Use the MICF Catalog Administration function under MWF to regenerate any MICF production jobstream inquiries so that they reference the target_unit.

__ 8. Back Up Libraries (Target Unit)

After completing the installation of the product in the target_unit, back up the complex and the target_unit's system libraries (e.g., CNTL, PARMS, ASM, USER.SOURCE, USER.LOAD).

__ 9. Transfer Files and Checkpoint Information

Complete the "Transfer Data Files and Checkpoint Information to the Target_Unit" checklist in Section 5.9.3.

5.9.3 Transfer Data Files and Checkpoint Information to the Target Unit

__ 1. Ensure Backup

Ensure that a current backup tape exists for the original and target_unit (if active); otherwise submit the unit's BACKUP job. Ensure that the job completes with a condition code of zero and that all timespans have been successfully copied.

__ 2. Suspend Production Processing (Original and Targeted Unit)

Suspend production operations for the unit database by using the Operational Status and Tracking SUSPEND command in MWF to deactivate production scheduling.

Suspend production processing in the original_unit and the target_unit if it is active.

__ 3. Create Online Database Copy Job

Create a job in the original_unit to copy all of the product's online database files to the target_unit. Use the JCL below as an example. Add DD and PROC COPY statements to the job if any of the product's files reside in a split timespan.

NOTE: If the product interfaces with the CA MICS Accounting and Chargeback product, invoke the MICS Workstation Facility (MWF). Use the following options: Management Support Applications, Accounting and Chargeback, Parameter Specification and Generation, Rate Tables (4;2;3;6) to copy the product journal files, where ff is the file identifier.

For more information about the Accounting Journal Files (ACTJxx), refer to the Accounting and Chargeback User Guide, Appendix C.2.

```
//         EXEC MICSSHRx,SYSPARM=
//NDETAIL  DD  DISP=OLD,DSN=target_prefix.MICS.DETAIL
//NDAYS    DD  DISP=OLD,DSN=target_prefix.MICS.DAYS
//NWEEKS   DD  DISP=OLD,DSN=target_prefix.MICS.WEEKS
//NMONTHS  DD  DISP=OLD,DSN=target_prefix.MICS.MONTHS
//NYEARS   DD  DISP=OLD,DSN=target_prefix.MICS.YEARS
  PROC COPY IN=DETAIL OUT=NDETAIL;
    SELECT iii:  iii: ;
    SELECT ACTJff: ;              (if ACT interface)
  PROC COPY IN=DAYS OUT=NDAYS;
    SELECT iii:  iii: ;
  PROC COPY IN=WEEKS OUT=NWEEKS;
    SELECT iii:  iii: ;
  PROC COPY IN=MONTHS OUT=NMONTHS;
    SELECT iii:  iii: ;
  PROC COPY IN=YEARS OUT=NYEARS;  (if active)
    SELECT iii:  iii: ;
```

where x is the identifier of the original_unit, target_prefix is the high-level qualifier of the target_unit, and iii is each information area within the product that you are moving.

__ 4. (CONDITIONAL) Deactivate Accounting and Chargeback

If this is the only product that interfaces with Accounting and Chargeback, then edit the DAY199 step of the DAILY job in the original_unit as follows:

```
SYSPARM=NODATA
```

NOTE: Accounting and Chargeback should not be deactivated until the accounting months-end processing has occurred.

__ 5. Execute Copy Job Created in a Previous Step

Add the job card (and PROCLIB card, if required) to the JCL and submit the job. Ensure that it completes with a condition code of zero.

__ 6. (CONDITIONAL) Rename IMS Suspend Files

If you are moving the IMS product and using the suspend file, rename the following input and output suspend files to the target_prefix:

```
target_prefix.MICS.IMSSUS1
target_prefix.MICS.IMSSUS2
```

Use the ISPF option 3.2, option 3.4 or the TSO RENAME command.

__ 7. (OPTIONAL) Print CHECKPOINT Data Set

Print the prefix.MICS.CHECKPT.DATA data set in the original_unit in order, in the event recovery is needed.

__ 8. (CONDITIONAL) Define Accounting Rates

If Accounting and Chargeback is installed, use the CA MICS Workstation Facility (MWF) to complete the RATETBL panels. Select option 4;2;3;6 from the MWF primary options panel. The JCL for the MAGRUN will be displayed automatically if any of the RATETBL specifications are altered.

NOTE: If the logical SYSID is checked before assigning a comp code value, review the ACTFnnnn members in sharedprefix.MICS.SOURCE, where 'nnnn' is the number of an accounting comp code. If the CPU processor or processor speed is used in determining your site's accounting logic, then logic for the new CPU or LPAR will need to be added.

__ 9. Modify CHECKPOINT

__ Use ISPF EDIT, COPY and PASTE commands to copy the checkpoint lines from the original_unit to target_prefix.MICS.CHECKPT.DATA.

These lines should apply to the product as installed in the target_unit: SYSIDs, subsystems, data sources, time-stamps, etc.

__ Edit target_prefix.MICS.CHECKPT.DATA to fit your organization's needs.

Ensure that your ISPF EDIT profile has the NUMBER OFF option in effect; this data set cannot have line numbers.

```
+------------------------------------------------------------+
| Reference(s): Section 4.3.4.4.2 (CHECKPT Command)          |
|               Section 4.3.7 (Checkpoint File)              |
+------------------------------------------------------------+
```

__ 10. Resume Operations (Targeted Unit)

Enable CA MICS operational jobs for this unit database. Use the Operational Status and Tracking RESUME command in MWF to enable CA MICS operational jobs.

__ 11. Execute BACKUP (Targeted Unit)

Submit target_prefix.MICS.CNTL(BACKUP). Ensure that it completes with a condition code of zero.

__ 12. Remove A Component

Complete the "Remove A Component from a Unit" checklist in Section 5.12.

__ 13. Transfer Tape Files

Complete the "Transfer Tape Files to the Target Unit" checklist in section 5.9.4.

5.9.4 Transfer Tape Files to the Target Unit

```
************************************************************
* Note: Ensure that the "Remove a Component From a Unit"   *
*       checklist, section 5.12, has been completed.       *
************************************************************
```

Determine if HISTW or HISTM has been specified in the prefix.MICS.PARMS(JCLDEF) member. Complete this checklist only if YES is specified for audit, weekly, or monthly archiving in the JCLDEF member of target_prefix.MICS.PARMS; otherwise, move to the checklist in section 5.9.6, Finalize the Process. Whatever you have defined for HISTW and HISTM will be replicated in the target_unit (unless you change them before completing the checklist).

The original_tapeprefix data sets for the product cannot be changed by simply renaming the catalogued data sets; the internal tape label will still carry the original_tapeprefix name. Therefore, you must modify your CA MICS operational JCL in order to copy the original_tapeprefix tape files to target_prefix tape files.

NOTE: The timing of this process is CRITICAL.

If the WEEKLY or MONTHLY job will run in the original_unit AFTER the jobs in the target_unit, point to the current (0) GDG level.

If the jobs in the original_unit will run BEFORE the jobs in the target_unit, change the GDG level to (-1) in the WEEKLYX and MONTHLYX jobs, since the current level of the history files in the original_unit will no longer contain data for the product that you have moved.

__ 1. If weekly history archive is active in the
       target_unit:

   __ Create target_prefix.MICS.CNTL(WEEKLYX). Copy in the JCL of the target_prefix.MICS.CNTL(WEEKLY) member.

   __ Modify the DD statements in the WEEKLYX member that begin with "//HIiiifff" in the WEEK300 step, where iii is the information area and fff is the file name. Change the DSN parameter to the original_tapeprefix.HISTW.iiifff(n) name for each of the product's files, where n is GDG level -1 or zero.

NOTE: Change only the DD statements that are associated with the product that you are moving.

WEEKLYX should be used for only the FIRST execution of the weekly job in the target_unit.

NOTE: Execution of this job after the first week will result in a loss of data!

Make sure that the normal WEEKLY job is executed in subsequent weeks.

__ 2. If monthly history archive is active in the target unit:

__ Create target_prefix.MICS.CNTL(MONTHLYX). Copy in the JCL of the target_prefix.MICS.CNTL(MONTHLY) member.

__ Modify the DD statements in the MONTHLYX member that begin with "//HIiiifff" in the MONTH300 step, where iii is the information area and fff is the file name. Change the DSN parameter to the original_tapeprefix.HISTM.iiifff(n) name for each of the product's files, where n is GDG level -1 or zero.

NOTE: Change only the DD statements that are associated with the product that you are moving.

MONTHLYX should be used for only the FIRST execution of the monthly job in the target_unit.

NOTE: Execution of this job after the first month will result in a loss of data!

Make sure that the normal MONTHLY job is executed in subsequent months.

__ 3. If audit archiving is performed for the product:

__ Consider copying the weekly audit tapes, named original_tapeprefix.MICS.AUDIT.iiifff(n), to the target unit, where n is GDG level -1 or zero.

NOTE: No retrofit logic is required in order to copy the tapes.

```
+-----------------------------------------------------------+
| Reference(s): Section 2.3.3.3.2.1.5, Database Unit Tape  |
|               Specifications                              |
+-----------------------------------------------------------+
```

__ If you choose NOT to copy these files, inform your users that the audit archive files for the product that were created prior to the transfer will remain under the original_tapeprefix name.

__ 4. Finalize Process

Complete the "Finalize the Process" checklist in Section 5.9.5.

5.9.5 Finalize the Process

__ 1. Notify that Product Transfer is Complete

Notify all users that the transfer of the product has been completed.

NOTE: Do not complete the following steps until the disposition of any audit archive files has been determined and the WEEKLYX and MONTHLYX jobs have completed in the target_unit.

__ 2. Uncatalog Archive Files

Uncatalog all original_tapeprefix audit and weekly/monthly history archive files for the product in the original_unit as applicable.

__ 3. Delete Tape Files

Delete the original_tapeprefix GDG base entries in the original_unit for the product's audit, and weekly/monthly history tape files as applicable.

__ 4. Back Up All Libraries

The code and text for the products on the PSP distribution medium are copied directly to your sharedprefix.MICS and sharedprefix.MICS.PSP libraries. To avoid potential space problems, compress your sharedprefix.MICS and PSP libraries before loading the new product.

```
************************************************************
*  This chapter, "Moving a CA MICS Product to a Different  *
*  Unit," is completed.                                    *
************************************************************
```

5.10 Removing the DAYSMF Step from the DAILY Job

During CA MICS DAILY processing, the DAYSMF step is used to split SMF data into separate data sets for each component that has SMF data as input. This helps to reduce I/O, CPU time, and execution time because each component processes only the SMF data it requires.

Instead of using the DAYSMF step in the CA MICS DAILY job, CA SMF Director or other processes can be used to split the SMF data into discrete data sets outside of the CA MICS DAILY job.

CA MICS supports this configuration by allowing the DAYSMF step to be removed from the DAILY job and allowing the installation to specify the input data sets of the SMF data for processing by each component.

The following checklist shows the steps necessary to deactivate the DAYSMF step and manually specify the input data sets for components that use SMF data.

__ 1. Determine components in the unit that use SMF as an input data source

MICSLOG message BAS10083 in any JCLGEN lists the components that have SMF as an input source. The following will regenerate the JCLGENU job with JCLGEN to identify the components that use SMF data as input.

Edit prefix.MICS.PARMS(JCLGENU) and make sure it has the following line in it:

```
JCLGENU
```

Save prefix.MICS.PARMS(JCLGENU).

Submit prefix.MICS.CNTL(JCLGENU). This job should complete with a return code of zero. Review the MICSLOG and locate message BAS10083. This message lists the components that use SMF data as input.

__ 2. Deactivate the DAYSMF step

Edit prefix.MICS.PARMS(JCLDEF) and find the DAYSMF keyword and change it to:

```
DAYSMF OFF
```

__ 3. Specify input data set for each component

For each defined component that has SMF data as an input source, a prefix.MICS.PARMS(INPUTccc) member must exist. If the INPUTccc member does not exist, it must be created or the CA MICS DAILY job will not generate.

For each INPUTccc, a DD for INPUTSMF or SMFDRCTR must be added or modified. Use this example for SMF input:

```
//INPUTSMF DD DISP=SHR,DSN=SMF.SPLIT.DATASET
```

Use this example for CA SMF Director input:

```
//SMFDRCTR DD DISP=SHR,DSN=CA.SMF.duplicate.index
```

Note: When running with DAYSMF OFF, prefix.MICS.PARMS(INPUTRDR) will be used to specify the input data source for the component step if it is the only remaining SMF eligible step. That is, the only component in the unit using SMF as the input source.

__ 4. Generate the DAILY job

Edit prefix.MICS.PARMS(JCLGENU) and add the following line:

```
DAILY
```

Edit prefix.MICS.CNTL(JCLGENU) and submit the job for execution. This job should complete with a return Code of zero.

If the JCLGENU fails with a user abend of 998 and the following message:

```
BAS10117E MEMBER INPUTccc INDICATED ON A ":INCLUDE" OR
          A ":INCLUDEPARM" STATEMENT NOT IN LIBRARY
```

then the INPUTccc member was not created per step 3. In this case, create the member; then run the prefix.MICS.CNTL(JCLGENU).

__ 5. (Optional) Delete SMF work data sets

When the DAYSMF step is present, an SMF work data set, prefix.MICS.ccc.DATA (where ccc is the component

```
          identifier), is created for each component that uses
          SMF data. This data set is either temporary or
          permanent, depending on the value of JCLDEF keyword
          DAYSMF.

          Since the DAYSMF is no longer being used, these
          data sets are not required and may be deleted.


   Note:  If the input data set names for a component change,
   the INPUTccc members, or INPUTRDR, can be modified and the
   DAILY regenerated using JCLGENU.
```

5.11 Delete a Unit Database from a Complex

If you need to transfer components that were installed in this unit prior to its deletion, complete the "Moving CA MICS Products to a Different Unit" checklists in section 5.9.

__ 1. Establish Time Frame

Establish a schedule and time frame for accomplishing the deletion. If CA MICS Accounting and Chargeback is installed in the unit, we strongly recommend that you wait until monthly processing has been completed in the unit before proceeding.

__ 2. Determine Reporting Affects

Determine how the removal of the unit will affect the following reporting mechanisms:

o Shared/Private MICF inquiries
o MICF production reporting jobstreams
o JCL for batch reporting jobs
o Adhoc SAS reporting programs

Any references to the deleted unit database must be removed from these reporting mechanisms.

Note: Existing shared and private inquiries and standard applications also include those affiliated with the Management Support Application components.

__ 3. Alert All CA MICS Users

Alert all CA MICS users that you will be deleting the unit database well in advance of the actual day and inform them of the changes that need to be made. The CA MICS System Administrator is responsible for the changes to the Shared MICF inquiries and MICF reporting jobstreams.

__ 4. Back Up All Libraries

The code and text for the products on the PSP distribution medium are copied directly to your sharedprefix.MICS and sharedprefix.MICS.PSP libraries. To avoid potential space problems, compress your sharedprefix.MICS and PSP libraries before loading the new product.

__ 5. Edit DBTABLE

Edit sharedprefix.MICS.GENLIB(DBTABLE) as follows:

__ Delete the line for the unit database that you are removing from the complex.

__ SAVE the changes that you have made to DBTABLE.

__ 6. (CONDITIONAL) Designate Primary Unit

Review the DATABASE statement in member prefix.MICS.PARMS(JCLDEF) for the unit that you are removing. If the DATABASE statement at the top of this member contains the word "PRIMARY," then you must designate one of your other unit databases as the new primary unit in this step.

You have the option to install a new unit database to become your new primary unit database; see section 3.8.4. If you choose this option, install the new unit at this point, that is, before continuing in this checklist. On the new unit's DATABASE statement, specify "UNIT" as the database type at this time.

The primary unit manages the backup and restoration of the following sharedprefix data sets:

```
    sharedprefix.MICS.ISPTLIB
    sharedprefix.MICS.SCREENS
    sharedprefix.MICS.TABLES
    sharedprefix.MICS.CAPACITY (if CAP installed)
```

Complete this step ONLY if you are deleting the primary unit you need to designate an existing (that is, installed and active) unit database as the new primary.

Designate the new primary unit by completing the following steps:

```
+----------------------------------------------------+
| Reference(s): Section 2.3.3.2.1 (JCLDEF)           |
+----------------------------------------------------+
```

__ Delete the first line in DBTABLE, which should read as follows:

```
*: d
```

where 'd' is the primary unit database identifier.

__ Delete the line that references the replacement unit. That line will be regenerated in the next step.

__ Designate a new primary unit by editing the prefix.MICS.PARMS(JCLDEF) member of any remaining production-level unit (i.e., contains the word "UNIT" in the DATABASE statement). Do NOT designate a SPECIAL or TEST unit as your new primary unit. These unit types cannot support the processing and data retention requirements of a production-level unit. Change the word "UNIT" in the DATABASE statement to "PRIMARY" and then SAVE the JCLDEF member.

__ Edit prefix.MICS.PARMS(JCLGENU) in the new primary unit. Replace the member with a line that reads:

```
BACKUP RESTORE
```

__ Submit prefix.MICS.CNTL(JCLGENU) in the new primary unit to regenerate the BACKUP and RESTORE jobs. The RSTRTBLS and RSTRTLIB jobs will also be regenerated during this process.

Ensure that there are no error messages in MICSLOG and that the last message is the normal termination message, BAS10999.

__ 7. Activate the Removal of the Unit

Regenerate a jobstream at the complex-level to activate the removal of the unit database as follows:

__ Edit sharedprefix.MICS.PARMS(JCLGENUC) and replace the member with a line that reads:

```
JCLGENUC
```

__ Submit sharedprefix.MICS.CNTL(JCLGENUC) to reconfigure CA MICS and to remove the unit.

Ensure that there are no error messages in the MICSLOG and that the last message is the normal termination message, BAS10999.

__ 8. Invoke MWF

Invoke the CA MICS Workstation Facility (MWF) and perform the following steps. Note that your TSO userid must be authorized to make changes in Options 2, 4, and 5 of the workstation.

__ Select option 5, MICS Administrator Facility (MICF). Type the word DBCLEAN on the command line and press ENTER.

__ The Database Table Clean panel will be presented. CAREFULLY type an 'S' next to the unit that you want to delete. Press ENTER. This command will remove MWF references to the deleted unit.

__ Return to the MWF Primary options panel when the DBCLEAN command has completed.

__ Verify that the unit was deleted. Type the word CONFIG on the command line and press ENTER. The CONFIG command will produce a list of active units at your installation. Ensure that there are no references to the deleted unit.

Note: If the unit you were trying to delete is still listed, STOP. Contact Technical Support for assistance.

__ Exit MWF.

__ 9. (CONDITIONAL) Determine if ARCHIVE is Active

Review the ARCHIVE statement in prefix.MICS.PARMS(JCLDEF) member for the unit that you are removing. If ARCHIVE YES is specified for either the AUDIT, HISTW, or HISTM keywords then complete this step.

__ Determine what will be done with the audit and history archive files. If you do not want to retain these tapes, uncatalog the following unit-level data sets:

```
tapeprefix.MICS.AUDIT.iiifff.GnnnnV00
tapeprefix.MICS.HISTW.iiifff.GnnnnV00
tapeprefix.MICS.HISTM.iiifff.GnnnnV00
```

where 'iiifff' is each CA MICS information area file that was selected for audit or history archive. ISPF option 3.4 can be used to uncatalog the tape files. You may also want to update your tape management system to expire these tape files.

__ 10. Delete Unit-Level Data Sets

Delete the following unit-level data sets for the unit if they exist:

```
prefix.MICS.ccc.DATA      (DAYSMF files are PERMANENT)
prefix.MICS.CHECKPT.DATA
prefix.MICS.CNTL
prefix.MICS.DAYS
prefix.MICS.DAYSn         (DBSPLIT)
prefix.MICS.DETAIL
prefix.MICS.DETAILn       (DBSPLIT)
prefix.MICS.IMSSUS1       (IMS)
prefix.MICS.IMSSUS2       (IMS)
prefix.MICS.MODEL         (allocated)
prefix.MICS.MONTHS
prefix.MICS.MONTHSn       (DBSPLIT)
prefix.MICS.MUOLIB
prefix.MICS.PARMS
prefix.MICS.RESTART.CNTL
prefix.MICS.SPECIAL.SOURCE (SPECIAL DB)
prefix.MICS.USER.LOAD
prefix.MICS.USER.SOURCE
prefix.MICS.WEEKS
prefix.MICS.WEEKSn        (DBSPLIT)
prefix.MICS.YEARS
prefix.MICS.YEARSn        (DBSPLIT)
```

where 'ccc' is the component identifier and 'n' represents a hexadecimal value from 0 to z.

__ 11. (CONDITIONAL) If Accounting and Chargeback Installed

If the CA MICS Accounting and Chargeback component was installed in the unit that you are deleting, perform the following steps; otherwise proceed to the next step.

__ Delete all members in sharedprefix.MICS.ISPTLIB with names ACTJffx, where ff is any journal file identifier and x is the database identifier of the original_unit.

__ Edit prefix.MICS.PARMS(JCLGENU) IN ONE OF YOUR REMAINING UNITS. Replace the member with a line that reads:

```
JCLGENU
```

__ Submit prefix.MICS.CNTL(JCLGENU) from the same unit. Ensure that there are no error messages in the MICSLOG and that the last message is the normal termination message, BAS10999.

__ Determine what will be done with the daily and monthly accounting audit tapes from this unit. The data set names are:

```
tapeprefix.MICS.ACCT.DAY1.GnnnnV00
tapeprefix.MICS.ACCT.DAY2.GnnnnV00
tapeprefix.MICS.ACCT.MONTH.GnnnnV00
```

Even though they are not used in the chargeback process, it may be necessary to retain these tapes for audit purposes depending on your site's business requirements. The DAY2 tape is a copy of the last DAY1 tape so it should not be needed. The latest generation of the DAY1 tape contains the month-to-date charges for the month up until the time the unit ceased operation; the latest generation of this tape may need to be retained for audit purposes. Each generation of the MONTH tape contains charges for one accounting month. These generations may also need to be retained for audit purposes.

__ 12. (CONDITIONAL) Delete User PROCLIB

If a user PROCLIB existed for the deleted unit, it can now be deleted. If a common PROCLIB is used, delete the procedures for the deleted unit from the common procedure library.

__ 13. Back Up All Libraries

The code and text for the products on the PSP distribution medium are copied directly to your sharedprefix.MICS and sharedprefix.MICS.PSP libraries. To avoid potential space problems, compress your sharedprefix.MICS and PSP libraries before loading the new product.

5.12 Remove a Component from a Unit

```
************************************************************
* Note:  The use of 'prefix' in this checklist is the      *
*        high-level qualifier of the unit from which the   *
*        CA MICS component will be removed.                *
*                                                          *
*        The use of original_unit in this checklist is the *
*        same as the unit from which you are removing the  *
*        components.                                       *
************************************************************
```

Several steps must be performed in order to successfully remove a component from a unit. These steps include editing some members, deleting members that are associated with the component and executing generation jobs for the unit.

The steps in this checklist should be completed only after all monthly processing has been completed for the unit. This includes MICF Production Reporting and User Reporting jobstreams. It ensures that all valid accounting and capacity data for the month has been processed right up to the removal event. We recommend that you inform all CA MICS users that you will be removing this component well in advance of the actual day. This notification will alert your users to the fact that the complex will be undergoing changes on a certain date and that this unit may be found unavailable during this process.

If the unit database contains only the component that is being removed (or it contains only the component and ACT) then you may want to complete the "Delete a Unit from a Complex" checklist in section 5.11 instead of this checklist.

__ 1. Edit JCLDEF

Edit prefix.MICS.PARMS(JCLDEF) in the original_unit. Remove the identifier of the component from the COMPONENT statement and also from the SMFRECORDING statement (if coded).

__ 2. (CONDITIONAL) Edit JCLDEF

If this component is the only component that interfaces with Accounting and Chargeback, and the accounting months-end processing has occurred, edit JCLDEF. Remove the ACT identifier from the COMPONENT statement.

__ 3. (CONDITIONAL) Edit DBMODEL

If the original_unit is not a special database unit, edit prefix.MICS.PARMS(DBMODEL). Delete all FILE and DATA statements associated with the component.

__ 4. Edit COMPRESS

Edit prefix.MICS.PARMS(COMPRESS). Delete all FILE and DATA statements associated with the component.

__ 5. Execute JCLGENB

Regenerate the CA MICS unit installation Jobs. Submit the following job:

```
  'prefix.MICS.CNTL(JCLGENB)'
```

```
+----------------------------------------------------+
| Reference(s): Section 3.5.3.3                      |
+----------------------------------------------------+
```

__ 6. (CONDITIONAL) Execute JCLGENF

If the original_unit is a special database unit, generate the special database installation jobs by submitting the following job:

```
  'prefix.MICS.CNTL(JCLGENF)'
```

```
__  7. (CONDITIONAL) Execute SELFILES

       If the original_unit is a special database unit, edit
       prefix.MICS.PARMS(SELFILES).  Delete all FILE and
       DATA statements associated with the component. Submit
       the following job:

         'prefix.MICS.CNTL(SELFILES)'

__  8. (CONDITIONAL) Edit EXECDEF

       If the original_unit is a special database unit, edit
       prefix.MICS.PARMS(EXECDEF).  Delete any identifier of
       the component.

__  9. Review UNITGEN Options

        Review sharedprefix.MICS.PARMS(UNITGEN) for the
        options in effect.

        +----------------------------------------------------+
        |  Reference(s):  Section 5.6.1                      |
        +----------------------------------------------------+
__ 10. Execute UNITGEN

       Submit the following job:

         'prefix.MICS.CNTL(UNITGEN)'

       Ensure that there are no error messages in MICSLOG or
       SYSTSPRT and that the MICSLOG contains the normal
       termination messages, BAS10999I or BAS11999I.

       If the UNITGEN NOAUTOSUBMIT, or UNITGEN AUTOSUBMIT and
       NOOPJCL parameters have been specified, UNITGEN will
       complete with a normal return code of 4.  Otherwise,
       UNITGEN and UNITGEN1 should complete with a return
       code of 0.
```

__ 11. (CONDITIONAL) Execute UNITGEN1

Perform this step only if your installation has specified

UNITGEN NOAUTOSUBMIT

You must manually submit the UNITGEN1 job to perform the required JCL and parameter generations.

After the UNITGEN job from the previous step completes successfully, submit the following job:

```
  'prefix.MICS.CNTL(UNITGEN1)'
```

__ 12. Verify UNITGEN1 Results

Verify that the UNITGEN1 job executed successfully.

The UNITGEN1 job contains multiple steps that perform COPYLIBU, CYCLEGEN, JCLGEN, cccPGEN, and GDGSGEN processing or part of it.

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

All steps should complete with a return code of zero.

If errors are encountered during UNITGEN1 job execution, correct the errors and rerun UNITGEN and UNITGEN1 jobs respectively.

```
+----------------------------------------------------+
|  Reference(s):  Sections 5.6 and 5.6.1             |
+----------------------------------------------------+
```

__ 13. (CONDITIONAL) Execute JCLGEND

Perform this step only if your installation has specified

```
UNITGEN NOOPJCL
```

If NOOPJCL is specified, UNITGEN1 did not generate the CA MICS operational jobs.

To generate these jobs, submit the following job:

```
  'prefix.MICS.CNTL(JCLGEND)'
```

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

All steps should complete with a return code of zero.

```
+-----------------------------------------------------+
|  Reference(s):  Section 5.6.1                       |
+-----------------------------------------------------+
```

__ 14. Update JCL in Production Library

Ensure that the JCL for the regenerated jobs is moved to the appropriate production library.

__ 15. Identify Component Files

Create a SAS PROC DATASETS job in the original_unit to identify the MICS component's files in each timespan (including any split timespans). Use the following job as a example:

```
//DATASETS EXEC MICSSHRx
//SYSIN DD *
PROC DATASETS DD=timespan;
PROC DATASETS DD=timespan;
PROC DATASETS DD=timespan;
```

where x is the database identifier of the original_unit, and timespan equals DETAIL, DAYS, WEEKS, MONTHS, YEARS (if active) or the DD name of any split timespan (e.g., DETAIL2).

__ Add a valid JOB statement (and PROCLIB statement, if required) and submit the job. Ensure that the job completes with a condition code of zero.

__ Print and save the output of the DATASETS job for use in the next step of this checklist.

__ 16. Delete All Online Database Files

Delete all of the online database files for the component in the original_unit, as follows:

__ Reference the output of the DATASETS job that was executed previously in this checklist. Indicate the component files that need to be deleted from each timespan.

__ When all timespans have been reviewed, create a job in the original_unit that deletes the files associated with the component. Use the following job as an example and code PROC DATASETS statements for each timespan in which the component is active.

NOTE: If the component has an accounting interface, also delete the component journal files in the original_unit. For more information about the Accounting Journal Files (ACTJxx), see the Accounting and Chargeback User Guide, section 7.1.

```
//DATASETS EXEC MICSDBx
//SYSIN    DD *
  PROC DATASETS DDNAME=timespan MEMTYPE=DATA
       NOWARN;
    DELETE iiifff01-iiifffnn iiifff01-iiifffnn
            iiifff01-iiifffnn iiifff01-iiifffnn
           ACTJff01-ACTJffnn;     (if required)
  PROC DATASETS DDNAME=timespan MEMTYPE=DATA
       NOWARN;
    DELETE iiifff01-iiifffnn iiifff01-iiifffnn
           iiifff01-iiifffnn iiifff01-iiifffnn;
```

where:

x   is the database identifier of the original_unit

timespan   equals DETAIL, DAYS, WEEKS, MONTHS, or YEARS (if active), and the DD name of any split timespan (e.g., DETAIL2)

iiifff   is the information area and file identifier for the component that is being moved

ff   is the file identifier

nn   is the highest cycle number of the file

__ Add a valid JOB statement (and PROCLIB statement, if required) to the JCL and submit the job. Ensure that the job completes with a condition code of zero.

__ 17. Edit CHECKPT DATA (Original Unit)

Edit prefix.MICS.CHECKPT.DATA in the original_unit. Delete the lines that contain checkpoint information for the component that is being moved. ADD the same number of blank lines to the member.

__ 18. Suspend Operations

Suspend production operations for the unit database by using the Operational Status and Tracking SUSPEND command in MWF to deactivate production scheduling.

__ 19. Back Up Libraries (Original Unit)

Back up the original_unit's libraries (e.g., CNTL, PARMS, USER.SOURCE, USER.LOAD) by submitting the following job:

```
'prefix.MICS.CNTL(BACKUP)'
```

Ensure that the job completes with a condition code of zero and that all timespans have been successfully copied.

__ 20. Resume Operations

Resume normal operations in this unit. Reactivate production processing in the original_unit.

5.13 Remove a Component from a Complex

Several steps must be performed in order to successfully remove a component from a complex. These steps include deleting members that are associated with the component and executing generation jobs to remove the component from all of the unit databases where it was installed.

This checklist comprises instructions for removing a component in accordance with contractual agreements with CA. Read this checklist in its entirety before proceeding with any of the steps. Note that completion of this checklist will permanently disable the component at your site; it is NOT intended as a temporary bypass to processing.

The steps in this checklist should be completed only after all monthly processing has been completed for each unit database that is affected. This includes MICF Production Reporting and User Reporting jobstreams. It ensures that all valid accounting and capacity data for the month has been processed right up to the removal event. We recommend that you inform all CA MICS users that you will be removing the component well in advance of the actual day. This notification will also alert your users to the fact that the complex will be undergoing changes on a certain date and that some units may be unavailable during this process.

If a unit database contains only the component that is being removed (or it contains only the component and ACT) then you may also want to complete the "Delete a Unit from a Complex" checklist in section 5.11 for that unit. If you want a current list of modules that may be deleted, contact CA Technical Support.

Perform the following steps to remove an installed CA MICS component from a CA MICS complex and all of its associated unit databases. If the component is installed in more than one complex, repeat the entire checklist for each complex that is affected.

__ 1. Backup All Libraries

The code and text for the products on the PSP distribution medium are copied directly to your sharedprefix.MICS and sharedprefix.MICS.PSP libraries. To avoid potential space problems, compress your sharedprefix.MICS and PSP libraries before loading the new product.

__ 2. (CONDITIONAL) Delete the CAP Data Set

If the CAP component is being removed, delete the sharedprefix.MICS.CAPACITY data set.

__ 3. Remove Component(s)

Complete the "Remove a Component from a Unit" checklist, Section 5.12 for each unit database that contains the component that is being removed.

__ 4. Execute DOCAGEN1

In order to regenerate the document tables for Document Access, submit the following job:

```
  'sharedprefix.MICS.CNTL(DOCAGEN1)'
```

This process may take some time and is dependent on the number of CA MICS components installed in the complex.

```
+----------------------------------------------------+
| Reference(s): Section 3.3.7 (Valid Return Codes)   |
+----------------------------------------------------+
```

__ 5. (CONDITIONAL) Update PSP Online Product Change Panels

If PSP Online has been previously activated, update the PSP libraries to reflect that the product has been removed from PSP Online as follows:

__ Delete @@ccc from sharedprefix.MICS.PSP.PC.TEXT.

__ Enter the CA MICS Workstation Facility. Select option 5, CA MICS System Administrator Functions (MAF). Select option 2;4 from the CA MICS Administrator Facility Panel.

__ Select option 1 (Receive PSP Tables) on the PSP Utilities Menu. Press END to re-receive the PSP tables. This process will remove product changes for the deleted component from the following PSP Online Product Change panels:

Apply Product Changes; Select Product Changes

The panels will retain information for all other installed components. This process will take some time; the length depends on the number of components licensed to your site.

__ When the receive process is completed, press ENTER to accept the new tables.

__ Exit PSP Online and MWF.

Note: The Status of Component panel will retain any historical information for the deleted component.

__ 6. Backup All Libraries

The code and text for the products on the PSP distribution medium are copied directly to your sharedprefix.MICS and sharedprefix.MICS.PSP libraries. To avoid potential space problems, compress your sharedprefix.MICS and PSP libraries before loading the new product.

__ 7. Notify CA MICS Users

Notify CA MICS users that information for the component is no longer available.

5.14 Vertical Checkpoint Expansion

Review your CA MICS configuration and if you still need to increase the checkpoint size, then complete the checklist below.

Note: This procedure assumes you have already completed all installation processing for this unit database and that the database is populated with valid data.

__ 1. Backup Checkpoint Data Set

Run a BACKUP operational job to backup all of your databases, especially your checkpoint data set.

__ 2. COPY Checkpoint Data Set

Manually copy the prefix.MICS.CHECKPT.DATA data set.

__ 3. Modify CKPTCNT Parameter

Modify the CKPTCNT parameter in prefix.MICS.PARMS(SITE). CKPTCNT must be a number between 100 and 9999. Specify no more checkpoint entries than you will reasonably need.

NOTE: A good rule of thumb is to set checkpoint size to twice the size of the original checkpoint. For example, if your original CKPTCNT specification is 100, then 200 is a reasonable size for your enlarged checkpoint. If you run out of checkpoint entries at 200, then try expanding the checkpoint size to 400 entries. Remember that the maximum checkpoint size is 9999.

__ 4. Execute BASPGEN

Execute the job in prefix.MICS.CNTL(BASPGEN) to validate your CKPTCNT specification and generate the %CKPTCNT macro for the unit database.

__ 5. Execute CKPTINIT

Execute the following job to resize your checkpoint data set:

```
'prefix.MICS.CNTL(CKPTINIT)'
```

__ 6. Resume Operations

Resume normal operational job processing.

5.15 Excluding a Component from the DAYSMF Step

During CA MICS DAILY processing, the DAYSMF step is used to split SMF data into separate data sets for each component that has SMF data as input. This helps to reduce I/O, CPU time and execution time because each component processes only the SMF data it requires.

Instead of using the DAYSMF step in the CA MICS DAILY job, CA SMF Director or other processes can be used to split the SMF data into discrete data sets outside of the CA MICS DAILY job.
CA MICS supports this configuration by allowing components to be excluded from the DAYSMF step on an individual basis.

The following checklist contains the steps necessary to exclude a component from the DAYSMF step.

__ 1. Determine components in the unit that use SMF or CA SMF Director as an input data source

MICSLOG message BAS10083 in any JCLGEN lists the components that have SMF and CA SMF Director as an input source. The following will regenerate the JCLGENU job with JCLGEN to identify the components that use SMF data as input.

Edit prefix.MICS.PARMS(JCLGENU) and make sure it has the following line in it:

```
JCLGENU
```

Save prefix.MICS.PARMS(JCLGENU).

Submit prefix.MICS.CNTL(JCLGENU). This job should complete with a return code of zero. Review the MICSLOG and locate message BAS10083. This message lists the components that use SMF data as input.

__ 2. Specify the components to exclude from DAYSMF

Edit prefix.MICS.PARMS(JCLDEF) and find the DAYSMF keyword and change it to:

```
DAYSMF EXCLUDE ccc ccc . . .
```

ccc is the 3 character identifier as defined on the COMPONENTS statement.

Note: If you wish to exclude all the SMF eligible components from the DAYSMF step, specify DAYSMF OFF and follow the checklist in section 5.10 of this guide.

Excluding a component from DAYSMF is not restricted to components that use CA SMF Director input, but can be any component that uses SMF input.

__ 3. Specify the input data set for the component step

For each defined component that has SMF data as an input source, a prefix.MICS.PARMS(INPUTccc) member must exist. If the INPUTccc member does not exist, it must be created or the CA MICS DAILY job will not generate.

For each INPUTccc, a DD for INPUTSMF must be added or, if the input data source is CA SMF Director, an SMFDRCTR DD must be added or modified. Use this example for SMF input:

```
//INPUTSMF DD DISP=SHR,DSN=SMF.SPLIT.DATASET
```

Use this example for CA SMF Director input:

```
//SMFDRCTR DD DISP=SHR,DSN=CA.SMF.duplicate.index
```

Note: Excluding a component (or multiple components) that results in a single component remaining as an SMF eligible component in this unit, will automatically deactivate the DAYSMF step. In this case, prefix.MICS.PARMS(INPUTRDR) will be used to specify the input data source for the remaining SMF component step.

__ 4. Generate the DAILY job

Edit prefix.MICS.PARMS(JCLGENU) and add the following line:

```
DAILY
```

Edit prefix.MICS.CNTL(JCLGENU) and submit the job for execution. This job should complete with a return code of zero.

Note: If the input data set names for a component change, the INPUTccc members or INPUTRDR can be modified and the DAILY regenerated using JCLGENU.

5.16 SAS Configuration Options Used by CA MICS

Your sharedprefix.MICS.PARMS library contains SAS configuration members that are used to set necessary and optimal SAS system options for the operation of CA MICS in several scenarios.

The following are the configuration members. A brief description for each member is also given.

ICFBATCH - options for MICF batch execution
ICFDMS - options for MSAS
MABATCH - options for batch PROCS except MICSRPTS
MATSO - options for MICF foreground executions, and MICF Production and User Reporting job streams
MQRBATCH - options for the Q&R Mainframe Server

Since SAS system options vary depending on the version of SAS being used, CA MICS supplies copies of these members that are version specific. The names of these version-specific members are as follows:

ICFBTCHx
ICFDMSx
MABATCHx
MATSOx
MQRBTCHx

where x is 7 for SAS version 8, or 9 for SAS version 9.

During the construction of a new CA MICS complex, the appropriate version-specific members are copied to the configuration members. For example, if you are using SAS version 9, the PARMS member MABATCH9 is copied to member MABATCH.

Comments are provided in these members to indicate which options are required for CA MICS operation and which may be changed. Note that CA MICS has been certified to run only with the distributed options and mandatory settings.

To add or change SAS configuration options, modifications should be made to the version-specific members so that the modifications will not be backed out by future maintenance. Then a run of the job in sharedprefix.MICS.CNTL(MICSLS1) will copy the correct version-specific members to the configuration members.

User modifications should be documented in the same manner as your other local modifications in sharedprefix.MICS.LOCALMOD.CNTL. This will serve as a record of your local modifications to the delivered CA MICS values.

The following steps should be executed when user modifications are done to one of these members:

__ 1. Make your change to the version-specific member that corresponds to the version of SAS being used at your installation.

___ 2. Make the same change to any higher SAS version version-specific members that may be present in sharedprefix.MICS.PARMS.

___ 3. Edit and submit the job contained in sharedprefix.MICS.CNTL(MICSLS1) to activate the appropriate members for the SAS version you are using.

```
+-----------------------------------------------------+
| Reference(s): Section 3.2.5 (Adapt CA MICS Complex)|
+-----------------------------------------------------+
```

5.17 Converting CA MICS Libraries to PDSE Format

This section contains information related to converting CA MICS libraries to PDSE format, and is presented in the following topics:

5.17.1 Overview of PDSEs

Historically, CA MICS has been installed as a collection of primarily partitioned data sets along with several SAS data libraries, and a few physical sequential data sets.

We recognize the benefits of migrating from partitioned data sets to partition data set extended format data sets. Following are the advantages of using PDSE data libraries:

- A PDSE can automatically reuse space within the data set without having to periodically run a utility to reorganize it. It can use noncontiguous space, reclaimed from deleted members.

- The size of a PDSE directory is flexible and expands to fit the members stored in it.

- The system reclaims space automatically whenever a member is deleted or replaces, and returns it to the pool of space available for allocation to other members of the same PDSE. The space is reused without having to do an IEBCOPY compress.

- PDSE members can be shared. This allows modification of separate members of the same PDSE, at the same time.

- Reduced directory search time. The PDSE directory, which is indexed, is searched using that index.

- Allows creation of multiple members simultaneously.

- PDSEs can reside in up to 123 extents.

- When written to DASD, logical records are extracted from the user's blocks and reblocked.

5.17.2 Limitations of using PDSEs with CA MICS

LOAD Modules:

As of this writing, the CA MICS LOAD library (sharedprefix.MICS.LOAD) cannot be converted to a PDSE. The CA MICS installation and PSP refresh processes utilize the STOW macro for storing the load members into a PDS. This macro can no longer be used to store program objects.

SAS 8.2:

SAS 8.2 has several known issues related to PROC PDS and PDSE data sets, as well as non-SMS managed PDSE data sets and simultaneous update. As SAS no longer actively develops new solutions for version 8.2, we recommend that you only convert your CA MICS installation (complex) to PDSE format if you are at SAS 9.1.3 or higher.

5.17.3 Considerations of using PDSEs with CA MICS

Each member of a PDSE library must reside in its own 4K block while it physically resides on DASD. For libraries with many small members, such as the CA MICS data dictionary and documentation libraries, (sharedprefix.MICS.DIC.TEXT and sharedprefix.MICS.DOC.TEXT, respectively), this results in a larger space requirement for these libraries. Our own in-house test results reflected a 50% increase in the space used by the dictionary library, and a 26% space increase in the documentation library.

Most other libraries also increased in size. We recommend when reallocating/copying a CA MICS PDS to a PDSE library, allocating 20% more space. Especially since, while it is possible to expand a PDSE data set up to 123 extents, to achieve the best performance you should avoid fragmenting your data sets whenever possible.

When allocating a PDSE library, directory blocks are not required when using ISPF to allocate the PDSE library. If you are going to use JCL to allocate the new PDSE library, you will need to specify the number of directory blocks or the job will fail with a JCL error. The directory block specification is only used should the PDSE library be converted back to a PDS library.

5.17.4 CA MICS Support of PDSEs

With the exclusion of the LOAD library as discussed earlier, existing CA MICS complex and unit level partitioned data sets (PDSes) can be MANUALLY converted to partition data set-extended data sets (PDSEs).

The CA MICS installation jobs as well as the Optional Product Installation jobs have been modified to allocate the installation (complex) data sets as either partitioned or partitioned-extended data sets by the use of a user specified parameter. The default is still PDS format.

The PSYyymmL job has been modified such that it recognizes a PDSE data set. Currently during the maintenance unload job, the PSP libraries generally get deleted and reallocated for one or both of the following two reasons:

o Insufficient directory blocks on the PDS to support the additional unloaded members

o Insufficient space allocated to the PDS to support the additional unloaded members

If during the maintenance unload job a library is identified as a PDSE, the directory blocks will no longer be interrogated to determine whether or not to delete and reallocate the PSP library. Only insufficient space will result in deletion and allocation of a new PSP data set.

Any data sets already converted to PDSE format will be reallocated in the same format. In addition, the PSP refresh processes have all been enhanced to allow specification of the data set type. Any libraries that get reallocated will use this parameter to determine data set format (PDS or PDSE). The default is PDS format.

Appendix A: Special CA MICS Processing

The CA MICS database should contain the information necessary to answer most of the questions which arise. There are times when the user may wish to access information which is not available in the CA MICS database. Some of the reasons this may be necessary are:

The data is required prior to the CA MICS daily run.

The data required is too old to be in the production system and is not available in audit archive files.

The data required is for a period prior to the CA MICS implementation.

The data required is for a SYSID not maintained in the production database.

A process is available which allows the creation of databases without disruption of normal CA MICS processing. This process is completely independent of the production CA MICS processing. It operates with its own database and is controlled by its own checkpoint data set.

The remainder of this appendix is organized to provide the user with an explanation of the special processing facilities available.

This section contains the following topics:

A.1 CA MICS SPECIAL Database Overview

Special CA MICS processing facilities allow the creation of a CA MICS database called a "SPECIAL" database. A special database is typically used for ad hoc studies and backload processing. Figure A-1 shows the flow of information and control for SPECIAL database processing.

```
+-----------+
| File      |
| Tailoring |    +--------------+
| Parameters|--->| SPECIAL      |
+-----------+    | Database     |
                 | File         |
                 | Tailoring    |
                 | Generation   |
                 +--------------+
                        |
                        V
      +-----------------+
      |
      |
      V
+------------+
| Database   |
| File       |-+
| Definition | |
| Data       | |
+------------+ |
               |
+------------+ |
| Execution  | |  +--------------+         +----------+
| Parameters |-+->|              |         | SPECIAL  |
+------------+    |  SPECIAL     |-------->| Database |
                  |  Database    |         |          |
+------------+    |  Processing  |         +----------+
| Input      |--->|              |
| Data       |    +--------------+
+------------+
```

Figure A-1. SPECIAL Database Processing

The CA MICS SPECIAL database looks much like any other database unit installed within the CA MICS Database Complex. There are several differences that cause the special database to be desirable.

- The files of a special database are not bound by global database timespan definitions. The files may be tailored in a special way that makes it possible to add or drop files to/from any timespan without affecting the production CA MICS database units. This is accomplished by user specifications in a parameter library member called SELFILES. A job called SELFILES inputs this parameter set, and creates executable code to tailor files in the next database run.

- The components processed by a SPECIAL database may be varied from run to run. Also, the data selection or forcing of data into the database may be varied from run to run. This is accomplished by user specifications in a parameter library member called EXECDEF.

RESTRICTIONS:

The SPECIAL database also has some restrictions worthy of note. The fact that SPECIAL databases are individually tailored may cause some files to appear in timespans that the component does not normally support. Also, no archiving is done. Further, the SPECIAL database does not issue the CA MICS 300 abend (the SYSPARM=NODATA option is automatically provided).

The CA MICS SCHEDULE process has no practical use in a SPECIAL database. Backloading and ad hoc studies of old data do not use the schedule process to the advantage of the database unit. SCHEDULE tends to package runs that execute WEEKLY and MONTHLY processes at inappropriate times.

The CA MICS Accounting and Chargeback Product may not be installed in a SPECIAL database. A SPECIAL database is designed to analyze 'data'. To test out an accounting method or actual implementation requires a TEST unit.

Finally, periodic changing of execution components may create a confusing mix of file cycles. The detail timespan, cycle 01, for file 'x' may represent data from yesterday, while the same cycle of the same timespan for file 'y' may contain data from several weeks ago.

A.2 File Tailoring for the SPECIAL Database

One advantage of the CA MICS SPECIAL database is the ability to tailor component files at will. Normal component file tailoring affects the entire database complex. Adding or deleting a file from a timespan will affect the ability of CA MICS to contain data in all database units in which the component appears.

The SPECIAL database gives the CA MICS user the ability to add files to a timespan, drop files from a timespan, or vary the number of cycles within a file and within a timespan.

This is accomplished by user specifications in a parameter library member called SELFILES. A job called SELFILES reads this parameter set, and creates executable code to tailor files in the next database run.

Reconfiguring the SPECIAL database by running the SELFILES job is typically done between database unit generation and input processing. The user may tailor files after unit installation, but tailoring is typically not done before each daily run.

The following topics are covered in this section:

A.2.1 SPECIAL Database File Tailoring Options (SELFILES)

The SPECIAL database is tailored by user specifications in prefix.MICS.PARMS(SELFILES). The purpose of this member is to tell the SPECIAL database tailoring job (also called SELFILES) which files are to be active, and in which timespans, in the online SPECIAL database.

Each line in the SELFILES member resembles the FILE statement in prefix.MICS.PARMS(DBMODEL) for the associated file. For example, the FILE statement for the ADMEXC file looks like:

```
    FILE BAS  ADMEXC 00 08 00 06 00 00 000 000 aud
    ---- ---  ------ -- -- -- -- -- -- ------- ---
      |   |      |    |  |  |  |  |  |    |     |
      | +-+      |    |  |  |  |  |  |    |     |
      | | +------+    |  |  |  |  |  |    |     |
      | | | +---------+  |  |  |  |  |    |     |
      | | | | +----------+  |  |  |  |    |     |
      | | | | | +-----------+  |  |  |    |     |
      | | | | | | +------------+  |  |    |     |
      | | | | | | | +-------------+  |    |     |
      | | | | | | | | +--------------+    |     |
      | | | | | | | | | +-----------------+     |
      | | | | | | | | | | +---------------------+
      | | | | | | | | | | |
      | | | | | | | | | | +-audit archive specification
      | | | | | | | | | |   (See 2.3.4.1.2)
      | | | | | | | | | +---history archive retention limits
      | | | | | | | | |     (not used in SPECIAL database)
      | | | | | | | | +-----cycles retained in TABLES
      | | | | | | | +-------cycles retained in YEARS
      | | | | | | +---------cycles retained in MONTHS
      | | | | | +-----------cycles retained in WEEKS
      | | | | +-------------cycles retained in DAYS
      | | | +---------------cycles retained in DETAIL
      | | +-----------------complete file name
      | +-------------------component ID
      +---------------------FILE statement keyword
```

Normal (complex-wide) activation of a file in a timespan is accomplished by the specification of the timespan indicator in the FILE statement of the component's CA MICS Component Generator (MCG) definition statements. See the CA MICS System Modification Guide section on the CA MICS Component Generator (MCG) for more information on this process.

Normal definition of the number of cycles of a file to be retained in a timespan in which the file is active is accomplished by the specification of the timespan retention indicators in the FILE statement of the database unit's DBMODEL definition statements. See the section in this document on database space modeling for more information on this process.

The file activation and aging limits specified in the MCG statements and in the DBMODEL will be overwritten by using the SELFILES process. Changing the cycle retention limits of a file in a timespan from zero to a non-zero number will activate the file in that timespan. Changing it from non-zero to zero will deactivate the file in the timespan. Changing all retention limits to zero will deactivate the entire file. Changing the retention limit from one non-zero value to another will alter the aging macros for the file.

Components are specified to the SPECIAL database in prefix.MICS.PARMS(JCLDEF). The components generated here are eligible for SPECIAL database processing. However, those components may or may not be active (see the SPECIAL database EXECDEF specification in the next section).

If the component is active, it will be processed. If it is not active, it will not be processed. All components that are eligible will have certain parts of their executable code changed to reflect the new specification, but only for the SPECIAL database being generated.

A.2.2 Generate Tailored File Descriptions

The SPECIAL database is tailored by user specifications in prefix.MICS.PARMS(SELFILES). The specifications are put into effect by running the SELFILES job.

The prefix.MICS.CNTL library for the SPECIAL database contains a member called SELFILES. The job contained in this member generates executable code from the SELFILES member of the PARMS library. This code causes files to be activated or deactivated, and causes the specified number of cycles of each file to be aged.

The SELFILES job occupies a unique spot in CA MICS. The job is not part of the database unit's generation, and thus is not one of the jobs run during initial database unit installation. It is not associated with changes made to any particular component, so it is not part of the ALLPGEN parameter generation process. It does not depend on the file timespan specifications in the generation library, and thus is not connected with the component generation process (jobs called cccCGEN, where ccc is a component ID). And SELFILES is not necessarily run with each detailed run of input data, the DAILY job, of the SPECIAL database unit.

The purpose of the SPECIAL database at your site will determine how often the SELFILES member of the PARMS library must be modified, and how often the SELFILES job will be run.

- Backloading applications may execute the SELFILES job to set up a backload configuration. After the SELFILES job sets up a backload process, the database unit's DAILY, WEEKLY, MONTHLY, and YEARLY processes are run as many times as necessary to produce the backloaded data desired. The backloaded data is then moved to the production database. The SPECIAL database unit's database may be scratched and reallocated, and the entire SELFILES and backload process may be repeated as many times as necessary to accomplish the backloading goals.
- Special studies of unarchived data typically involve running a SELFILES job to set up the desired component and files. Usually, one or two files per component suffice. A few daily runs may be made to satisfy the study requirements, and reports may be produced from the data. The SPECIAL database unit's database may be scratched and reallocated, and at some later time the entire SELFILES and load process may be repeated to accomplish new ad hoc study goals.

A.3 SPECIAL Database Execution Definitions

The components processed by a SPECIAL database may be varied from run to run. Also, the data selection or forcing of data into the database may be varied from run to run. This is accomplished by user specifications in a parameter library member called EXECDEF.

Parameters normally specified in EXECDEF are those which can easily be varied from run to run, such as the selection of management objective reports to be produced. Four statements that are not allowed in the EXECDEF member of a regular database unit are supported in a SPECIAL database. These are:

COMPONENTS - Specify the components that are to be treated by the current input processing run (DAILY job).

SELECT - Performs input data selection according to a window of time stamps.

FORCE - Forces data into the database (regardless of normal date/time duplicate data elimination criteria) according to a specified time stamp window.

ORGSYSID - Selects data from a specified system or systems, and automatically discards data from systems not listed. Note that the SYSID(s) specified here are the original (SMF) SYSIDs.

A sample prefix.MICS.PARMS(EXECDEF) is shown next. The statements in this parameter member that relate to special database processing are indicated by "--->" in the margin.

```
*
* EXECUTION TIME DEFINITIONS
*
CREATE    EXCEPTIONFILES ALL
REPORT    EXCEPTIONS     NONE
REPORT    MBODAILY       ALL
REPORT    MBOWEEKLY      ALL
REPORT    MBOMONTHLY     ALL
SCHEDULE  WEEKEND        NO
INPUTLOGLIMIT 99
TITLE     SPECIAL RUN
TITLE     FOR OPERATIONS DEPARTMENT
*
* SPECIAL PROCESSING PARAMETERS
* (ALLOWED ONLY FOR SPECIAL PROCESSING)
```

```
    *
--->COMPONENTS CIC IMS
--->SELECT     05AUG82:05:00:00.00   05AUG82:05:59:59.99
--->FORCE      05AUG82:05:00:00.00   05AUG82:05:59:59.99
--->ORGSYSIDS  1500
```

If necessary, statements in EXECDEF that apply to all components are automatically adjusted to the special database. For example, if the user has

```
    REPORT    MBOMONTHLY     SMF RMF TSO
    ...
    COMPONENTS CIC IMS
```

coded for component selection, the executed components conflict with the intended monthly management objective report list. In this case, no monthly management objective reports would be produced.

Also, it may be appropriate to change the TITLE statement periodically, certainly more often than for a regular database. Different backloading operations or ad hoc studies would have the TITLE in each operational job's output. An accurate and descriptive TITLE aids in identifying the configuration under which the run was made. Note that it is not always possible to tell the configuration under which the run was made simply by the prefix of the database unit.

A detailed explanation of each of the EXECDEF statements for SPECIAL databases is contained in the sections that follow.

A.3.1 Execution Components List

The components processed by a SPECIAL database may be varied from run to run. This is accomplished by user specification of the COMPONENTS statement in the SPECIAL database unit's parameter library member EXECDEF.

Only one COMPONENTS statement is allowed in EXECDEF. The components listed on the statement must have been defined as generated into the database unit, by the COMPONENTS statement in prefix.MICS.PARMS(JCLDEF). For example,

```
prefix.MICS.PARMS(JCLDEF):
    ...
    COMPONENTS TSO SMF RMF CIC IMS

prefix.MICS.PARMS(EXECDEF):
    ...
    COMPONENTS CIC IMS
```

is allowed. The execution components are the CICS and IMS components, and they have been used from the JCLDEF member to create the SPECIAL database unit's JCL by the JCLGEN process. However,

```
prefix.MICS.PARMS(JCLDEF):
    ...
    COMPONENTS TSO SMF RMF CIC IMS

prefix.MICS.PARMS(EXECDEF):
    ...
    COMPONENTS NPA
```

is not allowed, and will cause an execution time message with an abnormal termination of the step in which the inconsistency was found. The execution component NPA has not been used from the JCLDEF member to create the SPECIAL database unit's JCL by the JCLGEN process.

The effect of limiting execution components on database unit DAILY update run time and logic is dramatic. If the backload or ad hoc study needs only a few components, only those components need be processed. DAILY update routines, exception processing, file aging, and management objective report modules for components not selected are not executed.

A.3.2 SPECIAL Select/Force Option

The input data processed by a SPECIAL database may be selected from a to-date file of input data. It may also be desirable to bypass normal duplicate data elimination and force specified periods of data into the SPECIAL database.

SPECIAL databases are more open to such selection and forcing of data. However, a facility exists that can be used to define select and force parameters that apply to all component update routines active in the DAILY process. This is accomplished by user specification of the SELECT and FORCE statements in the SPECIAL database unit's parameter library member EXECDEF.

Only one SELECT statement is allowed in EXECDEF. The SELECT statement will cause the DAILY update to perform as if you coded the desired selection window on each component record in the checkpoint data set. The time stamp bounds listed on the statement must follow the rules of formatting such a time stamp:

```
    SELECT     ddmmmyy:hh:mm:ss.th   ddmmmyy:hh:mm:ss.th
               -------------------   -------------------
                  beginning time        ending time
```

where:

```
    ddmmmyy        - day, month, and two-digit year of date
    hh:mm:ss.th    - hour,  minute, and second to hundredths
                     of the time of day
    beginning time - the time-stamp that begins the selection
                     window
    ending time    - the time-stamp that ends the selection
                     window
    punctuation    - (: and .) is coded in the positions
                     shown
```

For example:

```
prefix.MICS.PARMS(EXECDEF):
    ...
    SELECT     05AUG82:05:00:00.00   05AUG82:05:59:59.99
```

which selects data from the fifth hour of August 5, 1982.

Only one FORCE statement is allowed in EXECDEF. The FORCE statement will cause the DAILY update to perform as if you coded the desired force window on each component record in the checkpoint data set. The time stamp bounds listed on the statement must follow the rules of formatting such a time stamp:

```
    FORCE      ddmmmyy:hh:mm:ss.th   ddmmmyy:hh:mm:ss.th
               -------------------   -------------------
                  beginning time         ending time
```

where:

```
    ddmmmyy        - day, month, and two-digit year of date
    hh:mm:ss.th    - hour, minute, and second to hundredths
                     of the time of day
    beginning time - the  time  stamp  that  begins the force
                     window
    ending time    - the time-stamp that ends the force
                     window
    punctuation    - (: and .) is coded in the positions
                     shown
```

For example:

```
prefix.MICS.PARMS(EXECDEF):
    ...
    FORCE      05AUG82:05:00:00.00   05AUG82:05:59:59.99
```

which forces data from the fifth hour of August 5, 1982.

See the discussion of the CA MICS Select and Force Facilities, in section 4.1.5 of this guide.

A.3.3 ORGSYSID Selection

The input data processed by a SPECIAL database may be selected from a composite file of input data that contains data from more systems than the user wishes to process. The method used to treat the low volume of such occurrences for a regular, production database is to code a user input selection exit that causes unwanted records to be dropped.

SPECIAL databases are more susceptible to such selection of data. Therefore, there exists a facility that can be used to define a list of system IDs to process in the DAILY update for the SPECIAL database. This is accomplished by user specification of the ORGSYSID statement in the SPECIAL database unit's parameter library member EXECDEF.

Only one ORGSYSID statement is allowed in EXECDEF. The ORGSYSID statement may contain a list of up to ten original system IDs. The ORGSYSID statement will cause the DAILY update to perform as if you had coded the desired system selection in each component's input selection mechanism. The system identifiers listed on the statement must be the ORIGINAL SMF system IDs (see the CA MICS data element ORGSYSID). A typical implementation example is:

```
    ORGSYSID Q033 R033 X081
```

A.4 Special Database Installation Checklist

Installing a special database unit differs from installing a regular database unit in the following ways:

- JCLGENF is executed to generate the special database installation jobs.

- JCLGENG is executed to generate the special database operational jobs.

- The SELFILES parameter member and the SELFILES job are used in place of the CYCLEGEN job.

- The EXECDEF parameter member is used to specify the special database execution options as well as the usual dynamic execution options.

If you plan to install a special database unit, we recommend that you contact CA Technical Support to discuss your plans and to review the differences between installation of a special database and other installation procedures.

__ 1. Define Catalog Index Prefixes/Tapeprefixes

Define the catalog index for the prefix and the tapeprefix.

No action is required if the high-level node name has already been defined as an alias in the master catalog.

```
+----------------------------------------------------+
| Reference(s): Section 3.5.1                        |
+----------------------------------------------------+
```

__ 2. (OPTIONAL) Worksheet Preparation

Complete the set of Installation Preparation Worksheets for the database unit being defined.

```
+----------------------------------------------------+
| Reference(s): Section 2.3, all subsections         |
|               Chapter 7, ccc Analyzer Guide(s)     |
+----------------------------------------------------+
__  3. Execute COPYLIBS

      Build the database unit libraries.

      __ Edit sharedprefix.MICS.CNTL(COPYLIBS) and fill in
         the parameters on the EXEC statement.  DO NOT enter
         parameter values on the PROC statement.

      __ Submit the following job:

          'sharedprefix.MICS.CNTL(COPYLIBS)'

      +----------------------------------------------------+
      | Reference(s): Section 3.5.2                        |
      +----------------------------------------------------+

__  4. Define JCL Parameters

      Define all of the members in the prefix.MICS.PARMS
      library that are required by JCLGEN:

      prefix.MICS.PARMS(JCLDEF)
      prefix.MICS.PARMS(USERJCL)
      prefix.MICS.PARMS(INPUTRDR or INPUTccc) if an SMF
        source component will be installed in this database
        unit
      prefix.MICS.PARMS(JCLNAMES) if the default PROC or
        data set names should not be used for this unit

      +----------------------------------------------------+
      | Reference(s): Section 2.3.3.2.1 (JCLDEF)           |
      |               Section 2.3.3.2.2 (USERJCL)          |
      |               Section 2.3.3.2.3 (INPUTRDR)         |
      |               Section 2.3.3.3.2.3 (JCLNAMES)       |
      |               Chapter 7, ccc Analyzer Guide(s)     |
      +----------------------------------------------------+

__  5. Verify JCL and CLIST Attributes

      Verify the JCL and CLIST attributes for each CA MICS
      product by reviewing
      sharedprefix.MICS.PARMS(JCL$Iccc), where ccc is the
      three character component identifier.  If changes are
      required, make the changes in
      prefix.MICS.PARMS(JCLINFO).

      +----------------------------------------------------+
      | Reference(s): Section 2.3.3.3.2.2 (JCLINFO)        |
      +----------------------------------------------------+
__  6. Execute JCLGENA

      Generate the CA MICS Unit Installation Jobs.

      __ Edit prefix.MICS.CNTL(JCLGENA) and fill in the
         parameters on the EXEC statement and the data set
         name (DSN=) parameter on the //ULOCKIT DD
         statement.

      __ Submit the following job:

           'prefix.MICS.CNTL(JCLGENA)'

      +----------------------------------------------------+
      | Reference(s): Section 3.5.3.2                      |
      +----------------------------------------------------+

__  7. Examine CNTL Members

      Check the JCL generated by JCLGENA by examining
      the following members in prefix.MICS.CNTL:

        PROCJOB2   MICSNDB   MICSDU

      If the JCL and JES parameters do not meet your
      installation's standards, update the
      prefix.MICS.PARMS members JCLDEF or USERJCL and
      resubmit JCLGENA.

      +----------------------------------------------------+
      | Reference(s): Section 3.3.3                        |
      +----------------------------------------------------+

__  8. Execute PROCJOB2

      Move the first group of CA MICS cataloged JCL
      procedures to the production PROC library by
      submitting the following job:

        'prefix.MICS.CNTL(PROCJOB2)'

      +----------------------------------------------------+
      | Reference(s): Section 3.5.4                        |
      +----------------------------------------------------+
__  9. Execute COPYLIBU

      Move the CA MICS parameter library members that are
      dependent on the products installed in this database
      unit.  Submit the following job:

        'prefix.MICS.CNTL(COPYLIBU)'

      +---------------------------------------------------+
      | Reference(s): Section 3.5.2.2                     |
      +---------------------------------------------------+

__ 10. Execute JCLGENF

      Generate the special database installation jobs.  For
      background execution, submit the following job:

        'prefix.MICS.CNTL(JCLGENF)'

__ 11. Define PARMS Members

      Define the required PARMS members.

      __ CA MICS base platform (BAS):

          'prefix.MICS.PARMS(SYSID)'
          'prefix.MICS.PARMS(ZONE)'
          'prefix.MICS.PARMS(SITE)'

      Update the members in prefix.MICS.PARMS for the
      appropriate products.

      __ CA MICS Analyzer Option for MeasureWare (AUM):

          'prefix.MICS.PARMS(INPUTAUM)'
          'prefix.MICS.PARMS(AUMAURT)'
          'prefix.MICS.PARMS(AUMOPS)'

      __ CA MICS Analyzer Option for VAX/VMS (DEX):

          'prefix.MICS.PARMS(INPUTDEX)'
          'prefix.MICS.PARMS(DEXOPS)'

      __ CA MICS Analyzer Option for IMS (IMS):

          'prefix.MICS.PARMS(INPUTIMS)'
          'prefix.MICS.PARMS(IMSAURT)'
          'prefix.MICS.PARMS(IMSOPS)'
          'prefix.MICS.PARMS(IMSRLRT)'
```

```
    'prefix.MICS.PARMS(IMSSPND)'

__ CA MICS Analyzer Option for VSE/Power (PWR):

    'prefix.MICS.PARMS(INPUTPWR)'
    'prefix.MICS.PARMS(PWRCONFG)'
    'prefix.MICS.PARMS(PWRGROUP)'
    'prefix.MICS.PARMS(PWRGPRTE)'
    'prefix.MICS.PARMS(PWROPS)'
    'prefix.MICS.PARMS(PWRTNRTE)'

__ CA MICS System Reliability Analyzer Option (SRL):

    'prefix.MICS.PARMS(INPUTSRL)'
    'prefix.MICS.PARMS(SRLDEVS)'
    'prefix.MICS.PARMS(SRLOPS)'

__ CA MICS Tandem Option (TDM):

    'prefix.MICS.PARMS(INPUTTDM)'
    'prefix.MICS.PARMS(TDMOPS)'

__ CA MICS Tape Analyzer Option, TLM Component:

    'prefix.MICS.PARMS(INPUTTLM)'
    'prefix.MICS.PARMS(TLMOPS)'
    'prefix.MICS.PARMS(VOLGPRTE)'

__ Field Developed Applications (FDAs) for MSI:

    'prefix.MICS.PARMS(cccOPS)'

__ Field Developed Applications (FDAs) for MTI:

    'prefix.MICS.PARMS(cccOPS)'
    'prefix.MICS.PARMS(INPUTccc)'
    'prefix.MICS.PARMS(cccSYSID)'

__ CA MICS Analyzer Option for VM/CMS (VMC):

    'prefix.MICS.PARMS(INPUTVMC)'
    'prefix.MICS.PARMS(VMCOPS)'

****************************************************
* --------------INPUT SOURCE NOTE------------------*
*                                                  *
* The following components may use SMF data as an  *
* input source:                                    *
*        AST, CIC, DB2, HSM, IDM, MQS, RMF,        *
*        SMF, SNT, VTS, TSO, VCA, WEB, and         *
*        FDAs (if applicable)                      *
*                                                  *
* For the components listed above, you must choose *
* one of the alternative methods shown below.      *
* These specifications identify the data set names *
* containing the raw data that will be processed   *
* by the unit's DAILY update job.                  *
*                                                  *
*   (1) If DAYSMF ON is specified in               *
*       prefix.MICS.PARMS(JCLDEF), then you must   *
*       identify the input data set names in       *
*       prefix.MICS.PARMS(INPUTRDR).               *
*                                                  *
*   (2) If DAYSMF OFF is specified in              *
*       prefix.MICS.PARMS(JCLDEF) you must         *
*       identify the input data set names in       *
*       prefix.MICS.PARMS(INPUTccc) for each       *
*       component in the unit.                     *
*                                                  *
*   (3) If this component is in a unit by itself,  *
*       then you must identify the input data set  *
*       names in prefix.MICS.PARMS(INPUTRDR).      *
*                                                  *
****************************************************

__ CA MICS CA ASTEX Option (AST):

    'prefix.MICS.PARMS(INPUTAST)'
    'prefix.MICS.PARMS(ASTAURT)'
    'prefix.MICS.PARMS(ASTOPS)'

__ CA MICS Analyzer Option for CICS (CIC):

    'prefix.MICS.PARMS(INPUTCIC)'
    'prefix.MICS.PARMS(CICALONE)'
    'prefix.MICS.PARMS(CICAURT)'
    'prefix.MICS.PARMS(CICMSAC)'
    'prefix.MICS.PARMS(CICOPS)'
    'prefix.MICS.PARMS(CICRLRT)'
    'prefix.MICS.PARMS(CICTHRSH)'

__ CA MICS Analyzer Option for DB2 (DB2):

    'prefix.MICS.PARMS(INPUTDB2)'
    'prefix.MICS.PARMS(DB2OPS)'

__ CA MICS Analyzer Option for CA IDMS (IDM):
    'prefix.MICS.PARMS(INPUTIDM)'
    'prefix.MICS.PARMS(IDMAURT)'
    'prefix.MICS.PARMS(IDMOPS)'
    'prefix.MICS.PARMS(IDMRLRT)'
    'prefix.MICS.PARMS(IDMRPTS)'

__ CA MICS Analyzer Option for MQSeries (MQS):

    'prefix.MICS.PARMS(INPUTMQS)'
    'prefix.MICS.PARMS(MQSOPS)'

__ CA MICS Hardware and SCP Analyzer (RMF):

    'prefix.MICS.PARMS(INPUTRMF)'
    'prefix.MICS.PARMS(RMFOPS)'

__ CA MICS Batch and Operations Analyzer (SMF):

    'prefix.MICS.PARMS(INPUTSMF)'
    'prefix.MICS.PARMS(JOBGROUP)'
    'prefix.MICS.PARMS(JOBGPRTE)'
    'prefix.MICS.PARMS(PRINTDEF)'
    'prefix.MICS.PARMS(SMFOPS)'
    'prefix.MICS.PARMS(SMFUOPTS)'
    'prefix.MICS.PARMS(TURNRTE)'

__ CA MICS Tape Analyzer Option, VTS Component:

    'prefix.MICS.PARMS(INPUTVTS)'
    'prefix.MICS.PARMS(VTSOPS)'
    'prefix.MICS.PARMS(VTSCMPID)'

__ CA MICS Network Analyzer Option (SNT):

    'prefix.MICS.PARMS(INPUTSNT)'
    'prefix.MICS.PARMS(NETGROUP)'
    'prefix.MICS.PARMS(NETGPRT)'
    'prefix.MICS.PARMS(NETAPPL)'
    'prefix.MICS.PARMS(SNTCIDRT)'
    'prefix.MICS.PARMS(SNTNIDRT)'
    'prefix.MICS.PARMS(SNTOPS)'
    'prefix.MICS.PARMS(SNTPLURT)'
    'prefix.MICS.PARMS(SNTRPTD)'
    'prefix.MICS.PARMS(SNTRPTM)'

__ CA MICS Analyzer for TSO:

    'prefix.MICS.PARMS(INPUTTSO)'
    'prefix.MICS.PARMS(TSOAURT)'
    'prefix.MICS.PARMS(TSOOPS)'

__ CA MICS Space Analyzer Option (VCA):

    'prefix.MICS.PARMS(INPUTHSM)'
    'prefix.MICS.PARMS(INPUTVCA)'
    'prefix.MICS.PARMS(HSMOPS)'
    'prefix.MICS.PARMS(VCAOPS)'

__ CA MICS Web Analyzer Option (WEB):

    'prefix.MICS.PARMS(INPUTWEB)'
    'prefix.MICS.PARMS(WEBOPS)'
    'prefix.MICS.PARMS(WEBSLIST)'
    'prefix.MICS.PARMS(WLECONF)'
    'prefix.MICS.PARMS(WLESCRPT)'

 __ Field Developed Applications (FDAs) using SMF data
    as input source:

     'prefix.MICS.PARMS(INPUTccc)'

 ***IMPORTANT****IMPORTANT***IMPORTANT****IMPORTANT***
 *                                                   *
 *  FDAs may or may not use SMF data as an input     *
 *  source.                                          *
 *                                                   *
 *  CA MICS does NOT provide an automated interface  *
 *  to CA SMF Director from within an FDA.  This     *
 *  means that CA MICS will NOT be able to read the  *
 *  CA SMF Director duplicate split index and obtain *
 *  the necessary input files.  However, any SMF data*
 *  sets output from CA SMF Director may still be    *
 *  directly input as standard SMF data.             *
 *                                                   *
 ***IMPORTANT****IMPORTANT***IMPORTANT****IMPORTANT***

 +---------------------------------------------------+
 |  References:  Section 2.3.2, all subsections      |
 |               Section 2.3.3.2, all subsections    |
 |               Chapter 7, ccc Analyzer Guide(s)    |
 |               Section 7.6, System Modification    |
 |                 Guide (for MSI)                   |
 |               Section 8.7, System Modification    |
 |                 Guide (for MTI)                   |
 |               Section 1.5.2.3, CA ASTEX Option    |
 |                 Guide                             |
 |               Section 7.2.2, Application          |
 |                 Extension Option for Tandem Guide |
       +---------------------------------------------------+

__ 12. (OPTIONAL) Update COMPRESS

      Update prefix.MICS.PARMS(COMPRESS).

      The COPYLIBU job generated the COMPRESS parameter
      library member.  Review and modify the COMPRESS
      contents.

      +---------------------------------------------------+
      | Reference(s): Section 2.3.2.6                     |
      +---------------------------------------------------+

__ 13. Execute SELFILES

      Prepare the input for SELFILES.

      __ Edit prefix.MICS.PARMS(SELFILES) and modify as
         necessary for file tailoring.

         If the SELFILES member is empty or does not exist,
         copy the contents of prefix.MICS.PARMS(DBMODEL) as
         a starting point.

      __ Tailor the FILE statements of the SELFILES member
         according to both the instructions in section A.2
         and your requirements.

         Note that the contents of DATABASE, SYSID, and DATA
         statements and comments in the SELFILES member will
         be ignored in subsequent processing.

      __ After the file tailoring, edit and submit the
         SELFILES job:

           'prefix.MICS.CNTL(SELFILES)'

      Be sure that the characteristics for all products that
      you are installing have been defined in the SELFILES
      parameters member.

      +---------------------------------------------------+
      | Reference(s): Section 2.3.4.1.2                   |
      |               Section A.2                         |
      |               Chapter 6, System Modification      |
      |                 Guide (User Components)           |
      +---------------------------------------------------+
```

__ 14. Review UNITGEN

Review sharedprefix.MICS.PARMS(UNITGEN) for the options in effect.

```
+-----------------------------------------------------+
|  Reference(s):  Section 5.6.1                       |
+-----------------------------------------------------+
```

__ 15. (CONDITIONAL) Execute UNITGEN

Submit the following job:

'prefix.MICS.CNTL(UNITGEN)'

Ensure that there are no error messages in MICSLOG or SYSTSPRT and that the MICSLOG contains the normal termination messages, BAS10999I or BAS11999I.

If the UNITGEN NOAUTOSUBMIT, or UNITGEN AUTOSUBMIT and NOOPJCL parameters have been specified, UNITGEN will complete with a normal return code of 4. Otherwise, UNITGEN and UNITGEN1 should complete with a return code of 0.

__ 16. (CONDITIONAL) Execute UNITGEN1

Perform this step only if your installation has specified

UNITGEN NOAUTOSUBMIT

You must manually submit the UNITGEN1 job to perform the required JCL and parameter generations.

After the UNITGEN job from the previous step completes successfully, submit the following job:

'prefix.MICS.CNTL(UNITGEN1)'

__ 17. Verify UNITGEN1 Results

Verify that the UNITGEN1 job executed successfully.

The UNITGEN1 job contains multiple steps that perform COPYLIBU, CYCLEGEN, JCLGEN, cccPGEN, and GDGSGEN processing or part of it.

Ensure that there are no error messages in MICSLOG or

```
      SYSTSPRT and that the MICSLOG contains the normal
      termination messages, BAS10999I or BAS11999I.

      All steps should complete with a return code of zero.

      If errors are encountered during UNITGEN1 job
      execution, correct the errors and rerun UNITGEN and
      UNITGEN1 jobs respectively.

      +---------------------------------------------------+
      |  Reference(s):  Sections 5.6 and 5.6.1           |
      +---------------------------------------------------+

__ 18. Execute PROCJOB3

      Move CA MICS database set-up PROCS to the installation
      PROCLIB by submitting the following job:

        'prefix.MICS.CNTL(PROCJOB3)'

      +----------------------------------------------------+
      | Reference(s): Section 3.5.4                        |
      +----------------------------------------------------+

__ 19. (CONDITIONAL) Execute JCLGEND

      Perform this step only if your installation has
      specified

      UNITGEN NOOPJCL

      If NOOPJCL is specified, UNITGEN1 did not generate the
      CA MICS operational jobs.

      To generate these jobs, submit the following job:

        'prefix.MICS.CNTL(JCLGEND)'

      Ensure that there are no error messages in MICSLOG or
      SYSTSPRT and that the MICSLOG contains the normal
      termination messages, BAS10999I or BAS11999I.

      All steps should complete with a return code of zero.

      +----------------------------------------------------+
      |  Reference(s):  Section 5.6.1                      |
      +----------------------------------------------------+

__ 20. Execute ALLOCATE
```

```
      Allocate the Online Database and associated files.

      __ Edit prefix.MICS.CNTL(ALLOCATE) and fill in the
         SPACE, VOLSER, and UNIT keyword values.

      __ Submit the following job:

          'prefix.MICS.CNTL(ALLOCATE)'

      +---------------------------------------------------+
      | Reference(s): Section 3.5.6.2                     |
      +---------------------------------------------------+

__ 21. Execute GDGSGEN

       Note:  Archiving is not used on a special database
              unit.

      Update/create GDG entries for installed products in
      the OS CVOL or VSAM catalog.  Submit the following
      job:

        'prefix.MICS.CNTL(GDGSGEN)'

      +---------------------------------------------------+
      | Reference(s): Section 3.5.6.3                     |
      +---------------------------------------------------+

__ 22. Execute JCLGENG

      Generate the operational JCL procedures.  Submit the
      following job:

        'prefix.MICS.CNTL(JCLGENG)'

      +---------------------------------------------------+
      | Reference(s): Section 3.5.3.3                     |
      +---------------------------------------------------+

__ 23. Review and Update EXECDEF

      Review the default values for the dynamic execution
      options in prefix.MICS.PARMS(EXECDEF) and update as
      needed.  Carefully review the entries affecting data
      retention limits.

      +---------------------------------------------------+
      | References: Section 2.3.5                         |
      |             Section A.3                         |
      |             Section 7.3.6 Batch and Operations  |
      |              Analyzer Guide                     |
      +-------------------------------------------------+

__ 24. Execute CKPTINIT

      Before performing the first database update,
      initialize the checkpoint file.  Submit the following
      job:

        'prefix.MICS.CNTL(CKPTINIT)'

      +-------------------------------------------------+
      | Reference(s): Section 3.5.7                     |
      +-------------------------------------------------+

__ 25. Final Review

       The system is fully generated and ready for checkout.
       Make a final review of all definitions and JCL,
       applying any final user modifications.

       +-------------------------------------------------+
       | Reference(s): Section 3.5.7                     |
       +-------------------------------------------------+

__ 26. Execute DAILY Process

      Run the first daily process.  Submit the following
      job:

        'prefix.MICS.CNTL(DAILY)'

      or

      Use the Operational Status and Tracking DAILY command
      in MWF to submit daily processing.

      +-------------------------------------------------+
      | Reference(s): Section 3.5.7 and Chapter 4       |
      +-------------------------------------------------+

__ 27. (OPTIONAL) Execute cccCHECK

      Run a database check job for each product in the unit
      to verify that key database definitions have been
      correctly defined and implemented.  Submit the
      following job:
        'prefix.MICS.CNTL(cccCHECK)'

      +----------------------------------------------------+
      | Reference(s): Section 3.5.7                        |
      +----------------------------------------------------+

__ 28. (OPTIONAL) Execute BACKUP and RESTORE

      Run the backup and restore jobs.  Submit the following
      jobs:

        'prefix.MICS.CNTL(BACKUP)'
        'prefix.MICS.CNTL(RESTORE)'

      or

      Use the Operational Status and Tracking BACKUP and
      RESTORE commands in MWF to submit the processing.

      +----------------------------------------------------+
      | Reference(s): Section 3.5.7 and Chapter 4          |
      +----------------------------------------------------+
```