

Unicenter[®] CA-SpaceMan[®] File Management for Adabas

User Guide

4.1



Computer Associates[®]

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been 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 the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

© 2003 Computer Associates International, Inc.

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.



Contents

Chapter 1: General Information

Purpose	1-1
Installation Materials	1-2
Unicenter CA-SpaceMan System Requirements	1-2
DASD Space Requirements	1-3
Distribution Tape Format	1-3
Naming Conventions	1-3
CA Common Services for z/OS and OS/390	1-4
Installing CA Common Services for z/OS and OS/390	1-4
CAIRIM	1-5
CA LMP	1-5
Requirements	1-6
Using CA LMP	1-6
Technical Support	1-8

Chapter 2: Unicenter CA-SpaceMan Installation

Installing Unicenter CA-SpaceMan	2-1
Required Unicenter CA-APAS Install Steps	2-1
Unicenter CA-SpaceMan Basic Reporting Facility	2-2
Unicenter CA-SpaceMan Repository and Natural Interface	2-2
Unicenter CA-SpaceMan IOLOG Processing	2-2
Unicenter CA-SpaceMan Configurator Facility	2-2
Required Unicenter CA-APAS Install Steps	2-3
Step 1. Basic Installation	2-3
Step 2. Unicenter CA-APAS Natural Interface and Data Collector	2-3

Unicenter CA-SpaceMan Basic Reporting Facility	2-4
Step 1. Define the Unicenter CA-SpaceMan Computer Associates LMP key	2-4
Step 2. Complete the Unicenter CA-SpaceMan Installation Worksheet	2-5
Unicenter CA-SpaceMan Installation Worksheet	2-5
Step 3. Create and Submit Initial Distribution Tape Extract Job	2-9
Step 4. Read SPREADME	2-10
Step 5. Modify the Unicenter CA-SpaceMan Installation Edit Macro	2-10
Step 6. Allocate and Load Unicenter CA-SpaceMan Product Files	2-11
Step 7. Verify the Basic Unicenter CA-SpaceMan Reporting Facility	2-12
Unicenter CA-SpaceMan Repository and Natural Interface	2-13
Step 1. Load the Unicenter CA-SpaceMan Repository File	2-13
Load a New Repository File	2-13
Modify an Existing Repository File	2-14
Step 2. Load the Unicenter CA-SpaceMan Natural Components	2-15
Step 3. Before Getting Started Online	2-18
Modify XSPMUPD	2-18
Modify XDIRUPD	2-18
Modify XHISUPD	2-19
Step 4. Verify Unicenter CA-SpaceMan Natural Interface and Repository	2-20
Verify Unicenter CA-SpaceMan Natural Interface	2-20
Verify the Repository	2-21
Unicenter CA-SpaceMan IOLOG Processing	2-22
Unicenter CA-SpaceMan IOLOG Records	2-22
Step 1. Install the Trace Exit Module DBGIOR5	2-23
ADARUN Parameters	2-23
Unicenter CA-SpaceMan Control Cards	2-24
Unicenter CA-SpaceMan DD Statements	2-24
Step 2. Implement IOLOG COPY File for Batch Processing	2-25
Prepare the MPM	2-25
Modify XSPMNENS	2-25
Step 3. Execute MPM with Unicenter CA-SpaceMan IOLOG	2-26
Starting the MPM and Unicenter CA-SpaceMan	2-26
Step 4. Verify Unicenter CA-SpaceMan IOLOG Processing	2-27
Unicenter CA-SpaceMan Configurator Facility	2-28
Configurator Options	2-28
Step 1. Install Unicenter CA-SpaceMan Configurator Subroutines	2-29
Step 2. Prepare Unicenter CA-SpaceMan Utility Card Generator JCL	2-30
Modify XGCRD1 JCL	2-30
Modify XGCRD2 JCL	2-30
Modify XGCRD3 JCL	2-30
Modify Natural JCL Members	2-30

Step 3. Prepare Unicenter CA-SpaceMan OS Space Allocator JCL	2-31
Modify XSPMALC	2-31

Chapter 3: Unicenter CA-SpaceMan Functionality

Features	3-1
Unicenter CA-SpaceMan Components	3-2
Physical I/O Monitoring and Tuning	3-4
Unicenter CA-SpaceMan I/O Logging	3-4
Batch Reporting of Physical I/O Activity	3-4
Real-time Reporting of Physical I/O Activity	3-5
Reporting of Physical I/O Activity	3-6
Physical I/O Reports	3-7
SUMMARIZE Requests	3-7
EXTRACT Requests	3-9
Unicenter CA-SpaceMan Database Maintenance Options	3-10
Configurator Options	3-10
Completely Reorganizing the Database	3-11
Reallocating the Database	3-12
Maintaining Individual Files	3-13
Allocating New Files	3-13
Reorganizing Existing Files	3-13

Chapter 4: Understanding Adabas Space Management

Unicenter CA-SpaceMan Analysis	4-2
Padding Factors	4-2
Descriptors	4-2
ADAM Files	4-2
Physical I/O Breakdown	4-2
Work Data Sets	4-2
Areas of Application	4-3
Managing Adabas Logical Files	4-3
Managing DASD	4-3
File Design and Application Tuning	4-3
Real-Time Performance Monitoring	4-4
Capacity Planning	4-4
Chargeback and Accounting	4-4
Debugging	4-4

Space Management	4-5
General Space Management	4-5
Adabas Space Management	4-6
Data Set Positioning	4-6
Logical File Positioning	4-7
Blocksizes	4-8
Data Storage	4-8
Associator	4-8
Work	4-8
Cache	4-9
Compression	4-9
Padding Areas	4-10
Associator Padding	4-11

Chapter 5: SPMREP Database Analysis Reports

Generating Unicenter CA-SpaceMan Database Analysis Reports	5-1
Modifying Report Program Control Cards	5-1
GLOBALS Statement	5-2
Syntax	5-2
Example	5-2
GLOBALS Parameters and Options	5-3
REPORT Statement	5-5
Syntax	5-5
Example	5-5
REPORT Parameters and Options	5-6
Address Converter Reports	5-6
Associator Reports	5-7
Data Storage Reports	5-8

Chapter 6: Using and Understanding SPMREP Reports

Typical Adabas Space Management Problems	6-1
Data Storage Analysis	6-1
Index Analysis	6-10
Address Converter Analysis	6-16
Work Analysis	6-18
Disk Device Analysis	6-23

Chapter 7: Before Getting Started Online

The Unicenter CA-SpaceMan Repository	7-1
Purposes of the Repository	7-2
Types of Data in the Repository	7-2
Record Types	7-3
Repository File Sources	7-3
The Space Management Process	7-4
Adding Current Database Information to the Repository File	7-5
Updating Current Database Information	7-5
Adding I/O Samples to the Repository	7-6
I/O Sample Timing	7-6
I/O Sampling Procedure	7-7
Online Maintenance	7-8
Creating Projections, Configurations, and Utility Control Cards	7-8

Chapter 8: Getting Started Using Online Unicenter CA-SpaceMan

Online Menu System	8-1
Sample Online Session	8-2
Getting Started Online	8-2
The Unicenter CA-SpaceMan Main Menu	8-3
The Current Databases Menu	8-4
The Select a Current Database Panel	8-4
Displaying Current Database Information	8-6
Displaying I/O Sample Information	8-9
Creating a Projection to Reload/Add Files	8-13
Creating a Configuration From a Projection	8-19
Creating a Projection to Reallocate the Database	8-26
Creating a Projection to Reorganize the Database	8-35

Chapter 9: Unicenter CA-SpaceMan Configurator

Configurator Overview	9-1
Option 1: Completely Reconfiguring the Database	9-2
Input to the Configurator	9-2
Gathering Data	9-2
Sorting Data	9-3
Configurator Processing	9-3
Configurator Processing Algorithm	9-3
Output from the Configurator	9-4
Option 2: Reallocating the Database	9-5
Input to the Configurator	9-5
Gathering Data	9-6
Sorting Data	9-6
Configurator Processing	9-6
Configurator Processing Algorithm	9-6
Output from the Configurator	9-7
Option 3: Adding/Reloading Individual Files	9-8
Input to the Configurator	9-8
Gathering Data	9-8
Sorting Data	9-9
Configurator Processing	9-9
Configurator Processing Algorithm	9-9
Output from the Configurator	9-10
Understanding the Unicenter CA-SpaceMan OS Allocation Utility	9-11

Chapter 10: Reporting and Forecasting DASD Space Usage

Unicenter CA-SpaceMan Space Usage History	10-2
User Processing of HISTRECS Space Usage Data	10-2
HISTRECS File	10-2
Sample User Processing	10-3
Posting Space Usage Data to the Repository	10-3
Retention Control Records	10-4
Adding Retention Control Records	10-4
Using Accounting Control Records	10-5
Creating Accounting Control Records	10-5
Applying Accounting Control	10-5
Accounting Control Effective Period	10-5

Reporting Space Usage History	10-5
Selecting Reports	10-6
Batch Reporting	10-6
Forecasting Future Space Usage	10-6
Base Record Contents	10-6
Space Forecasting Process	10-7
Selecting Reports	10-7
Batch Reporting	10-7
Reporting Space Usage for Natural FUSER System Files	10-8
Selecting Reports	10-8

Chapter 11: IOLOG Record Processing

Generating IOLOG Records	11-1
IOLOG Record Layout	11-2
Data Collector Control Card Syntax	11-3
GLOBALS: Definition of Global Environment	11-3
Syntax	11-3
Example	11-3
Parameters	11-4
DECLARE: The Declare A User-Defined Field	11-5
Syntax	11-5
Example	11-5
Parameters	11-6
EXTRACT: Report/Output Individual Command Data	11-6
Syntax	11-6
Example	11-7
Parameters	11-7
SUMMARIZE: Report/Output Interval and Total Summary Data	11-10
Syntax	11-10
Example	11-11
Parameters	11-11
COPY: Copy Command Log Records for Later Batch Processing	11-14
Syntax	11-14
Example	11-14
Parameters	11-14

Chapter 12: IOLOG Data Fields

CMDLOG Fields	12-1
IOLOG Data Fields	12-2

Chapter 13: Unicenter CA-SpaceMan Batch Messages

Unicenter CA-SpaceMan Batch Messages	13-1
--	------

Chapter 14: Unicenter CA-SpaceMan Online Messages

Unicenter CA-SpaceMan Online Messages	14-1
---	------

Index

General Information

This guide contains the information and instructions you need to install and use Computer Associates Unicenter CA-SpaceMan File Management for Adabas (Unicenter CA-SpaceMan). Unicenter CA-SpaceMan is a set of programs used with the Adabas database management system to address the problems of physical space management.

Purpose

This chapter provides a brief overview of materials and system requirements needed to install Unicenter CA-SpaceMan.

Certain knowledge is required to complete the installation process and for the execution of Unicenter CA-SpaceMan. For example, you must be familiar with z/OS and OS/390 IEBCOPY and JCL; Adabas ADALOD compress and load utilities; and features and functionality of Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS).

Installation Materials

Before beginning the installation procedure, make sure that you have the correct Unicenter CA-SpaceMan installation tape. Verify the volume serial number on the tape matches the volume serial number specified in the Product Maintenance Letter received with the product package. If you did not receive the proper tape, please contact the Computer Associates Order department at 1-800-841-8743.

In addition, verify that you received a Product Authorization letter containing a numeric password required for installation. If you did not receive the Product Authorization letter, please contact your local Technical Support group.

IBM users, make sure you also received a CA LMP Key Certificate for Unicenter CA-SpaceMan with the product package and installation tape. If you did not receive a Unicenter CA-SpaceMan LMP Key Certificate, contact your Computer Associates account manager or the Total License Care Hot Line at 1-800-338-6720.

Unicenter CA-SpaceMan System Requirements

The following hardware and software requirements are necessary for the successful installation of Unicenter CA-SpaceMan:

Adabas versions 7.1, 7.2, or 7.4

- Natural version 3.1
- z/OS or OS/390 operating systems that support Adabas V7.1 or higher
- Fujitsu MSP operating systems that support Adabas V7.1 or higher
- Hitachi VOS3 operating systems that support Adabas V7.1 or higher
- 3270-type display device with a minimum screen size of 24 by 80 characters
- CICS version 4.1
- CICS Transaction Server versions 1.1, 1.2, 1.3, 2.1 or 2.2
- COM-LETE versions 5.1, 6.1, or 6.2
- IMS versions 5.1 or 6.1

Unicenter CA-SpaceMan also requires an APF-authorized library for execution if Adabas is executing from an APF-authorized library.

Note: Unicenter CA-SpaceMan does *not* require any SVCs or hooks.

DASD Space Requirements

The following table lists the Unicenter CA-SpaceMan product libraries and data sets created. DASD space requirements, based on 3390 geometry, are included for each library.

Lowqual	Type	PRODVOL Size	Contents
SPMVvvv.NATDDM	PS	1 Cylinder	Natural DDMs
SPMVvvv.NATOBJ	PS	2 Cylinders	Natural Object Modules
SPMVvvv.NATSRC	PS	5 Cylinders	Natural Source Modules
SPMVvvv.UNLOAD	PS	1 Cylinder	Empty, unloaded Repository File

Distribution Tape Format

The distribution tape contains data sets for Unicenter CA-APAS, for Unicenter CA-SpaceMan and for Unicenter CA-PLEU. In addition to the product libraries, the tape also contains the unloaded Natural applications for Unicenter CA-APAS, the Unicenter CA-APAS Performance History System and Unicenter CA-SpaceMan. The tape also contains unloaded, empty Adabas files for the Unicenter CA-APAS History File and for the Unicenter CA-SpaceMan Repository File.

Naming Conventions

The naming conventions used to allocate and build the Unicenter CA-SpaceMan product libraries and files are as follows:

h1q.m1q.lowqual

The value for **lowqual** is unique and predefined for each product library and file.

CA Common Services for z/OS and OS/390

To help you quickly understand all that CA Common Services for z/OS and OS/390 offers, this section provides a brief description of the common services that can be used by Unicenter CA-SpaceMan.

The CA Common Services for z/OS and OS/390 are a group of system services that protect your investment in software products by helping you manage your data center more efficiently. The CA Common Services for z/OS and OS/390 offer individual benefits to the user.

The CA Common Services for z/OS and OS/390 that are used with, and benefit, Unicenter CA-SpaceMan is CAIRIM and CA LMP, which assist you in getting Unicenter CA-SpaceMan up and running.

Installing CA Common Services for z/OS and OS/390

According to your data center's specific needs, you can choose to use only certain CA Common Services for z/OS and OS/390. Some of the services are dependent on one or more other services to function properly when interfacing with your Computer Associates software. The interservice dependencies for the services used by Unicenter CA-SpaceMan are listed below.

If you are installing:	You must also install:
CAIRIM	No other services are required
CA LMP	CAIRIM

If the necessary services have not already been installed on your system, you must do so now. Refer to the *CA Common Services for z/OS and OS/390 Getting Started* for detailed instructions.

If the CA Common Services for z/OS and OS/390 services are not installed and you attempt to use Unicenter CA-SpaceMan, a S122 abend results.

CAIRIM

CAIRIM, CAI Resource Initialization Manager, is the common driver for a collection of dynamic initialization routines that eliminate the need for user SVCs, SMF exits, subsystems, and other installation requirements commonly encountered when installing systems software. These routines are grouped under the Computer Associates z/OS and OS/390 dynamic service code, S910. Some of the features of CAIRIM include:

- Obtaining SMF data
- Verification of proper software installation
- Installation of z/OS and OS/390 interfaces
- Automatic startup of Computer Associates and other vendor products
- Proper timing and order of initialization
- No other services are required for proper operation.

Note: CAIRIM is mandatory for Unicenter CA-SpaceMan. It must be installed and started within 30 minutes of IPL time. CAIRIM is part of the CA Common Services for z/OS and OS/390.

CA LMP

The Computer Associates License Management Program (CA LMP) provides a standardized and automated approach to the tracking of licensed software. It uses common real-time enforcement software to validate the user's configuration. CA LMP reports on activities related to the license, usage, and financials of Computer Associates software solutions. The routines that accomplish this are integrated into the Computer Associates z/OS and OS/390 dynamic service code, S910 (the CAIRIM service). Some of the features of CA LMP include:

- Common Key Data Set can be shared among many CPUs
- "Check digits" are used to detect errors in transcribing key information
- Execution Keys can be entered without affecting any Computer Associates software solution that is already running
- There are no special maintenance requirements

Requirements

Unicenter CA-SpaceMan requires CA Common Services for z/OS and OS/390 at genlevel 9901 or above.

Refer to eSupport for additional Unicenter Services minimum genlevel requirements for your release of OS/390 or z/OS.

Using CA LMP

Unicenter CA-SpaceMan requires CA LMP (License Management Program), one of the Common Services, to initialize correctly. CA LMP also provides a standardized and automated approach to the tracking of licensed software.

CA LMP is provided as an integral part of CAIRIM (Resource Initialization Manager), another one of the Common Services. If CAIRIM has not already been installed on your system, you must do so now. Once CAIRIM has been installed or maintained at Service Level C1/9901 or higher, CA LMP support is available for all CA LMP – supported CA software solutions. See the *CA Common Services for z/OS and OS/390 Getting Started* guide for detailed instructions on installing CAIRIM.

Examine the CA LMP Key Certificate you received with the installation or maintenance tape for Unicenter CA-SpaceMan. That certificate contains the following information:

Fields	Descriptions
Product Name	The trademarked or registered name of the CA software solution licensed for the designated site and CPUs.
Product Code	A two-character code that corresponds to Unicenter CA-SpaceMan.
Supplement	The reference number of your license for Unicenter CA-SpaceMan, in the format <i>nnnnnnn - nnn</i> . 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 of Unicenter CA-SpaceMan is valid.
Execution Key	An encrypted code required by CA LMP for Unicenter CA-SpaceMan initialization. During installation, it is referred to as the LMP Code.
Expiration Date	The date (<i>ddmmmyy</i> as in 01AUG02) your license for Unicenter CA-SpaceMan expires.

Fields	Descriptions
Technical Contact	The name of the technical contact at your site responsible for the installation and maintenance of Unicenter CA-SpaceMan. 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 at 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.

The CA LMP execution key, provided on the Key Certificate, must be added to the CAIRIM parameters to ensure proper initialization of Unicenter CA-SpaceMan. To define a CA LMP execution key to the CAIRIM parameters, modify member KEYS in the OPTLIB data set.

The parameter structure for member KEYS is as follows:

PROD (*pp*) **DATE** (*ddmmmyy*) **CPU** (*tttt-mmmm/sssss*) **LMPCODE** (*kkkkkkkkkkkkkkkk*)

Where:

pp – Required. The two-character product code that corresponds to Unicenter CA-SpaceMan. For any given CA LMP software solution, this code agrees with the product code already in use by the CAIRIM initialization parameters for earlier gen levels of that software solution.

The two-character product codes for Unicenter CA-SpaceMan are:

HU z/OS and OS/390

ddmmmyy – The CA LMP licensing agreement expiration date.

tttt-mmmm – Required. The CPU type and model (for example: 3090 - 600) on which the 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.

sssss – Required. The serial number of the CPU on which the 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.

The following is an example of a control statement for the CA LMP execution software parameter. Although this example uses the Unicenter CA-SpaceMan two-character product code, the CA LMP execution key value is invalid and is provided as an example only!

```
PROD(HU) DATE(01AUG02) CPU(3090- — -600 /370623) LMPCODE(52H2K06130Z7RZD6)
```

For a full description of the procedure for defining the CA LMP execution key to the CAIRIM parameters, see the *CA Common Services for z/OS and OS/390 Getting Started*.

Technical Support

If questions arise during the installation or operation of Unicenter CA-SpaceMan, or if you have suggestions regarding the use of the product, please call Computer Associates Technical Support:

- U.S. and Canadian customers: (425) 825-2770
- International customers: Contact your nearest Computer Associates office

Unicenter CA-SpaceMan Installation

Unicenter CA-SpaceMan File Management for Adabas (Unicenter CA-SpaceMan) is fully integrated with the Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS). The steps you are required to perform during the installation of Unicenter CA-SpaceMan depend on whether or not you have already installed the corresponding version of Unicenter CA-APAS.

Unicenter CA-SpaceMan has a number of functions and components. If you are a first-time user, we suggest you do NOT install and use all of its features immediately. Instead, you should take the staged approach described in this chapter. This simplifies the installation process and allows you to use and understand features in a logical progression.

Installing Unicenter CA-SpaceMan

The following briefly describes the steps necessary to install the Unicenter CA-SpaceMan Adabas repository file, Natural components and prepare Natural for Unicenter CA-SpaceMan execution. Installation member names, enclosed in parentheses, are referenced where applicable. This list is intended as an overview and does not contain the details necessary to install Unicenter CA-SpaceMan.

Required Unicenter CA-APAS Install Steps

1. Complete the Unicenter CA-APAS basic installation as described in the *Unicenter CA-APAS Installation Guide*.
2. Complete the installation of the Unicenter CA-APAS Natural Interface and Data Collector as described in the *Unicenter CA-APAS Installation Guide*.

Unicenter CA-SpaceMan Basic Reporting Facility

1. Define the Unicenter CA-SpaceMan Computer Associates LMP key.
2. Complete the Unicenter CA-SpaceMan Installation Worksheet
3. Create and Submit Initial Distribution Tape Extract Job
4. Read SPREADME
5. Modify the Unicenter CA-SpaceMan Edit Macro (SPEDIT).
6. Allocate and Load Unicenter CA-SpaceMan Product Libraries (SPUNLOAD).
7. Verify the Basic Unicenter CA-SpaceMan Reporting Facility (XSPMREP).

Unicenter CA-SpaceMan Repository and Natural Interface

1. Load the Existing Unicenter CA-SpaceMan Repository File.
2. Load the Unicenter CA-SpaceMan Natural Components (LDSPMNAT).
3. Before Getting Started Online (XSPMUPD, XDIRUPD, XHISUPD).
4. Verify Unicenter CA-SpaceMan Natural Interface and Repository.

Unicenter CA-SpaceMan IOLOG Processing

1. Install the Trace Exit Module DBGIOR5 (JCLSPMAN).
2. Implement IOLOG COPY File for Batch Processing (JCLSPMIO, XSPMNENS).
3. Execute MPM with Unicenter CA-SpaceMan IOLOG Processing.
4. Verify Unicenter CA-SpaceMan IOLOG Processing.

Unicenter CA-SpaceMan Configurator Facility

1. Install Unicenter CA-SpaceMan Configurator Subroutines (LNKCNFC).
2. Prepare Unicenter CA-SpaceMan Utility Card Generator JCL (XGCRD1, XGCRD2, XGCRD3, V6FLOAD, V6REORFIL, V6UNRLD).
3. Prepare Unicenter CA-SpaceMan OS Space Allocator JCL (XSPMALC).

Required Unicenter CA-APAS Install Steps

This section provides instructions on the components of Unicenter CA-APAS required for successful installation of Unicenter CA-SpaceMan.

Step 1. Basic Installation

Note: This step is required whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

If you have already completed the Unicenter CA-APAS basic installation as described in Part 1: Basic Installation for z/OS and OS/390 in the chapter “Unicenter CA-APAS Installation” in the *Unicenter CA-APAS Installation Guide*, proceed to the following step - Unicenter CA-APAS Natural Interface and Data Collector.

Otherwise, complete at this time the Unicenter CA-APAS basic installation as described in Part 1: Basic Installation for z/OS and OS/390 in the chapter “Unicenter CA-APAS Installation” in the *Unicenter CA-APAS Installation Guide*. After completing those steps, proceed to the next step.

Step 2. Unicenter CA-APAS Natural Interface and Data Collector

Note: This step is required whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

If you have already completed Unicenter CA-APAS Natural Interface and Data Collector as described in Part 2: Install Unicenter CA-APAS Natural Interface and Data Collector in the chapter “Unicenter CA-APAS Installation” in the *Unicenter CA-APAS Installation Guide*, proceed to the next section, Unicenter CA-SpaceMan Basic Reporting Facility.

Otherwise, complete at this time Unicenter CA-APAS Natural Interface and Data Collector as described in Part 2: Install Unicenter CA-APAS Natural Interface and Data Collector in the chapter “Unicenter CA-APAS Installation” in the *Unicenter CA-APAS Installation Guide*. After completing those steps, proceed with the next section, Unicenter CA-SpaceMan Basic Reporting Facility.

Unicenter CA-SpaceMan Basic Reporting Facility

This section provides step-by-step instructions for the basic installation of Unicenter CA-SpaceMan and the reporting facilities. You may wish to modify some of the steps for your particular installation.

Step 1. Define the Unicenter CA-SpaceMan Computer Associates LMP key

Note: This step is required for IBM z/OS and OS/390 users only.

Make sure you received a CA LMP Key Certificate for Unicenter CA-SpaceMan with your product package and installation tape. If you did not receive a Unicenter CA-SpaceMan LMP Key Certificate, contact your Computer Associates account manager or the CA Total License Care Hot Line at 1-800-338-6720.

You must define the CA LMP execution key to the CAIRIM parameters. For a full description of the procedure, see the *CA Common Services for z/OS and OS/390 Getting Started* manual.

Step 2. Complete the Unicenter CA-SpaceMan Installation Worksheet

Note: This step is required whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

The JCL members you use in the installation steps that follow need to be modified to conform to your site's specifications.

In this step, you complete the Unicenter CA-SpaceMan Installation Worksheet below with the defaults to be used to modify the installation PDS members.

Unicenter CA-SpaceMan Installation Worksheet

Enter the values you assign to the environment variables for the following table.

Environment Variable	Value
ADAASSO	Specify a value for the Adabas Associator data set name to be used in distributed JCL.
ADACLOG	Specify a value for the Adabas Command Log data set name to be used in distributed JCL.
ADADATA	Specify a value for the Adabas Data data set name to be used in distributed JCL.
ADADBID	Specify a value for the DATABASE parameter to be used in distributed JCL.
ADADTYP	Specify a value for the DEVICE parameter to be used in distributed JCL.
ADALOAD	Specify a value for the Adabas load library to be used in distributed JCL.
ADALP	Specify a value for the ADARUN LP parameter to be used in distributed JCL.

Environment Variable	Value
ADALWKP2	Specify a value for the ADARUN LWKP2 parameter to be used in distributed JCL.
ADASVC	Specify a value for the Adabas SVC number to be used in distributed JCL.
ADAWORK	Specify a value for the Adabas Work data set name to be used in distributed JCL.
CICSLOAD	Specify a value for the CICS load library to be used in distributed JCL.
CNFLOAD	Specify a value for the CICS STEPLIB load library to be used in distributed JCL for the CICS versions of the Configurator modules.
DISKUNIT	Specify a value for the generic DASD unit name to be used in distributed JCL.
FSYSDBID	Specify a value for the DBID for the Adabas file containing the Natural System file (FNAT or FDIC) to which the DDMs should be loaded.
FSYSFNR	Specify a value for the file number for the Adabas file containing the Natural System file (FNAT or FDIC) to which the DDMs should be loaded.
FSYSPSWD	Specify a value for the password for the Adabas file containing the Natural System file (FNAT or FDIC) to which the DDMs should be loaded.
FUSRDBID	Specify a value for the DBID for the Adabas file containing the Natural User library (FUSER) to which the Natural source and object members should be loaded.

Environment Variable	Value
FUSRFNR	Specify a value for the file number for the Adabas file containing the Natural User library (FUSER) to which the Natural source and object members should be loaded.
FUSRPSWD	Specify a value for the password for the Adabas file containing the Natural User library (FUSER) to which the Natural source and object members should be loaded.
HLQ	Specify a value for the high-level qualifier of all data sets to be used in distributed JCL.
JOB CARD1	Specify a value for jobcard line one to be used in distributed JCL.
JOB CARD2	Specify a value for jobcard line two to be used in distributed JCL.
MLQ	Specify a value for the mid-level qualifier of all data sets to be used in distributed JCL.
NATBATCH	Specify a value for the Natural batch nucleus name to be used in distributed JCL.
NATLOAD	Specify a value for the Natural load library to be used in distributed JCL.
PRODVOL	Specify a value for the DASD volume on which product libraries are allocated.
SPMLIBID	Specify the value for the Natural library-id where Natural source and object members are to be loaded for Unicenter CA-SpaceMan. The default value is SPACEMAN. You can accept the default or specify a new library-id.

Environment Variable	Value
SPMPSWD	Specify the eight-digit number found in the Product Authorization letter included in the product package.
TAPEUNIT	Specify a value for the generic tape unit name to be used in the distributed JCL used to unload product libraries.
TAPEVOL	Specify a value for the VOLSER of the distribution tape to be used in the JCL to unload product libraries.
VVVV	Specify a value for the Unicenter CA-SpaceMan version of all data sets to be used in distributed JCL.

Step 3. Create and Submit Initial Distribution Tape Extract Job

Note: This step is required whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

In this step, you create a JCL job stream to copy the Unicenter CA-SpaceMan INSTALL library from the Unicenter CA-APAS distribution tape.

The VOLSER of the distribution tape is

ASPvvv

where *vvv* is the current version, release and maintenance level. Be sure to check the tape label for the correct VOLSER.

Follow these steps to create the INSTALL library:

1. Create the following JCL, providing values for the entries that appear in italics, for example, *entry*.

```
//SPMINST JOB (acct info), 'COPY INSTALL LIB', CLASS=x, MSGCLASS=y
//SPMSTP1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//SYSUT3 DD UNIT=diskunit, SPACE=(TRK, (1,1))
//SYSUT4 DD UNIT=diskunit, SPACE=(TRK, (1,1))
//*
//TAPE DD DSN=CA.APASvvv.SPACEMAN.INSTALL,
// LABEL=(12,SL),
// UNIT=tapeunit,VOL=SER=ASPvvv,
// DISP=(OLD,KEEP)
//DISK DD DSN=hlq.mlq.APASvvv.SPACEMAN.INSTALL,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120),
// SPACE=(CYL, (1,1,10)),
// UNIT=diskunit,VOL=SER=prodvol,
// DISP=(NEW,CATLG,DELETE)
//SYSIN DD *,DCB=BLKSIZE=80
COPY INDD=TAPE,OUTDD=DISK
/*
//
```

Note: The LRECL and BLKSIZE in the DCB card can be modified to best suit your DASD environment.

2. Submit the job. If you receive a non-zero return code, correct the JCL and resubmit the job.

Step 4. Read SPREADME

Note: This step is required whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

Read the SPREADME member in the Unicenter CA-SpaceMan INSTALL library for additional information you may need for this installation and/or product usage.

Step 5. Modify the Unicenter CA-SpaceMan Installation Edit Macro

Note: This step is required whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

We recommend that you use the edit macro member SPEDIT in the Unicenter CA-SpaceMan INSTALL library to quickly and accurately make changes to the PDS members used to install Unicenter CA-SpaceMan. You modify SPEDIT with the values entered on the Unicenter CA-SpaceMan Installation Worksheet.

Follow these steps to modify SPEDIT:

1. Replace the rightmost parameters of each ISREDIT CHANGE macro with the corresponding values entered on the Unicenter CA-SpaceMan Installation Worksheet.
2. Store SPEDIT in a library that is concatenated to the SYSPROC DD in your TSO log on procedure.
3. Each time you edit an installation member, type SPEDIT on the TSO command line, and press Enter to replace the generic environment variables in the member with your site specific specifications.

Step 6. Allocate and Load Unicenter CA-SpaceMan Product Files

Note: This step is required whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

Follow these steps to modify JCL member SPUNLOAD in the Unicenter CA-SpaceMan INSTALL library to allocate and load the Unicenter CA-SpaceMan product files.

1. Modify SPUNLOAD.

Use the SPEDIT macro to change the values of the environment variables in the SPUNLOAD member in the Unicenter CA-SpaceMan INSTALL library as specified on the Unicenter CA-SpaceMan Installation Worksheet.

Note: The Unicenter CA-SpaceMan Source and Load tape file members are to be inserted into the Unicenter CA-APAS SOURCE and LOAD data sets.

2. Submit the SPUNLOAD JCL.

If you receive a non-zero return code, correct the JCL and re-submit the job. You may use SPDELETE in the INSTALL library to delete any data sets allocated with SPUNLOAD.

Step 7. Verify the Basic Unicenter CA-SpaceMan Reporting Facility

Note: If you are migrating from a previous release of Unicenter CA-SpaceMan, this step is optional. This step is strongly recommended if you are installing for the first time.

The objective of this step is to become familiar with the basic batch reporting capabilities of Unicenter CA-SpaceMan. This step also verifies that the Unicenter CA-SpaceMan load modules are usable.

The SPMREP program reads a physical database and produces reports about various conditions of the database and its logical files. SPMREP accesses the database through the standard Adabas ADAIOR module.

Follow these steps to modify the Unicenter CA-SpaceMan SPMREP JCL, XSPMREP.

1. Modify XSPMREP.

Use the SPEDIT macro to change the values of the environment variables in the XSPMREP member in the Unicenter CA-APAS SOURCE library as specified on the Unicenter CA-SpaceMan Installation Worksheet

2. Submit the XSPMREP JCL.

If you receive a return code greater than 4, review the errors and take appropriate corrective steps. Resubmit the job if necessary.

For detailed information about the SPMREP control cards, see the chapter "SPMREP Database Analysis Reports." For examples of the reports themselves, see the chapter "Using and Understanding SPMREP Reports" in this manual.

Unicenter CA-SpaceMan Repository and Natural Interface

This section provides step-by-step instructions you use to install the Unicenter CA-SpaceMan Repository facility and Natural Interface. You may wish to modify some of the steps for your particular installation.

Step 1. Load the Unicenter CA-SpaceMan Repository File

Note: This step is required whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

If you are installing Unicenter CA-SpaceMan for the first time or wish to create a new file, proceed with the following section Load a New Repository File.

If you are an existing Unicenter CA-SpaceMan user, skip to the section Modify an Existing Repository File.

Load a New Repository File

If you do not already have a Unicenter CA-SpaceMan Repository file or wish to create a new file, you need to load a new, empty file. This can be done in one of the following two ways:

Approach 1

Execute the appropriate Adabas ADACMP and ADALOD utilities for the database you intend to load the repository file to. The necessary ADACMP field definition statements required for input to ADACMP are in the SPMCMP source member in the Unicenter CA-APAS SOURCE library.

Approach 2

Execute the appropriate Adabas ADALOD utility including as the DDEBAND input data set the empty file unloaded from the distribution tape by the unload job, SPUNLOAD.

hlq.mlq.SPMvvvv.UNLOAD

Alternatively, you can input the data set directly from the distribution tape:

Tape File Seq #18 - CA.APASvvvv.SPACEMAN.UNLOAD

The field definition statements are in the SPMCMP source member in the Unicenter CA-APAS SOURCE library.

Modify an Existing Repository File

If you have an existing Unicenter CA-SpaceMan Repository file, you may need to add some new field definitions to the file and change some of the field definitions before attempting to execute the new Unicenter CA-SpaceMan programs against the file.

If you are upgrading from V3.4 or V3.5, complete the following steps to add the new fields, change some existing fields, and then release and re-invert some superdescriptors:

1. Using the NEWFIELD function of the Adabas ADADBS utility, add the new field definitions found in the SPMCNUF source member in the Unicenter CA-APAS SOURCE library.
2. No existing fields or superdescriptors changes required for V4.1.

Step 2. Load the Unicenter CA-SpaceMan Natural Components

Note: This step is required whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

This version of Unicenter CA-SpaceMan is for use with Natural V3.1 and higher.

Load the Unicenter CA-SpaceMan DDMs, source, and object members into the appropriate Natural System file under the SPACEMAN library-id or some other id of your choice. The program DGRLDNAT in the Unicenter CA-APAS LOAD library is used to load the DDMs, source and object members.

Any source or object members already in the SPACEMAN library are replaced if their names match those members unloaded from the tape. All other old members in the library are unaffected. The same is also true for the DDMs.

Follow these steps to load the Unicenter CA-SpaceMan Natural DDMs, source, and object members:

1. If using Natural security, you must define the Unicenter CA-SpaceMan application id. This is the value of the SPMLIBID variable, as defined in the Installation Worksheet.
2. Use the SPEDIT macro to change the values of the environment variables in the LDSPMNAT member in the Unicenter CA-APAS SOURCE library.
3. Submit the LDSPMNAT JCL to load the Natural DDMs, source, and object members. If you receive a non-zero return code, correct the JCL and resubmit the job.
4. Logon to Natural and use the online Natural utility SYSDDM to change the Unicenter CA-SpaceMan DDMs database id and file number to refer to the Repository file loaded in Step 1. Load the Unicenter CA-SpaceMan Repository File in this chapter.

Change only the DDMs that begin with SPACEMAN- and SPMAN-. Do NOT change the FUSER-TEXT DDM; it must remain set for DBID=255 and FNR=0.

5. Look at the source program SPACEMAN in the Unicenter CA-SpaceMan Natural library. If you wish, edit the statements which set the default values for online navigation mode and display of lowercase characters, then use the SAVE command to save the revised source program.

6. If you changed any of the application ids INSIGHT, APASHIST and/or SPACEMAN, you must make the following changes to the following Natural source programs.

Modify the GUIDMAIN program in the Unicenter CA-SpaceMan library by changing the library-ids INSIGHT and APASHIST in the lines STACK TOP COMMAND 'LOGON' 'INSIGHT' and STACK TOP COMMAND 'LOGON' 'APASHIST' to the application ids you specified. Use the SAVE command to save the program.

Modify the MENU program in the History System library by changing the library-id SPACEMAN in the line STACK TOP COMMAND 'LOGON' 'SPACEMAN' to the application id you specified. Use the STOW command to save and catalog the modified version of the MENU program.

Modify the INMENU program in the Unicenter CA-APAS library by changing the library-id SPACEMAN in the line STACK TOP COMMAND 'LOGON' 'SPACEMAN' to the application id you specified. Use the STOW command to save and catalog the INMENU program.

7. Execute the GLOBALS command and note the value of the decimal character (DC) parameter for your Natural session. Change it to the period, ., if it is any other value.
8. Execute the Natural CATALOG command to catalog all the programs that refer to the Unicenter CA-SpaceMan DDMs.

Note: For Unicenter CA-SpaceMan users who installed the new Natural code in an existing Unicenter CA-SpaceMan library, delete the source ONLY of CENTRTXT. This subprogram is an object only distributed module whose source was mistakenly distributed in an earlier release. The old source must be deleted so the new object module is not overwritten by CATALOG.

- a) On the Natural V3.1 Catalog Objects CATALOG screen, mark the options:

```

02:50:22          ***** NATURAL CATALOG COMMAND *****          2002-11-04
User ABEJU01      - Catalog Objects in Library -          Library SPACEMAN

Catalog Objects from .. * _____ (start value, range, input list)
                  to .... _____ (end value)

Select object types:
  Global data areas
  Parameter data areas
  Local data areas
  Copycodes
  Texts
X External Subroutines
X Subprograms
  Help routines
  Maps
X Programs
  Classes

X Recatalog only existing modules
  Catalog all sources
Select function:
  Save
  X Catalog
  Stow
  Check
Select options:
  Condition code in batch
  Renumber source-codes lines
  Keep result list
  X Processing information
  X Error report

Command ==>>>
    
```

- b) After the CATAL operation has finished, examine the error report screen. It should list only Error 82 and only for the following programs.

CENTRTXT	NATSPCBT
NATSPCOB	NATSPCOL
NATSPCPR	SPHLMGR
SPMNUMGR	SPRECOVR

Ignore these errors.

- c) If you loaded the Unicenter CA-SpaceMan Natural programs into a library that contained an earlier version of the Unicenter CA-SpaceMan Natural application, then the CATAL could also list:

ADHLMGR	ADRECOVR
ADMNUMGR	SCONVERT

Ignore these errors and scratch these programs since they are no longer needed.

9. If you changed the DC parameter earlier, change it back to its original value.

Step 3. Before Getting Started Online

Note: This step is required whether you are installing for the first time or if you created a new Repository file.

Before you can use the online components of Unicenter CA-SpaceMan, you must first gather database layout and I/O load data and update the Repository file with this information.

For more information, review the chapter “Before Getting Started Online.”

Modify XSPMUPD

XSPMUPD extracts file/extent information from the database for input to the Repository update job (XDIRUPD).

Follow these steps to modify the XSPMUPD JCL in the Unicenter CA-APAS SOURCE library.

1. Modify XSPMUPD.

Use the SPEDIT macro to change the values of the environment variables in the XSPMUPD member as specified on the Unicenter CA-SpaceMan Installation Worksheet.

2. Submit the XSPMUPD JCL.

If you receive a non-zero return code, correct the JCL and re-submit the job.

Modify XDIRUPD

XDIRUPD updates the Unicenter CA-SpaceMan Repository file with component and/or I/O load information.

Follow these steps to modify the XDIRUPD JCL in the Unicenter CA-APAS SOURCE library.

1. Modify XDIRUPD.

Use the SPEDIT macro to change the values of the environment variables in the XDIRUPD member as specified on the Unicenter CA-SpaceMan Installation Worksheet.

2. Submit the XDIRUPD JCL.

If you receive a non-zero return code, correct the JCL and resubmit the job.

Modify XHISUPD

XHISUPD updates the Unicenter CA-SpaceMan Repository file with space usage history information.

Follow these steps to modify the XHISUPD JCL in the Unicenter CA-APAS SOURCE library.

1. Modify XHISUPD.

Use the SPEDIT macro to change the values of the environment variables in the XHISUPD member as specified on the Unicenter CA-SpaceMan Installation Worksheet.

2. Submit the XHISUPD JCL.

If you receive a non-zero return code, correct the JCL and resubmit the job.

Note: For information about controlling the SPMHISUP space usage records posted and maintained in the Repository file, first time users should review the sections Posting Space Usage Data to the Repository and Using Accounting Control Records in the chapter “Reporting and Forecasting DASD Space Usage.”

Step 4. Verify Unicenter CA-SpaceMan Natural Interface and Repository

Note: This step is not required but is recommended whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

Verify Unicenter CA-SpaceMan Natural Interface

Invoke Natural from a terminal, logon to the SPACEMAN library or other library id you may have used in the LDSPMNAT job stream and execute the command SPACEMAN.

The SPACEMAN Copyright / logo screen is displayed:

```

                COPYRIGHT 2003 COMPUTER ASSOCIATES INTERNATIONAL, INC.
          ALL RIGHTS RESERVED. CONTAINS TRADE SECRETS AND CONFIDENTIAL PROPRIETARY
                INFORMATION OF COMPUTER ASSOCIATES INTERNATIONAL, INC.
          REVERSE ENGINEERING PROHIBITED. COPYRIGHT NOTICE DOES NOT IMPLY PUBLICATION.

SSSSSSS  PPPPPPP  AAAAAAA  CCCCCC  EEEEEEE  MM      MM  AAAAAAA  NN      NN
SS       PP  PP  AA   AA  CC       EE       MMM   MMM  AA   AA  NNN   NN
SS       PP  PP  AA   AA  CC       EE       MMMM  MMMM  AA   AA  NN  N  NN
SSSSSSS  PPPPPPP  AAAAAAA  CC       EEEEE   MM  M  M  MM  AAAAAAA  NN  N  NN
        SS  PP      AA   AA  CC       EE       MM  MM  MM  AA   AA  NN  N  NN
        SS  PP      AA   AA  CC       EE       MM      MM  AA   AA  NN   NNN
SSSSSSS  PP       AA   AA  CCCCCC  EEEEEEE  MM      MM  AA   AA  NN   NN

*****
*           S P A C E   M A N A G E M E N T   F O R   A D A B A S           *
*****

          CA-SPACEMAN VERSION 4.01.01 FOR ADABAS V7 & NATURAL V3
CHANGE SESSION OPTIONS BELOW, IF DESIRED:
          NAVIGATION MODE: G F=FREE, G=GUIDED
          LOWERCASE READOUT: Y Y=YES, N=NO

                                PRESS < ENTER > FOR LIFTOFF
    
```

If the Unicenter CA-SpaceMan logo screen is displayed, you have successfully installed and accessed the Unicenter CA-SpaceMan Natural Interface component.

If you do not receive the above logo screen, review the installation steps in this section.

Verify the Repository

From the Unicenter CA-SpaceMan logo screen, specify NAVIGATION MODE: G and press Enter to access the CA-SPACEMAN GUIDED NAVIGATION MAIN MENU.

1. Select the MAIN MENU function code:
1 CURRENT DATABASES.
2. From the CURRENT DATABASES MENU, select the code:
A SELECT A DATABASE FROM A LIST.

The current database layout, ****CURRENTnnnnn****, is listed for the database that was added with XSPMUPD and XDIRUPD jobs run earlier.

If the current database layout is not listed, review the installation steps in this section.

Unicenter CA-SpaceMan IOLOG Processing

The steps described in this section makes it possible to capture data about the Adabas physical I/O and to subsequently analyze that data and/or post it to the Unicenter CA-SpaceMan Repository file.

This section provides step-by-step instructions you use to install the Unicenter CA-SpaceMan IOLOG facility. You may wish to modify some of the steps for your particular installation.

Unicenter CA-SpaceMan IOLOG Records

Adabas physical I/O activity is captured via a Unicenter CA-SpaceMan exit from the Adabas ADAIOR module. The Unicenter CA-SpaceMan exit routine, DBGIOR5, looks at all calls to ADAIOR before and after execution and creates a short log record for each call, the Unicenter CA-SpaceMan IOLOG record. The exit routine creates IOLOG records from either single-user or MPM sessions of an Adabas nucleus.

DBGIOR5 passes the IOLOG records to the Unicenter CA-APAS Data Collector. The records are then processed by the Data Collector in a manner analogous to the processing of Adabas Command Log records for Unicenter CA-APAS. IOLOG records can be written to an output file for later batch processing by SPMNENSU or the Data Collector can use them during an MPM session to produce printed reports, output files, or online displays which may be viewed from the Unicenter CA-APAS Natural interface, Insight. Data collected online may become input to the Repository file, the Configurator and the Utility Control Card Generator.

Step 1. Install the Trace Exit Module DBGIOR5

Note: This step is required whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

The JCLSPMAN member in the Unicenter CA-APAS SOURCE library is provided to show the things that must be added or changed in your standard MPM JCL when you execute an MPM session with the Data Collector supporting Unicenter CA-SpaceMan IOLOG facility.

Modifications include:

- ADARUN Parameters
- Unicenter CA-SpaceMan Control Cards
- Unicenter CA-SpaceMan DD Statements
- Use the SPEDIT macro to change the values of the environment variables in the JCLSPMAN member in the Unicenter CA-APAS SOURCE library as specified on the Unicenter CA-SpaceMan Installation Worksheet.

As each of the necessary MPM modifications is discussed in detail below, use JCLSPMAN to assist with the modifications to your MPM JCL.

ADARUN Parameters

The following ADARUN parameters must be added to the Adabas MPM DDCARD input stream:

```
ADARUN TRACE=YES          (enable the ADAIOR Trace exit)
ADARUN TMOD=DBGIOR5      (the Trace exit module to execute)
```

Refer to the “ADARUN Parameters” in JCLSPMAN.

Note: Modifications to DBGIOR5 for Adabas V7 compatibility now require that the Associator be defined with DISP=SHR in the MPM.

Warning! The parameters are for MPM use ONLY and must NOT be used with SPMREP, SPMUPD, SPMNENSU or any other batch process where the Adabas MPM is not active.

Unicenter CA-SpaceMan Control Cards

Sample Unicenter CA-SpaceMan control cards are shown in the Unicenter CA-APAS SOURCE library member SPMDFLTS.

Use the SPEDIT macro to change the value of the environment variables in the SPMDFLTS member in the Unicenter CA-APAS SOURCE library as specified on the Unicenter CA-SpaceMan Installation Worksheet.

Unicenter CA-SpaceMan DD Statements

DD statements are needed for receiving printed reports and output data requested in the SPMDFLTS requests.

- DGIOSUM is required for the Unicenter CA-SpaceMan I/O Sample summary request, DGIOSUM.
- Other DD statements as needed for ddnames specified in PRINT TO and/or OUTPUT FILE parameters of Unicenter CA-SpaceMan requests.

Refer to “DD Statements For Unicenter CA-SpaceMan” in JCLSPMAN.

Step 2. Implement IOLOG COPY File for Batch Processing

Note: This step is optional.

Unicenter CA-SpaceMan IOLOG records can be written to sequential data sets during an MPM session and processed later with the Unicenter CA-SpaceMan batch IOLOG analysis program, SPMNENSU.

Prepare the MPM

The JCLSPMIO member in the Unicenter CA-APAS SOURCE library is provided to show the things which must be added or changed in your standard MPM JCL.

Use the SPEDIT macro to change the values of the environment variables in the JCLSPMIO member in the Unicenter CA-APAS SOURCE library as specified on the Unicenter CA-SpaceMan Installation Worksheet.

Modify XSPMNENS

The XSPMNENS JCL in the Unicenter CA-APAS SOURCE library is used to execute SPMNENSU against an IOLOG COPY file created during an MPM session.

Use the SPEDIT macro to change the values of the environment variables in the XSPMNENS member in the Unicenter CA-APAS SOURCE library as specified on the Unicenter CA-SpaceMan Installation Worksheet.

Step 3. Execute MPM with Unicenter CA-SpaceMan IOLOG

Note: This step is required whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

Warning! If Unicenter CA-APAS detects errors in its control card input stream (DBGIN) **and** you specified IGNORE-INIT-ERRORS=NO in the GLOBALS statements, it abnormally terminates the MPM with a U0026. It does this to avoid having a session that does not produce the output that you want.

Starting the MPM and Unicenter CA-SpaceMan

Start the new MPM session including the necessary ADARUN parameter changes, DBGIN GLOBALS parameters and automatically started IOLOG requests, plus any additional DD statements needed for receiving output.

Step 4. Verify Unicenter CA-SpaceMan IOLOG Processing

Note: This step is not required but is recommended whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

Invoke Natural from a terminal, logon to the library id specified for Unicenter CA-APAS in the LDINSNAT job stream and execute the command INSIGHT.

```

      ADABAS PERFORMANCE ANALYSIS SYSTEM (CA-APAS)
      COPYRIGHT 2003 COMPUTER ASSOCIATES INTERNATIONAL, INC.
      ALL RIGHTS RESERVED. CONTAINS TRADE SECRETS AND CONFIDENTIAL PROPRIETARY
      INFORMATION OF COMPUTER ASSOCIATES INTERNATIONAL, INC.
      REVERSE ENGINEERING PROHIBITED. COPYRIGHT NOTICE DOES NOT IMPLY PUBLICATION.

      IIII      NNN      NN      SSSSSSS      IIII      GGGGGGGG      HH      HH      TTTTTTTT
      II       NNNN     NN      SS              II       GG              HH      HH      TT
      II       NN NN    NN      SS              II       GG              HH      HH      TT
      II       NN NN    NN      SSSSSSS      II       GG              HHHHHHHH      TT
      II       NN      NN NN      SS          II       GG      GGGG      HH      HH      TT
      II       NN      NNNN      SS          II       GG      GG      HH      HH      TT
      IIII     NN      NNN      SSSSSSS      IIII     GGGGGGGG      HH      HH      TT

-----
I N T E R A C T I V E   A D A B A S   P E R F O R M A N C E   M O N I T O R
-----

      CA-INSIGHT V4.01.01 FOR ADABAS V7 & NATURAL V3

      CMD: _____ REQ: _____ DBID: ____0 SMPID: ____0 > USE PF KEY, CMD, OR ENTER <
      PFK: 1=HELP 3=MENU 4=SELECT 5=SHOW 6=DISPLAY 9=SNAP 12=REQMGT INSTART

```

From the Unicenter CA-APAS logo screen, start the Unicenter CA-SpaceMan request IOTRACE by entering START in the CMD field and IOTRACE in the REQ field and pressing Enter.

Press PF5 to display the SHOW screen and display the IOTRACE request.

If the display of IOTRACE does not contain any IOR FUNC, FUNC or RW data, either the TRACE and TMOD ADARUN parameters were not implemented with this MPM session or the IOLOG-SWITCH is OFF.

Review the appropriate installation steps in this section.

Unicenter CA-SpaceMan Configurator Facility

The steps described in this section enable the online Configurator, the Generate Card Utility, and the OS Space Allocator Utility. The installation of this facility and its utilities is not required.

This section provides step-by-step instructions you use to install the Unicenter CA-SpaceMan Configurator facility and Configurator utilities. You may wish to modify some of the steps for your particular installation.

Configurator Options

The function of the Unicenter CA-SpaceMan Configurator is to optimize the placement of Adabas files and/or data sets based on averaging the I/O load across disk volumes. The Configurator provides three separate options for reorganizing a database:

- Option 1 - Complete Reconfiguration of the Database
- Option 2 - Reallocation of the Database
- Option 3 - Individual File Maintenance

Once you have used the SPMREP reports to analyze your space management problems, you can select one or more of these configuration options as your space management solution.

Information on the general procedure for installing subroutines for use under Natural is given in the Software AG *Natural Operations Manual*.

The Unicenter CA-SpaceMan Configurator subroutines DGFCNF, DGNCNF, and DGSCNF from the Unicenter CA-APAS LOAD library must be made available for use under Natural.

Step 1. Install Unicenter CA-SpaceMan Configurator Subroutines

Note: This step is required whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

Instructions for installation as dynamic subroutines are as follows:

TSO

Move the three Configurator modules, DGFCNF, DGNCNF, and DGSCNF, to the library where the TSO Natural nucleus is being called or executed as a command.

CICS

1. Use the SPEDIT macro to change the values of the environment variables in the LNKCNFC member in the Unicenter CA-APAS SOURCE library as specified on the Unicenter CA-SpaceMan Installation Worksheet.
2. Submit the LNKCNFC JCL to linkedit the CICS versions of the Configurator modules, DGFCNFC, DGNCNFC and DGSCNFC, into a CICS STEPLIB load library.
3. Rename the CICS versions of the Configurator modules in the CICS STEPLIB load library to remove the C suffix:

Rename	DGFCNFC	to	DGFCNF
Rename	DGNCNFC	to	DGNCNF
Rename	DGSCNFC	to	DGSCNF
4. Generate PPT entries for the Configurator modules, DGFCNF, DGNCNF, and DGSCNF (assembler language and non-resident).
5. The subroutines are available the next time CICS is brought up. If the subroutines are put into or replaced in a load library of an active CICS, issue a NEWCOPY for each of the three modules to make them known to CICS.

COM-PLETE

1. Move or link the three Configurator modules, DGFCNF, DGNCNF and DGSCNF, into a COM-PLETE STEPLIB load library.
2. Catalog the subroutines with a ULIB CAT,DGxCNF command.

Step 2. Prepare Unicenter CA-SpaceMan Utility Card Generator JCL

Note: This step is required whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

Modify XGCRD1 JCL

XGCRD1 builds the Adabas utility statements for Option-1 configurations.

Use the SPEDIT macro to change values of the environment variables in the XGCRD1 member in the Unicenter CA-APAS SOURCE library as specified on the Unicenter CA-SpaceMan Installation Worksheet.

Modify XGCRD2 JCL

XGCRD2 builds the Adabas utility statements for Option-2 configurations.

Use the SPEDIT macro to change the values of the environment variables in the XGCRD2 member in the Unicenter CA-APAS SOURCE library as specified on the Unicenter CA-SpaceMan Installation Worksheet.

Modify XGCRD3 JCL

XGCRD3 builds the Adabas utility statements for Option-3 configurations.

Use the SPEDIT macro to change the values of the environment variables in the XGCRD3 member in the Unicenter CA-APAS SOURCE library as specified on the Unicenter CA-SpaceMan Installation Worksheet.

Modify Natural JCL Members

The Natural members V6FLOAD, V6REORFIL and V6UNRLD are used by the GCRD function to generate JCL to execute a series of Adabas utilities.

Follow these steps to modify the Natural JCL members in the Unicenter CA-SpaceMan Natural library.

1. Invoke Natural from a terminal and logon to the SPACEMAN library or other id you may have used in the LDSPMNAT job stream.
2. Edit V6FLOAD. Modify this member to conform to your local JCL standards and Adabas procedures and save it.
3. Edit V6REORFIL. Modify this member to conform to your local JCL standards and Adabas procedures and save it.
4. Edit V6UNRLD. Modify this member to conform to your local JCL standards and Adabas procedures and save it.

Step 3. Prepare Unicenter CA-SpaceMan OS Space Allocator JCL

Note: This step is required whether you are installing for the first time or migrating from a previous release of Unicenter CA-SpaceMan.

Modify XSPMALC

The Unicenter CA-SpaceMan OS Space Allocator uses standard OS space allocation routines to allocate data sets in a more flexible fashion than is possible using only JCL. To access these routines, SPMALC must be executed from an authorized library.

Use the SPEDIT macro to change values of the environment variables in the XSPMALC member in the Unicenter CA-APAS SOURCE library as specified on the Unicenter CA-SpaceMan Installation Worksheet.

Note: The SPMALC utility cannot be used on SMS controlled volumes.

Unicenter CA-SpaceMan Functionality

Unicenter CA-SpaceMan File Management for Adabas (Unicenter CA-SpaceMan) offers a comprehensive solution to the problems of physical space management in your Adabas installation. You will find the major aspects of Unicenter CA-SpaceMan use described below.

Features

Unicenter CA-SpaceMan addresses the problems of physical space management through the entire space management cycle: analyzing the database (the Physical Database reports), creating a modifiable space management plan (the “Projection”), generating a new database image (the “Configuration”) and executing the plan.

Using Unicenter CA-SpaceMan, you can:

- Analyze the internal logical structure of the database completely, highlighting problem areas and identifying any need to reposition components
- Analyze I/O loads for selected periods, identifying bottlenecks and load imbalances
- Combine the space analysis and I/O load analysis to allow repositioning components optimally
- Centralize data necessary for loading and maintaining Adabas files into the Unicenter CA-SpaceMan Repository file, which allows capacity planning based on a reliable history of file space usage
- Automatically generate utility control cards to implement your new database configuration

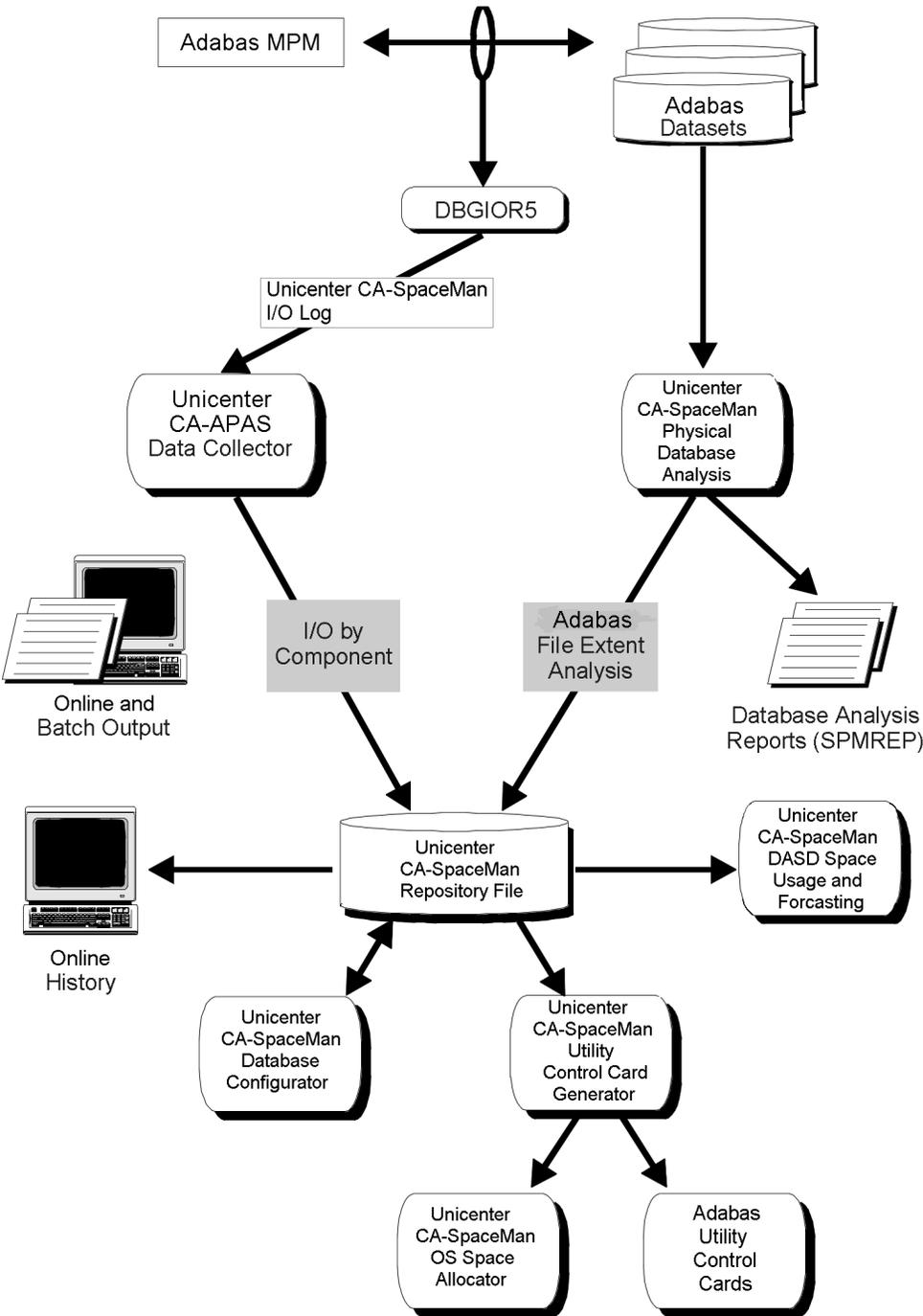
Unicenter CA-SpaceMan has a range of facilities and options that allow it to grow with the requirements of your organization.

Unicenter CA-SpaceMan Components

Unicenter CA-SpaceMan components perform the following major functions:

- The physical database analysis program comprehensively reports the internal state of the Adabas database files.
- The Data Collector monitors physical I/O to Adabas data sets, providing breakdowns of this activity on a real-time basis or as printed reports or machine-readable files. The Data Collector collects data from the I/O Log, which is generated by the DBGIOR5 module.
- The Repository file maintains a directory of current Adabas databases and files, I/O samples, projections, new configurations, and a history of Adabas file growth.
- The Database Configurator automatically generates optimal configurations for a database based on space available and measured I/O loads.
- The Utility Control Card Generator (GCRD) generates Adabas utility control cards for maintaining (reorganizing, reallocating, or reloading) the database.
- The OS Space Allocator (SPMALC) allocates multiple extent data sets quickly and easily across DASD volumes.

The following graphic, Unicenter CA-SpaceMan: Major Components and Data Flow shows all of the major Unicenter CA-SpaceMan components and the basic data flows between them. Different space management tasks use different combinations of the components. For more information on how to use the components, see Unicenter CA-SpaceMan Database Maintenance Options in this chapter.



Physical I/O Monitoring and Tuning

Physical I/O activity is captured by monitoring calls to the Adabas module (ADAIOR) which performs all operating system functions for the Adabas nucleus.

Unicenter CA-SpaceMan I/O Logging

Data about each call is captured and then processed by the Data Collector module. Unicenter CA-APAS users are familiar with both the command syntax and operation of the Unicenter CA-SpaceMan Data Collector, since it is an extension of the Unicenter CA-APAS Data Collector.

The data produced on physical I/O loads and performance has several uses:

- The information can be easily correlated with data collected by other monitors that show I/O performance at the volume level. It is possible to tell which component of a database is being so heavily accessed as to cause an I/O bottleneck.
- The data forms the basis for reconfiguration of the database to balance loads across disk devices.
- Low level traces of I/O calls show the exact functioning of Adabas and possible performance bottlenecks.

Batch Reporting of Physical I/O Activity

Writing physical I/O activity data directly to a disk log requires minimal overhead in the Adabas processing region. The log is generally an X/Y set of files with provision for automatically switching from one to the other when one file becomes full. Creating the disk log adds less than 1 per cent to I/O activity in the Adabas region. Proper placement of the disk log data sets avoids any contention with Adabas data sets; there should be no increase in Adabas response time.

Once a log has been created, it can be summarized with the Data Collector program executed in stand-alone batch mode. This is a report writer which gives you flexibility to select and summarize I/O activity. Summary information can be written to machine-readable files or as formatted reports.

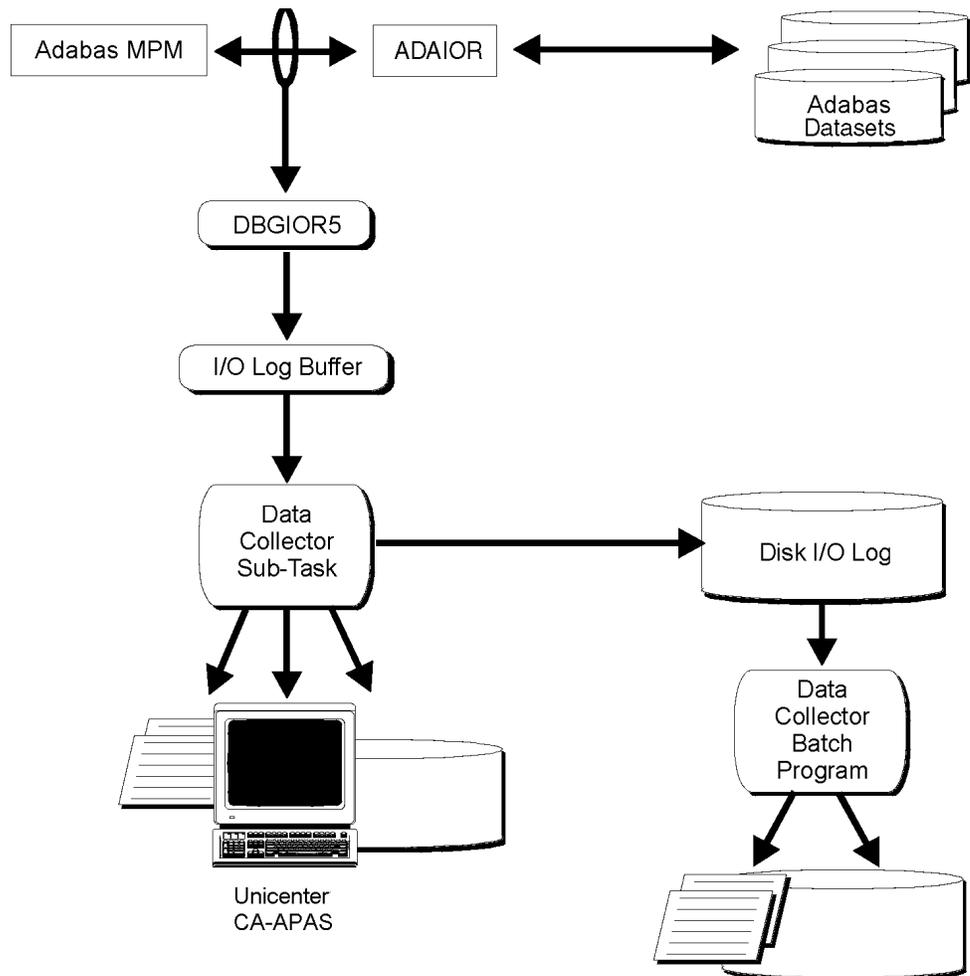
Real-time Reporting of Physical I/O Activity

Physical I/O log data can be passed directly to the Data Collector operating in MPM mode. In MPM mode, the Data Collector functions as a subtask when the operating system supports subtasking; this assures minimal impact on throughput of the Adabas nucleus.

Users of Unicenter CA-APAS, the Adabas performance monitor, can create, add, view and delete report requests interactively.

The same output file capabilities and reporting options are available from the Data Collector when executing in either batch or MPM mode.

The following shows Unicenter CA-SpaceMan: The Data Collector in MPM Mode



Reporting of Physical I/O Activity

Database Considerations

The following are database considerations that affect I/O activity.

- Optimize LBP size for best Adabas I/O buffer efficiency without excessive paging overhead.
- Optimize LWP/LS size to reduce I/O to Work Part 2.
- Consider a larger blocksize for Work if there is heavy complex search activity to Work Part 2.
- Consider a larger blocksize for Data Storage if most file accesses are sequential reads.
- Consider a larger blocksize for Associator if logical sequential reading far exceeds searches.

File Considerations

The following are file considerations that affect I/O activity.

- Release unnecessary descriptors to decrease index I/O during add, update, and delete processing.
- Use periodic groups for repeating sets of fields to reduce the number of records that must be read.
- Reorder DATA of a file by the descriptor that is used most frequently for READ LOGICAL commands.
- Use ADAM files where there are frequent searches on a unique key. Use adequate bit truncation when READ LOGICAL is done on the ADAM key.

Application Considerations

The following are some application considerations that affect I/O activity.

- Simplify search criteria and use super-descriptors to reduce index I/O and Work Part 2 I/O.
- Use HISTOGRAM or READ LOGICAL in place of a FIND with one descriptor and a range of values.
- Use READ PHYSICAL with a WHERE clause when processing over 15 per cent of the records in a file.
- RETAIN ISN lists to avoid re-reading the Upper and Normal Indexes.
- Make judicious use of OBTAIN/GET SAME to minimize re-reads of data records.

Physical I/O Reports

Reports are of two types:

- SUMMARIZE requests produce summaries of I/O broken down by a variety of user-selected criteria.
- EXTRACT requests produce detailed reports of selected individual I/Os and other system function requests.

Requests can direct their output to a report, a machine-readable file, or a user-supplied exit routine. Each request specifies the fields to be reported and flexible selection criteria for which I/O log events are to be reported. A variety of options are available for determining the output format of the reports and data.

SUMMARIZE Requests

A SUMMARIZE request specifies the output of summary functions for I/O grouped by specified fields. These summaries are useful for a wide variety of measurements of I/O loads and performance.

The following is the output from sample request VOLCYL, supplied with Unicenter CA-SpaceMan for measuring I/O loads on physical cylinders.

SUMMARY OF I/O BY VOLUME BY CYLINDER BY 10			
VOLSER	CYLN	I/O COUNT	MEAN DURAT SECS
-----	-----	-----	-----
GSW001	60	282	0.49147
GSW001	70	159	0.08003
GSW001	80	140	0.12114
GSW001	90	32	0.10977
GSW002	150	103	0.23445
GSW002	160	31	0.33977
GSW002	170	1	0.00094

This report summarizes I/O in 10-cylinder increments. For example, 70 means 70 to 79. The report gives a picture of loads at the device level which can be correlated with data from operating-system monitors.

The following is the output from sample request IOSUM, supplied with Unicenter CA-SpaceMan for measuring separate READ and WRITE loads according to logical file components.

SUMMARY OF I/O BY FILE AND COMPONENT						
FILE	FUNCT	FILE COMPT	R W	I/O COUNT	SUM DURAT SECS	MEAN DURAT SECS
<hr/>						
0	WORK	W1	R	1	0.271	0.27085
0	WORK	W1	W	6	0.875	0.14578
0	WORK	W3	R	73	39.153	0.53634
0	WORK	W3	W	2	0.002	0.00099
1	ASSO	AC	R	1	0.001	0.00099
1	ASSO	FC	R	2	0.665	0.33247
1	ASSO	FD	R	1	0.001	0.00093
1	ASSO	NI	R	2	0.002	0.00091
1	ASSO	UI	R	3	0.003	0.00096
1	DATA	DA	R	29	10.466	0.36090
5	ASSO	AC	R	1	0.535	0.53494
5	ASSO	FC	R	2	0.605	0.30269
5	ASSO	FD	R	1	0.001	0.00091
5	ASSO	NI	R	1	0.001	0.00090
5	ASSO	UI	R	2	0.002	0.00094
5	DATA	DA	R	1	0.001	0.00094

This summary of a minute of processing shows the I/O load on Part 3 of the Work data set generated by GET NEXT processing from a Natural FIND loop. For example, two writes were done to place the ISN list on Work Part 3 and 73 reads were done to retrieve the list.

The following report summarizes READS and Total I/O for each volume. A volume whose READ activity exceeds 80 per cent of its total I/O should be considered for placement behind a cache controller.

DETERMINE READ/WRITE RATIO FOR CACHE PLACEMENT		
VOLSER	TOTAL I/O	SUM READS
<hr/>		
GSW001	1,365	1,358
GSW002	1,209	851
GSW004	560	477

EXTRACT Requests

An EXTRACT request specifies the output of data values for individual physical I/Os or other operating system function calls. You do not ordinarily need to trace I/O to this level, but the capability proves useful in understanding the functioning of Adabas and its interaction with hardware such as cache storage.

A standard request, IOTRACE, is supplied with Unicenter CA-SpaceMan to report the activity of the ADAIOR module.

TRACE OF ADAIOR ACTIVITY

IOR FUNC	R FUNCT	W	RABN	FILE	FILE COMPT	C M D	DESC	HEAD TRAVEL	VOLSER	CYLN
01	ASSO	R	44	14	FC	S1		3	GSW002	302
01	ASSO	R	103	14	FD	S1		0	GSW002	302
01	ASSO	R	45	15	FC	S1		0	GSW002	302
03	DATA	R	3,440	14	DA	S1		4	GSW003	869
03	DATA	R	3,540	15	DA	S1		1	GSW003	870
01	ASSO	R	548	15	UI	S1	AA	1	GSW002	303
01	ASSO	R	547	15	UI	S1	AA	0	GSW002	303
01	ASSO	R	560	15	NI	S1	AA	1	GSW002	304
01	ASSO	R	497	14	AC	S1		1	GSW002	303
03	DATA	R	3,437	14	DA	S1		1	GSW003	869
01	ASSO	R	523	14	UI	S1	AA	0	GSW002	303
01	ASSO	R	522	14	UI	S1	AA	0	GSW002	303
01	ASSO	R	535	14	NI	S1	AA	0	GSW002	303
03	DATA	R	3,537	15	DA	S1		1	GSW003	870
03	DATA	R	3,403	14	DA	L1		1	GSW003	869
03	DATA	R	3,411	14	DA	L1		0	GSW003	869
03	DATA	R	3,423	14	DA	L1		0	GSW003	869
01	ASSO	R	498	15	AC	L1		0	GSW002	303

This trace shows accesses to individual descriptor blocks generated by a FIND(S1) command. Fields shown include the I/O function and the cylinders of head movement since the previous I/O on a given volume.

Unicenter CA-SpaceMan Database Maintenance Options

Significant performance improvements can often be obtained by reconfiguring an Adabas database with careful placement of all components, balancing I/O across DASD volumes. Unfortunately, because this work has been highly labor-intensive, few organizations have been able to afford the necessary manpower to manage their database at this level.

Configurator Options

Unicenter CA-SpaceMan database maintenance facilities automate these functions, making them available to every DBA. For more detailed information about the database maintenance functions and how to use them, including a tutorial exercise using the online functions, see the chapter “Getting Started Using Online Unicenter CA-SpaceMan.”

Unicenter CA-SpaceMan helps you maintain the Adabas database at several levels of complexity:

- Option 1: Completely reorganizing the database, both reallocating it across volumes and reorganizing it internally.
- Option 2: Reallocating the database across multiple DASD volumes without internal reorganization of the database.
- Option 3: Adjusting and adding individual files.

Each of these reorganization procedures balances I/O across physical volumes to the maximum extent possible, thus achieving optimal performance with available resources. The following is a brief overview of the database maintenance functions Unicenter CA-SpaceMan provides.

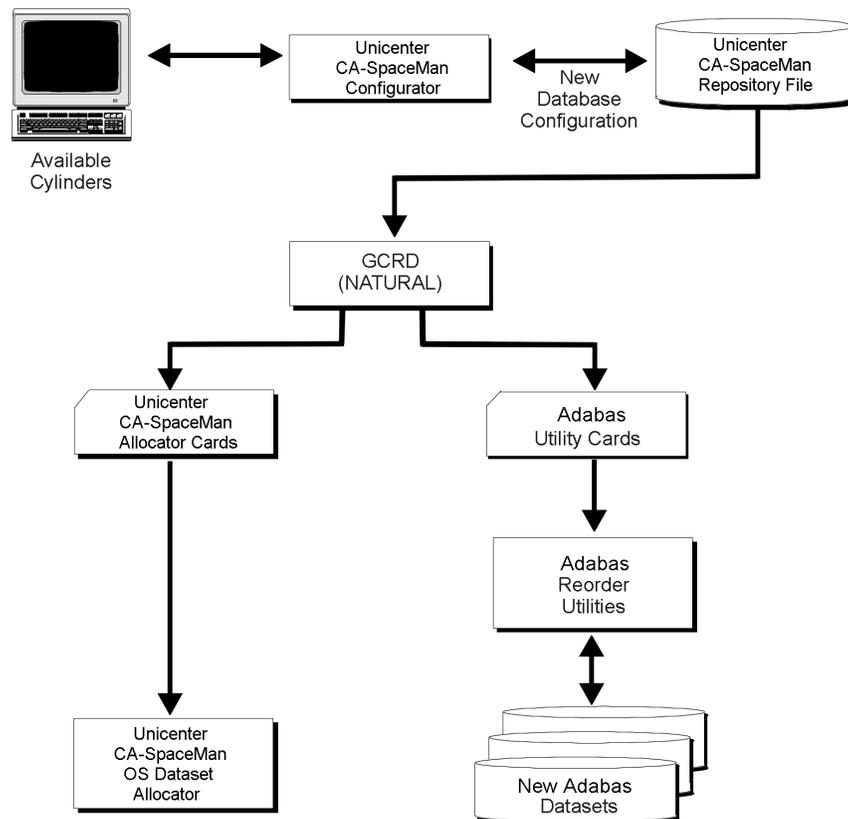
Completely Reorganizing the Database

This procedure both reallocates the database and reconfigures it internally. The DBA enters volumes and available space and selects an I/O sample with an I/O load to be balanced. The Unicenter CA-SpaceMan Configurator then generates a new configuration, placing each file component within volumes and determining the data set allocations to balance the I/O.

You should perform the following reorganization procedure:

- SAVE the database to a backup data set
- Use the generated Unicenter CA-SpaceMan OS Space Allocator control cards to reallocate Adabas data sets
- RESTORE the database from the backup data set
- Use the Adabas utility control cards generated by Unicenter CA-SpaceMan to reorder the Adabas Associator and Data Storage

This procedure optimizes the database by carefully placing every component of every file. A complete reorganization can require a good deal of processor time, but it results in the best possible database performance. The following diagram shows Unicenter CA-SpaceMan New Database Configuration.



Reallocating the Database

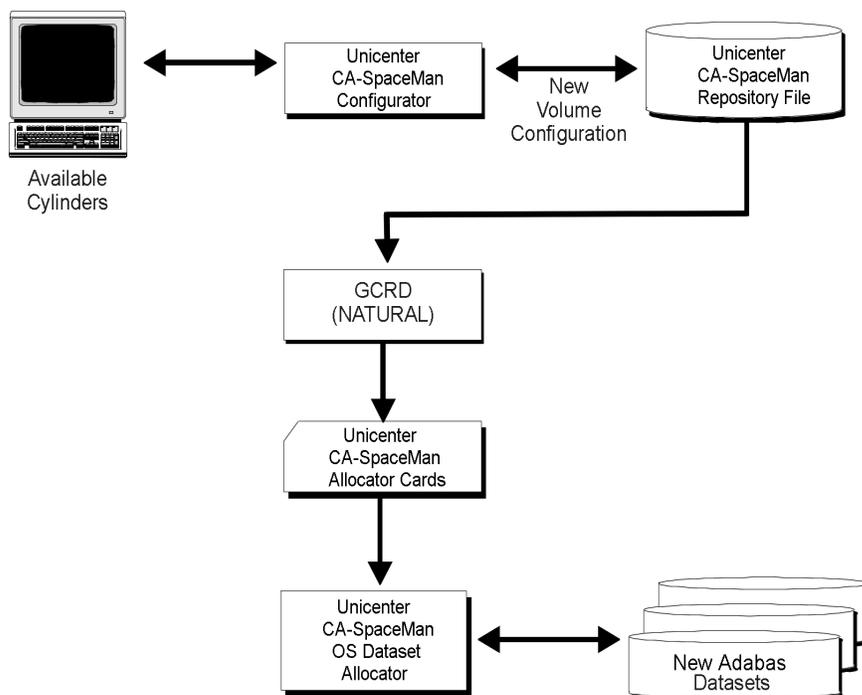
This method of database reconfiguration achieves the best possible load balance across physical DASD volumes without changing the internal location of database components.

The DBA specifies a set of volumes and space available on the volumes. A sample I/O load is selected and the Configurator partitions the Adabas data sets in such a way as to equalize I/O load for this sample as much as possible.

To reconfigure the database, you must:

- SAVE the database to a backup data set
- Use the generated Unicenter CA-SpaceMan OS Space Allocator control cards to reallocate Adabas data sets
- RESTORE the database from the backup data set

The following diagram shows Unicenter CA-SpaceMan: New Volume Configuration.



This procedure has the advantage of minimal resource usage and complexity for the benefits gained. However, it does not obtain the benefits of fully organizing and optimizing the internal structure of the database.

Maintaining Individual Files

Unicenter CA-SpaceMan allows incremental growth and maintenance of the database to be done simply and largely automatically. Using Unicenter CA-SpaceMan, you can automatically generate control cards necessary to unload and reload files with explicit control over optimal placement on DASD volumes.

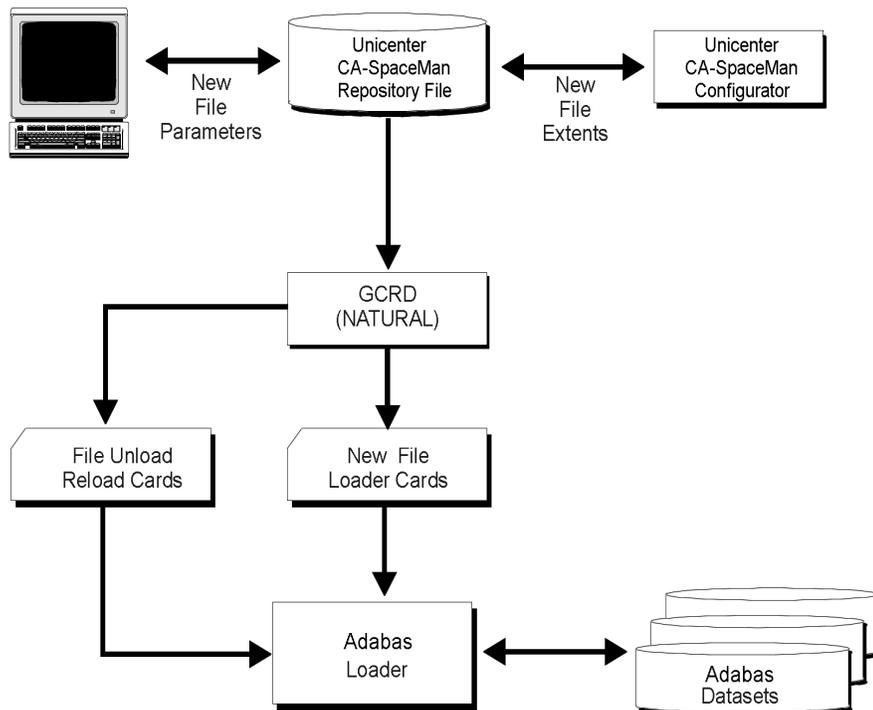
Allocating New Files

Unicenter CA-SpaceMan allows the DBA to enter loader parameters for new files along with projected I/O loads. The Repository system helps the DBA to find comparable files and their I/O loads. The Configurator is then invoked to find the best possible allocation for the new file's components. The GCRD program generates control cards for the Adabas loader utility.

Reorganizing Existing Files

You can generate unload/reload control cards for any existing file in the database. This procedure is especially recommended for files whose components have gone into multiple extents. Select a set of files to be unloaded and reloaded when the database has become too fragmented to reallocate a large file.

The following diagram shows Unicenter CA-SpaceMan: Reorganizing Existing Files.



Understanding Adabas Space Management

You can execute the Unicenter CA-SpaceMan File Management for Adabas (Unicenter CA-SpaceMan) physical database report program as soon as the Unicenter CA-SpaceMan libraries have been loaded from the installation tape, allowing you to generate valuable space analysis information immediately. You can also use these reports to regularly monitor your database on the file or component level.

For a more complete picture, you can extract and analyze I/O loads from standard Adabas Command Logs or with the more comprehensive data available from the Unicenter CA-SpaceMan I/O capture facility.

Once you have analyzed and defined your space management problems, you can use Unicenter CA-SpaceMan to help execute the solution you select – from a simple unload or reload of a single file to total database reconfiguration.

Because controlling Adabas file placement on volumes can be difficult and time-consuming, balancing I/O loads across physical disk devices is seldom attempted.

Unicenter CA-SpaceMan makes this very important aspect of database tuning an almost automatic process. Information from the Unicenter CA-SpaceMan Repository file is combined with user-supplied information on available disk space to generate a balanced configuration for the database. The algorithm used takes into account the internal structure of the Adabas database, current I/O peak activity, and “rules of thumb” for placement of database components.

Unicenter CA-SpaceMan Analysis

Much of the data you can obtain and analyze using Unicenter CA-SpaceMan is not available from any other source. Some of the areas analyzed are described below.

Padding Factors

The degree to which both Data Storage and Associator padding areas have been used is fully reported, indicating when padding areas should be reestablished and whether padding factors should be adjusted.

Descriptors

Both static and dynamic information is available on index (descriptor) usage. Reports of space usage and value distribution for descriptors are extremely useful in identifying descriptors that can be eliminated. Information on which descriptors are actually accessed by physical I/O is available from the I/O logs.

ADAM Files

The actual placement of ADAM file records is analyzed, showing the number of records which are in blocks to which they did not randomize.

Physical I/O Breakdown

The Adabas component, disk volume and cylinder, and the duration of individual I/Os are available. Disk head movement is also computed.

Work Data Sets

Usage of each of the four parts of the Work data set is analyzed, indicating whether sizes or locations should be changed.

Areas of Application

Unicenter CA-SpaceMan contains a range of facilities addressing all levels of disk space management for the Adabas database. Space management is related to a variety of other areas of database administration. Some of the areas in which you can use Unicenter CA-SpaceMan reporting and configuration options are discussed below.

Managing Adabas Logical Files

Each Adabas database may contain up to 65535 physical files. Each file consists of a number of components and extents. Monitoring and managing the internal structure of these database files is one of the primary tasks of the Adabas database administrator.

Unicenter CA-SpaceMan offers comprehensive reports of the internal structure of the database, and points out areas where attention is warranted. Some of the specific areas addressed are:

- Proper setting of file loading parameters
- Files needing adjustments in any aspect of space usage

Managing DASD

The allocation of Adabas files to physical devices is transparent to the logical functioning of Adabas, but is a significant performance factor. Unicenter CA-SpaceMan fully reports Adabas I/O at the physical level, recommends optimal allocation schemes for the database, and offers tools to easily perform this allocation.

File Design and Application Tuning

Unicenter CA-SpaceMan offers data on index (descriptor) usage that is not available from other sources. Using Unicenter CA-SpaceMan, you can evaluate the allocation and usage of descriptors and adjust file and application design accordingly.

The capability of Unicenter CA-SpaceMan for tracing physical I/O permits micro-analysis of I/O patterns for extremely time-critical applications and for understanding conflicting access requirements within or between applications.

Real-Time Performance Monitoring

Unicenter CA-SpaceMan complements other operating system performance monitoring tools which report I/O activity at the physical device level or the OS file level. Unicenter CA-SpaceMan reports the distribution of activity within the Adabas logical components, providing the means to more fully understand I/O contention problems.

Capacity Planning

The Unicenter CA-SpaceMan Repository file maintains a history of file size and I/O loads, allowing you to more accurately predict future growth and DASD requirements by application.

Chargeback and Accounting

Unicenter CA-SpaceMan maintains accurate accounting of space allocated and used by logical files for easy chargeback. In the case of Natural system files, space can be accounted for by individual application.

Debugging

Unicenter CA-SpaceMan allows detailed tracing of all or selected I/O, giving an exact record of disk accesses.

Space Management

This section is a survey of space management both for general consideration and for those areas which are of specific concern to Adabas users.

General Space Management

Cost of DASD	One approach to a perceived shortage of space is to purchase additional DASD and its associated hardware. However, it may be possible to delay or avoid additional purchases by better utilizing the available space. Look for data sets which are not being used, data sets which are used but over-allocated in size, and data sets whose organization could be improved to reduce their space requirements.
Cost of I/O Processing	<p>Actual and billed cost, of course, depends upon the hardware in use. However, fewer and more efficient I/Os result in less expensive online transactions with better online response, and less expensive batch job executions.</p> <p>Each I/O uses CPU time, as well. A reduction in I/O eases the load on the CPU.</p>
I/O Queueing	<p>A disproportionate amount of I/O activity to any one volume may result in a backlog queue of I/O requests, causing delays in throughput. Distributing the same I/O load across other, less active, volumes may lessen or remove that backlog.</p> <p>A common cause of occasional degraded response time is a particular work mix that causes a relatively short period of I/O queueing. Looking at I/O loads over long intervals do not detect relatively brief periods where a great deal of I/O activity was concentrated on one device. I/O load balancing should be based on analysis of short intervals and should aim to eliminate the worst problems.</p>
Head Movement	I/O delays can also result when the device read/write heads must physically travel from one cylinder location to another. In a random access environment, it is difficult to predict where one I/O will occur in relation to the previous I/O. However, with specific information about I/O load activity for a volume, it is possible to place files in such a way as to greatly reduce the average head movement between accesses.

Adabas Space Management

Adabas Space Structures

The Adabas database consists of:

An Associator data set which contains all the pointers and descriptions necessary to find and interpret data records

- A Data Storage data set which contains all of the database data records
- A Work data set which is used for restart/recovery in the event of abnormal termination, and also as a temporary file for transient information, such as ISN lists
- An optional Command Log data set which is a sequential file of all commands that pass through the Adabas nucleus
- An optional Protection Log data set which is a sequential file of the before and after images of changed records
- TEMP and SORT data sets for use by various utilities

The Associator and Data Storage areas may consist of up to five separate data sets each, intermixed over any number of physical volumes. Work must be a single data set, but it is logically divided into three or four parts, each of which should be considered separately for space management purposes. The components of Adabas logical files reside within the Associator and Data Storage data sets.

Data Set Positioning

The relative positioning of the various Adabas data sets should be considered in any significant space management effort. Without any actual I/O statistics, many DBAs tend to simply separate Associator from Data in an attempt to reduce contention and balance the I/O load across available space. However, files are not equally loaded or equally accessed. Without actual statistics of all I/O at the cylinder level, repositioning data sets may not reduce the delays caused by high activity at peak processing times.

The Work data set is usually separated from Associator and sometimes placed with Data Storage. Again, this is usually based on the DBA's estimate of where most I/O activity takes place. The relative I/O load factors for each part of the Work data set are discussed below.

- The number of writes to Work Part 1, called the LP area, is dependent upon the number of updates that are done to the database.
- Work Part 2, called the LZ area, could be the most heavily used area of the database if frequent complex searches are part of normal processing, or it may be seldom used if these searches are discouraged. An I/O monitor is needed to identify your specific situation.

- The ADARUN parameter NSISN can be used to reduce physical I/O to Work Part 3.
- Work Part 4 will depend upon the amount of distributed transaction processing.

Due to the differences in the I/O activity for the four parts of Work, it is sometimes better to allocate each part so that its respective I/O load complements the remaining load on the volume. Knowing the size and total I/O count for the Work data set is not sufficient to perform this allocation; you must know the size and I/O load of each distinct part in order to effectively position data sets.

Logical File Positioning

Initial extents for each of the four logical parts of an Adabas file (Upper Index, Normal Index, Address Converter and Data Storage) may be positioned independently of the other parts by means of loader parameters. You should try to keep upper and normal indexes for the same file together. It is less important for the Address Converter to be located nearby. Data Storage for a file should be on a different volume than its Associator components, and even a different channel, if possible.

If any initial extent is exceeded, Adabas automatically allocates up to four additional extents of the same type. Since the DBA has little control over the sizes or locations of such additional extents, it becomes important to monitor them closely or to avoid them altogether. However, some control over the size of additional extents is provided with the MAXDS, MAXNI, and MAXUI parameters of the ADALOD utility. Carefully setting these parameters avoids situations where a large additional extent is allocated to accommodate just a few overflow records, leaving most of the extent wasted and unused.

Identify heavily used components from different files. These components may then be placed adjacent to one another on a volume, thereby concentrating I/O activity to a fewer number of cylinders, and reducing the average head movement between I/Os.

Using information about I/O loads by file components, you can allocate an Adabas file in a manner that minimizes head movement on each volume and reduce contention between volumes. The result is shorter command durations and increased throughput.

Blocksizes

Adabas uses default blocksizes for Associator, Data and Work data sets and provides artificial device types, such as the 8380, which may be used when you want larger blocksizes.

Often a single file is identified which can benefit from a larger blocksize, but using the artificial device type increases the blocksize for all file components of the same type. This would be detrimental to those files with heavy random access. In this case, a large block of data must be read and transferred to the I/O buffer pool in order to extract a single record.

One solution to the above problem is to allocate multiple data sets for Associator and/or Data Storage, each with a different user-defined device type and blocksize. Then the files which may benefit from a specific blocksize can be loaded onto the appropriate device.

Data Storage

A file which is read most frequently in physical sequential order or even in a logical sequence which approaches the physical order can benefit greatly from a larger Data Storage blocksize. More records are then brought into the I/O buffer pool by a single physical read, with a corresponding improvement in buffer efficiency. However, larger blocksizes can be detrimental for files that are subject to frequent random accesses.

Associator

Similarly, a larger Associator blocksize can improve response for a FIND to a file with many ISNs per value. A larger Associator blocksize also benefits those files with frequent READ LOGICAL processing.

Work

Increasing the Work blocksize has proven beneficial for those installations with heavy complex search processing using Work Part 2, significantly reducing the number of I/O required to write and re-read large ISN lists. There is little or no benefit from a large blocksize for Work Part 1 or Part 3.

Cache

Cache controllers can improve Adabas performance when applied in a selective manner. Early models of the cache controllers were not recommended with Adabas. An update to the database caused a write to cache, which then did the physical write to disk asynchronously before returning to Adabas. However, later versions which return immediately after the write to cache have proven worthwhile.

Cache is most effective where the same data is read frequently, or when data which is written to the database is immediately re-read.

The first part of the Associator, containing the FCB and FDT blocks, benefits from cache. These blocks are frequently flushed and replaced in the I/O buffer pool, then soon re-read.

Associator components for files with heavy index searches could benefit from cache if the same index blocks were frequently read. Data components for files which are frequently read could also benefit.

The benefits to Work Part 1 are negligible since this area is normally write-only. Work Part 2 and Part 3 are very good candidates for cache because blocks which are written are soon re-read. The benefits to Work Part 4 are negligible.

Once file components with heavy read activity have been identified, move those components to a string with a cache controller. This maximizes the effectiveness of the cache facility.

Compression

Data compression is an important Adabas feature. Depending upon the options chosen at file load time, data records can be shortened through the reduction of leading zeroes in numeric fields, trailing blanks in alpha fields, and representing consecutive null-value fields together with a single indicator byte.

Data records may also be shortened through the careful use of the FI option for those fields whose values do not contain leading zeroes or trailing blanks. Without this option, the additional compressed length byte can actually result in Adabas data records which are longer than their source records.

The file designer must work with the end user to anticipate the data values which will be stored. Then the physical file can be described to maximize compression. Shorter, more compressed records result in more records per block, fewer physical I/O and improved buffer efficiency.

After the file has been loaded, the file designer should periodically perform a record analysis to verify that the options chosen at file load time are still producing the shortest possible records. It is quite common for the data initially loaded to have different characteristics than expected, resulting in the compression options being less than optimum.

Also, as files are updated, compression options that were effective at initial load may become less effective. In either case, it may be possible to create significant savings in storage space and I/O activity by unloading a file and reloading it with different compression options.

Padding Areas

To minimize the frequency of file reorganization, Adabas provides for a padding area to be reserved in each Associator and Data Storage block when a file is loaded. The amount of padding, expressed as a percent of blocksize, is specified at file load time. The range is 1 per cent to 90 per cent and can be changed using the ADADBS MODFCB function.

Data Storage Padding

Adabas Data Storage records are stored in a compressed format as explained previously. When field values are changed, the updated record may have an increased length. Adabas returns the record to its original block if space is available, otherwise the record must be moved to a block with more room. The purpose of data padding is to accommodate this growth in compressed records as they are updated.

Because of the overhead associated with moving a Data Storage record from one block to another, it is important that the padding areas be sufficient to accommodate the anticipated growth of Data Storage records. A padding area which is too large, however, reduces the amount of data per block at file load time, resulting in additional I/O and wasted space. Alternatively, if records never grow, then any Data Storage padding area is wasted.

A properly set padding percentage, based on actual analysis of the file blocks, strikes the optimum balance to maximize the amount of usable data per block while reducing the frequency of record movement between blocks.

Associator Padding

The values of index (descriptor) fields and the ISNs of Data records which contain those values are stored in Associator index blocks. A padding area reserved at file load time allows the additional ISNs of new or updated records to be inserted within the block. When the padding area has been completely used, the second half of the data in the block is moved to a newly-created index block.

Because of the overhead associated with index block splits, it is important that padding areas be sufficient to accommodate a reasonable amount of expansion in index blocks. A padding area which is too large, however, reduces the number of ISNs which can be stored in a block at load time, resulting in additional I/O and wasted space.

A properly set padding percent, based on actual analysis of the file blocks, strikes the optimum balance to maximize the amount of usable area per block while reducing the frequency of index block splits within the file.

SPMREP Database Analysis Reports

Unicenter CA-SpaceMan File Management for Adabas (Unicenter CA-SpaceMan) includes a batch reporting program that provides complete database space usage information. Use this report set to investigate and define your particular space management problem.

Generating Unicenter CA-SpaceMan Database Analysis Reports

Based on the analysis these reports provide, you can decide which of the three Unicenter CA-SpaceMan options to choose when you go online to create a configuration that solves your problem.

Modifying Report Program Control Cards

To run SPMREP reports, use the sample JCL provided in the XSPMREP member of the Unicenter CA-APAS SOURCE library. The following syntax descriptions for the GLOBALS and REPORT statements are presented here to help you tailor the SPMREP program parameters to meet the needs of your installation. For more information on the uses and analysis of SPMREP reports, see the chapter “Using and Understanding SPMREP Reports.”

GLOBALS Statement

The GLOBALS statement specifies general characteristics of the database, sets default reporting options, and requests maps of the database. The report specifications set in the ASSO, ADRC, and DATA parameters apply to all reports not produced by a REPORT statement. To produce a report on a file or files using different settings than specified in the GLOBALS statement, use one or more REPORT statements.

Syntax

The GLOBALS statement has the following syntax.

```

GLOBALS
  [ ADRC = ( { [ LONG|SHORT|OFF ]
               [ ,UNUSED-ISN-INC = {8|number} ]
               [ ,ZERO-ENTRY-PRINT = {OFF|ON} ] } ) ); ]

  [ ASSO = ( { [ LONG|SHORT|OFF ]
               [ ,UNUSED-BYTE-INC = number ]
               [ ,ZERO-ENTRY-PRINT = {OFF|ON} ] } ) ); ]

  [ CYLINDER-INC = number ]

  [ DATA = ( { [ LONG|SHORT|OFF ]
                [ ,RECS-PER-BLOCK-INC = {10|number} ]
                [ ,RECSIZE-INC = {50|number} ]
                [ ,UNUSED-BYTE-INC = number ]
                [ ,ZERO-ENTRY-PRINT = {OFF|ON} ] } ) ); ]

  DATABASE = database_number
  DEVICE = device_type
  [ EXTSEC = {YES|NO} ]
  [ LOGIO = {YES|NO} ]
  [ LP = block_count ]
  [ LWKP2 = block_count ]
  [ MAP = {YES|NO} ]
  [ SVC = number ]
    
```

Example

```

GLOBALS DA = 9
DE          = 3390
LP          = 500
MAP         = YES
ASSO        = (LONG,UNUSED-BYTE-INC=500,ZERO-ENTRY-PRINT=ON)
ADRC        = (LONG,UNUSED-ISN-INC=10)
DATA        = (LONG,UNUSED-BYTE-INC=50,RECS-PER-BLOCK-INC=12,
               RECSIZE-INC=50)
    
```

GLOBALS Parameters and Options

The various parameters for the GLOBALS statement are explained below.

ADRC

Default report parameters for the Address Converter reports may be specified. These apply to all subsequent REPORT statements unless overridden. See ADRC under REPORT Parameters and Options in this chapter for a description of the valid options.

ASSO

Default report parameters for the Associator reports may be specified. These apply to all subsequent REPORT statements unless overridden. See ASSO under REPORT Parameters and Options in this chapter for a description of the valid options.

CYLINDER-INC

This specifies the size of the report interval in cylinders for the LOGIO report. Only applicable when LOGIO=YES. The default is 10 cylinders.

DATA

Default report parameters for the Data Storage reports may be specified. These apply to all subsequent REPORT statements unless overridden. See DATA under REPORT Parameters and Options in this chapter for a description of the valid options.

DATABASE (Synonym DA)

Must be identical to the DATABASE parameter in ADARUN cards for the database. This database-id number is checked against the internal database-id in the General Control Block. An error is reported if the database-id does not match.

DEVICE (Synonym DE)

Must be identical to the DE parameter in the ADARUN cards for the database. This is the device type of the first Associator data set.

EXTSEC

This parameter is required only if an external security product such as ADAESI or ADASAF is in use.

LOGIO

This parameter requests an analysis of LOGIO from an Adabas Command Log by volume by cylinder. The default is NO.

LP

Must be identical to the LP parameter in the ADARUN cards for the MPM running against the database. This parameter is used in computing the size of Part 1 of the Work data set. This parameter is not required and defaults to the LP value of the active MPM session, or to the LP value of the previous session if the MPM is not active.

LWKP2

If your installation uses the LWKP2 parameter to size Work Part 2, the value should be specified with this parameter. If omitted, Unicenter CA-SpaceMan computes the required size of Work Part 2.

MAP

This parameter is used to request a map of all components of the database by volume. The default is NO.

SVC

This parameter must be set to the identical value as the SVC parameter in the ADARUN cards. The parameter is required only if the SVC has been linked with ADAESI or ADASAF.

REPORT Statement

A REPORT statement requests reports for a file or range of files. To request different report options for different files, you can specify as many REPORT statements as needed.

Syntax

The REPORT statement has the following syntax.

```
REPORT
FILE = ( file [, file, ...] [, file THRU file]... )
[ ADRC = ( { [ LONG|SHORT|OFF ]
             [ ,UNUSED-ISN-INC = number ]
             [ ,ZERO-ENTRY-PRINT = {OFF|ON} ] } ) ); ]

[ ASSO = ( { [ LONG|SHORT|OFF ]
             [ ,UNUSED-BYTE-INC = number ]
             [ ,ZERO-ENTRY-PRINT = {OFF|ON} ] } ) ); ]

[ DATA = ( { [ LONG|SHORT|OFF ]
             [ ,RECS-PER-BLOCK = number ]
             [ ,RECSIZE-INC = number ]
             [ ,UNUSED-BYTE-INC = number ]
             [ ,ZERO-ENTRY-PRINT = {OFF|ON} ] } ) ); ]
```

Example

```
REPORT FILE = (1 THRU 10, 71, 75, 100 THRU 125)
ADRC = (LONG, ZERO-ENTRY-PRINT=ON, UNUSED-ISN-INC=25)
ASSO = (SHORT)
DATA = (OFF)
```

REPORT Parameters and Options

The parameters and options of the REPORT statement are explained below.

FILE

Use the FILE parameter to specify the file(s) for which you want to generate reports. The list of files is separated by commas. A range of files may be specified with the THRU keyword.

Address Converter Reports

ADRC

The ADRC parameter specifies the options you want for the Address Converter reports. Address Converter reports are printed only if you include an ADRC specification in the GLOBALS or the REPORT statement. This parameter overrides any ADRC specification in the GLOBALS statement.

Valid report options for the Address Converter reports include:

SHORT | LONG | OFF

SHORT requests only the Address Converter report, although if ISN REUSAGE is YES, the List of Current Reusable ISNs report is also produced.

LONG requests all Address Converter reports. This results in reading all Address Converter blocks for the file.

OFF, the default value, requests no Address Converter reports. This overrides a GLOBALS specification.

UNUSED-ISN-INC = *number*

This option specifies the range increment for the Summary of Unused ISNs Per Block report. The default value is 8 and the synonym UII can be substituted for UNUSED-ISN-INC.

ZERO-ENTRY-PRINT = {OFF|ON}

This option specifies whether ranges of unused ISNs with zero blocks are to be included in the Summary of Unused ISNs Per Block report. The default value is OFF and the synonym ZP can be substituted for ZERO-ENTRY-PRINT.

Associator Reports

ASSO

The ASSO parameter specifies the options you want for the Associator report. Associator reports are printed only if you include an ASSO specification in the GLOBALS or the REPORT statement. This parameter overrides any ASSO specification in the GLOBALS statement.

Valid report options for Associator reports include:

SHORT | LONG | OFF

SHORT requests only the Associator Summary of Extents report.

LONG requests all Associator reports. This results in reading all used Associator blocks for the file.

OFF, the default value, requests no Associator reports. This overrides a GLOBALS specification.

UNUSED-BYTE-INC = *number*

This option specifies the range increment for the Summary of Unused Bytes report. The default is one-fourth the number of bytes in the Associator block padding area. The synonym UBI can be substituted for UNUSED-BYTE-INC.

ZERO-ENTRY-PRINT = {OFF | ON}

This option specifies whether ranges of unused bytes with zero blocks are to be included in the Summary of Unused Bytes report. The default value is OFF and the synonym ZP can be substituted for ZERO-ENTRY-PRINT.

Data Storage Reports

DATA

The DATA parameter specifies the options you want for the Data Storage report. Data Storage reports are printed only if you include a DATA specification in the GLOBALS or the REPORT statement. This parameter overrides any DATA specification in the GLOBALS statement.

Valid report options for the Data Storage reports are:

SHORT | LONG | OFF

SHORT requests only the Data Storage Summary of Extents report.

LONG requests all Data Storage reports. This results in reading all Data Storage blocks for the file.

OFF, the default value, requests no Data Storage reports. This overrides a GLOBALS specification. The default value is OFF.

RECS-PER-BLOCK-INC = *number*

This option specifies the range increment for the Summary of Records Per Block report. The default value is 10 records and the synonym RBI can be substituted for RECS-PER-BLOCK-INC.

RECSIZE-INC = *number*

This option specifies the range increment for the Summary of Compressed Record Sizes report. The default value is 50 bytes and the synonym RSI can be substituted for RECSIZE-INC.

UNUSED-BYTE-INC = *number*

This option specifies the range increment for the Summary of Unused Bytes report. The default is one-fourth the number of bytes in the data block padding area. The synonym UBI can be substituted for UNUSED-BYE-INC.

ZERO-ENTRY-PRINT = {OFF|ON}

This option specifies whether ranges with zero entries are to be included in the Data Storage Summary reports. The default value is OFF and the synonym ZP can be substituted for ZERO-POINT-ENTRY.

Using and Understanding SPMREP Reports

This chapter discusses specific space management problems, linking each identified problem to a particular SPMREP batch Unicenter CA-SpaceMan File Management for Adabas (Unicenter CA-SpaceMan) report.

Typical Adabas Space Management Problems

As each report is introduced, the manual presents the options that should be added to the REPORT statement of the SPMREP JCL to produce a given report. Once you have analyzed and defined your space management problem, you can select a Unicenter CA-SpaceMan solution option, for example, reorganize, reallocate, or file reload/add, then go online to create the necessary configuration and generate utility control cards from it. The batch reporting and online tasks are explained more completely in the chapters “SPMREP Database Analysis Reports” and “Getting Started Using Online Unicenter CA-SpaceMan”, respectively.

Data Storage Analysis

Inefficient Data Storage Padding Factors

The Data Storage padding area exists so that a compressed record may be updated with additional data but still be stored back into the same block from which it was read. If the updated record does not fit, additional internal overhead is required to find a block with sufficient space, move the record from the old block to the new block, update the block lengths, update the corresponding DSST bytes, and adjust the pointer in the Address Converter.

Setting the padding percent value at file load time is often just a guess. The Summary of Unused Bytes report (SPM102 reports) analyzes unused space by block, showing how many bytes are unused before and within the padding area.

If records have fully expanded and significant padding area remains, it is possible to decrease the padding percentage, resulting in more records per block and improved performance through fewer I/Os. If the padding area has been exhausted but further record expansion is expected, an increase in the padding percentage reduces the number of record migrations. You can change Data Storage padding using the ADADBS MODFCB function. The new padding value does not change existing records, but is considered whenever records are added, deleted, or modified.

Report SPM102 is produced with either the “short” or “long” report option, as specified in the program control cards for the SPMREP job. The “short” form reads the Data Storage Space Table (DSST) blocks used by this file and displays the distribution of unused space by block. This form of the report shows unused bytes before the padding area only.

The “long” form of the report reads through all blocks of Data Storage up to the first unused block. The number of unused bytes in each block is noted. Each block is counted in a table which is incremented according to the UNUSED-BYTE-INC (UBI) value of the Data Storage GLOBALS or REPORT statement. If omitted, the UBI value defaults to one-fourth the number of padding bytes shown on report SPM101.

Report SPM102 shows the number of blocks which fall into each increment of unused bytes, for example, in the sample report below, 15 blocks contain between 300 and 399 unused bytes per block. The “unused bytes” are the bytes between the last record and the end of the block.

2002-11-06

COMPUTER ASSOCIATES INTERNATIONAL, INC.
DATA STORAGE ANALYSIS REPORT

18:37:43

REPORT SPM102 : SUMMARY OF UNUSED BYTES

DATABASE NAME : DATABASE-71
DATABASE NUMBER : 00071
DATABASE VERSION: 712

FILE NAME : EMPLOYEES
FILE NUMBER : 00001

NUMBER OF BLOCKS	UNUSED BYTES
4	0 - 99
15	100 - 199
9	200 - 299

15	300 - 399
2	400 - 499
1	3,700 - 3,799

46 TOTAL BLOCKS
316 AVERAGE UNUSED BYTES PER BLOCK
18.0% OF THE PADDING AREA IS USED

If you set ZERO-ENTRY-PRINT (ZP)=ON, a line is printed for each increment regardless of the number of blocks which are counted. With ZP=OFF, those increments with a zero counter are not printed, resulting in a shorter print report.

The dashed line on the report represents the padding line. Setting the UBI value equal to the number of padding bytes shown on report SPM101, or a fraction thereof, clearly shows the number of blocks that have expanded into the padding area, which appear above the dashed line, and the number of blocks that have not. Using this method, the amount of expansion into the padding area can also be seen.

Inefficient Data
Compression Options

One of the major benefits of Adabas is its ability to compress data, resulting in more records per block and fewer I/O. The ADACMP utility tells you about compression of the original set of records used to load a file. However, it is common to load a file initially with no records at all. In this case, you must set the compression options based on your estimation of what the data might look like, which may differ substantially from the data that is actually loaded.

The Summary of Compressed Record Sizes report (SPM104) displays the effect of the selected compression options on the real data once it has been loaded. Report SPM104 is produced with the long report option. As the used Data Storage blocks are read, the compressed length of each record is noted. Each record is counted in a table which is incremented according to the RECSIZE-INC (RSI) value of the Data Storage GLOBALS or REPORT statement.

The report shows the number of records which fall into each increment of compressed record length. In the sample report below, for example, 422 records are between 200 and 299 bytes long.

2002-11-06	COMPUTER ASSOCIATES INTERNATIONAL, INC. DATA STORAGE ANALYSIS REPORT	18:37:43
REPORT SPM104 : SUMMARY OF COMPRESSED RECORD SIZES		
DATABASE NAME : DATABASE-71		
DATABASE NUMBER : 00071		
DATABASE VERSION: 712		
FILE NAME : EMPLOYEES		
FILE NUMBER : 00001		
NUMBER OF RECORDS	COMPRESSED RECORD SIZE	
-----	-----	
1	0 - 99	
684	100 - 199	
422	200 - 299	
1,107 TOTAL RECORDS		
188 AVERAGE BYTES PER COMPRESSED RECORD		

If you set ZERO-ENTRY-PRINT (ZP)=ON a line is printed for each increment regardless of the number of records which are counted. When you set ZP=OFF, those increments with a zero counter are not printed, resulting in a shorter print report.

This report shows you the actual results of Adabas compression. It may be used to substantiate the space savings attained by changing field compression options.

Better compression means shorter records, which allows more records to fit into a fixed-length Data Storage block. This result can be seen on the Summary of Records per Block report (SPM103). With more records per block, the file requires less DASD space, and fewer physical I/O are needed to read the data. Better compression also increases the possibility of a record being in the I/O buffer pool, resulting in greater buffer efficiency.

To improve compression, you must unload and reload a file with new compression options.

Report SPM103 is produced with the long report option. As the used Data Storage blocks are read, the number of records in each block is computed. Each block is counted in a table which is incremented according to the RECS-PER-BLOCK-INC (RBI) value of the Data Storage GLOBALS or REPORT statement.

A warning message is issued when any file component reaches three or more extents. The file analysis job completes normally, but with a return code of four.

Index Analysis

The use, or misuse, of descriptors is one of the most significant factors in the performance of an Adabas application. Response time for an add, update, or delete command is directly proportional to the number of descriptors in the file because each descriptor is located in a different set of Upper and Normal Index blocks. The file designer is often led to believe that many of the fields in the file must be defined as descriptors, because there is a need to access records based on those fields.

The Summary of Descriptor Allocations report (SPM202) lists all of the fields in the file which have been defined as descriptors. Any fields which have no values should be investigated. This report also shows the number of blocks of DASD space that can be saved for each descriptor released with the Adabas utility RELEASE function.

Report SPM202 is produced with the long report option. The index structure is read top-down beginning at the RABN of the highest index level and proceeds to the last used normal index block of the last descriptor. Each descriptor is listed and an indicator (UQ) is added if the field is defined unique in the Field Description Table (FDT).

The number of distinct values is shown as well as the number of ISNs (records) that have a value for that descriptor. Descriptors with only a few values or records should be investigated. A descriptor with only two values, for example, and a number of ISNs near the total size of the file should be evaluated for its effectiveness as a search key. The number of Upper and Normal Index blocks occupied by each descriptor is reported, which helps you to measure the space impact of any descriptor which is added or deleted.

A new enhancement in Adabas V7 is the Forward Index Compression feature that removes redundant prefix information from index values. The FIC column on the SPM202 report reports the percent of compression for each descriptor in the file. The value represents the space reduction that occurs when Adabas removes the repeating portion of a descriptor value. Index compression is in effect when a file is loaded with INDEXCOMPRESSION=YES.

2002-11-06

COMPUTER ASSOCIATES INTERNATIONAL, INC.
ASSOCIATOR ANALYSIS REPORT

18:37:43

REPORT SPM202 : SUMMARY OF DESCRIPTOR ALLOCATIONS

DATABASE NAME : DATABASE-71
DATABASE NUMBER : 00071
DATABASE VERSION: 712

FILE NAME : EMPLOYEES
FILE NUMBER : 00001

DESCRIPTOR	VALUES	ISNS	MINIMUM ISNS/VALUE	AVERAGE ISNS/VALUE	MAXIMUM ISNS/VALUE	AVERAGE DE LENGTH	UI BLOCKS	NI BLOCKS	FIC
AA (UQ)	1,107	1,107	1	1	1	9	1	5	62%
AE	804	1,107	1	1	19	8	1	5	29%
AH	922	1,107	1	1	43	5	1	5	45%
AJ	309	1,107	1	3	141	11	1	4	19%
AO	168	1,107	1	6	98	7	1	3	63%
AP	237	1,106	1	4	75	16	1	4	32%
AZ	21	1,812	1	86	843	4	1	3	7%
H1	259	1,081	1	4	93	5	1	3	51%
PH	604	1,107	1	1	33	4	1	4	23%
S1	13	1,107	1	85	206	5	1	2	15%
S2	1,051	1,107	1	1	5	14	1	6	45%
S3	1,567	3,383	1	2	25	14	1	10	72%

Descriptors With Too Few Values

An effective Adabas descriptor is one whose values divide the file into small, related sets of records which make up no more than 15 per cent of the total file. A FIND on a descriptor value which returns more than 15 per cent of the file may actually be less efficient than doing a physical READ of all data records in the file and selecting only those records with the desired value. This is because the number of I/O needed to read all of Data Storage for this file is less than the I/O required to read the Associator index, Address Converter, and Data Storage for the desired subset of records.

The Summary of Descriptor Allocations report (SPM202) see above, identifies any descriptor which has only a few values, but is associated with a large number of records equal or nearly equal to the total number of records in the file. These fields should be investigated for their effectiveness as descriptors.

Also to be investigated are fields that are almost unique, that is, when the number of distinct values is close to the number of ISNs. This could be an indication of erroneous or duplicate values. Ineffective descriptors should be released and application logic adjusted accordingly, resulting in fewer I/Os, less CPU processing and better response time. Report SPM202 shows the number of blocks of DASD space saved for each descriptor released with the Adabas ADADBS RELEASE utility.

Inefficient Associator
Padding Factors

The Associator padding area exists so that there is room to insert ISNs of new or updated records into a block. A great deal of overhead occurs when an ISN cannot be placed in the block where it belongs due to insufficient padding space. A copy of the original block is written to the Work data set for recovery purposes. The block is split 50/50, with half of the ISNs remaining in the original block and half written to the next available empty or reusable index block. This alteration in the Normal Index causes a change in the Upper Index and possible additional block splits as the change ripples up through the index chain.

Setting the padding percentage at file load time is often just a guess. The Summary of Unused Bytes report (SPM203) analyzes unused space by block, showing how many bytes are unused before and within the padding area. The extent to which index block splits have occurred is quickly apparent and a projection of future splits can be made. Adjustments to the padding percentage should be made to reduce the I/O and CPU overhead of index block splits.

Associator padding may be changed by unloading and reloading the file or by the REORASSO function of the ADAORD utility. The Associator padding can also be changed with the ADADBS MODFCB function. The new padding value does not change existing index elements, but is considered whenever elements are added, deleted, or modified.

Report SPM203 is produced with the long report option. Each block of the index structure is read, from the RABN of the highest index level to the last block used. The number of unused bytes in each block is noted. Each block is counted in a table which is incremented according to the UNUSED-BYTE-INC (UBI) value of the Associator GLOBALS or REPORT statement. If omitted, the UBI value defaults to one-fourth the number of padding bytes shown on report SPM201.

Unavailable Index
Blocks

Files with a high amount of activity in which old records are deleted and new records are added will very quickly use their allocated amount of Normal and Upper Index space and require additional index extents.

A chain of RABNs is maintained for the upper index and normal index blocks that are available for reuse. For example, in the sample SPM204 report below, RABN 1,398 is the LAST-UNUSED-NI-RABN to become available for reuse and is the next RABN selected when an additional block of normal index is needed.

2002-11-06	COMPUTER ASSOCIATES INTERNATIONAL, INC. ASSOCIATOR ANALYSIS REPORT	18:37:43
REPORT SPM204 : SUMMARY OF USED BLOCKS		
DATABASE NAME : DATABASE-71		
DATABASE NUMBER : 00071		
DATABASE VERSION: 712		
FILE NAME : EMPLOYEES		
FILE NUMBER : 00001		
INDEX BLOCK TYPE	BLOCKS USED	BLOCKS REUSEABLE
-----	-----	-----
U15	0	0
U14	0	0
U13	0	0
U12	0	0
U11	0	0
U10	0	0
U9	0	0
U8	0	0
U7	0	0
U6	0	0
U5	0	0
U4	0	0
U3	1	0
MI	12	0
NI	65	2
LAST-UNUSED-UI-RABNS :		0
LAST-UNUSED-NI-RABNS :		1,398

Upper Index Too Big The amount of UISIZE allocated at file load time is often much larger than necessary. For example, nearly 83,000 records can be addressed from a single block of main index for a 9 byte descriptor value stored on a 3380 Associator device.

Use the Unicenter CA-SpaceMan Summary of Extents report (SPM201) to quickly see how much Upper Index space is used/unused and adjust the UISIZE at the next file UNLOAD/RELOAD or REORDER ASSOCIATOR. If there are additional extents of upper and normal index which are largely unused, adjust these with the MAXUI and MAXNI parameters. Report SPM201 is produced with either the short or long report option. The File Control Block (FCB) for each file is read to obtain the information on extents allocated to the Upper and Normal indices.

On the SPM201 report, as shown below, note the highest index. The value ranges from a minimum of 3 to a maximum of 15, which indicates that the file has reached the highest possible upper index level.

The RABN range and number of blocks allocated are shown, along with the number of blocks and percentage of blocks actually used. This information should be used to monitor the growth of a file's index areas. The padding percentage is converted to a number of bytes in the padding area based on device type and blocksize. The VOLSER displayed is the pack on which the extent begins. This information is obtained from the Adabas volume serial table.

A warning message is issued when any file component reaches three or more extents. The file analysis job completes normally, but with a return code of 4.

2002-11-06		COMPUTER ASSOCIATES INTERNATIONAL, INC.				18:37:43			
ASSOCIATOR ANALYSIS REPORT									
REPORT SPM201 : SUMMARY OF EXTENTS									
DATABASE NAME : DATABASE-71									
DATABASE NUMBER : 00071									
DATABASE VERSION: 712									
FILE NAME : EMPLOYEES									
FILE NUMBER : 00001									
PADDING FACTOR : 010%									
HIGHEST INDEX : 03									
ISNSIZE : 4									
UI	FIRST	LAST	BLOCKS	BLOCKS	BLOCK	PAD	DEVICE	VOL	
EXT	BLOCK	BLOCK	ALLOC	USED	SIZE	BYTES	TYPE	SER	
1	1,532	1,544	15	86.6%	2,544	254	3390	GSW003	
	TOTAL	ALLOC	15						
	TOTAL	USED	13						
NI	FIRST	LAST	BLOCKS	BLOCKS	BLOCK	PAD	DEVICE	VOL	
EXT	BLOCK	BLOCK	ALLOC	USED	SIZE	BYTES	TYPE	SER	
1	1,395	1,531	137	48.9%	2,544	254	3390	GSW003	
	TOTAL	ALLOC	137						
	TOTAL	USED	67						

Address Converter Analysis

Incorrect ISNREUSE
Option

The ISNREUSE option is an effective means to delay or prevent additional Address Converter extents, but can also be a hindrance to performance when it is used incorrectly. Additional Address Converter I/O is required to search for a reusable ISN, only to discover that no such ISN is available. Setting this option should be based on current information about the availability of ISNs for reuse.

The Summary of Unused ISNs per Block report (SPM302) provides this information. Report SPM302 is produced with the long report option. The Address Converter is read from the first block to the highest block used. The number of unused ISNs in each block is noted. Each block is counted in a table which is incremented according to the UNUSED-ISN-INC (UII) value of the Address Converter GLOBALS or REPORT statement. If omitted, the UII value defaults to 8, the size of the reusable ISN list maintained in the File Control Block.

```

2002-11-06          COMPUTER ASSOCIATES INTERNATIONAL, INC.          18:37:43
                   ADDRESS CONVERTER ANALYSIS REPORT

REPORT SPM302      : SUMMARY OF UNUSED ISNS PER BLOCK

DATABASE NAME      : DATABASE-71
DATABASE NUMBER    : 00071
DATABASE VERSION   : 712

FILE NAME          : EMPLOYEES
FILE NUMBER        : 00001

      NUMBER      UNUSED ISNS
      OF          PER
      BLOCKS     BLOCK
-----
          1         0 - 7
          1        160 - 167

      2 TOTAL BLOCKS
      82 AVERAGE UNUSED ISNS PER BLOCK
    
```

The report shows the number of blocks which fall into each increment of unused ISNs. In the sample report above, for example, 1 block has between 0 and 7 unused ISNs per block.

If you set ZERO-ENTRY-PRINT (ZP)=ON, a line is printed for each increment regardless of the number of blocks which are counted. When you set ZP=OFF, those increments with a zero value are not printed, resulting in a shorter print report.

This report should be used to determine the availability of ISNs for ISNREUSE. If the only blocks which contain unused ISNs are at the end of the Address Converter, ISNREUSE=YES only causes additional I/O to the Associator without finding any ISNs available for reuse.

Set ISNREUSE=YES when TOP ISN approaches MAX ISN and report SPM302 shows sufficient available ISNs for the expected number of records which are added to the file.

If ISNREUSE=YES, a List of Current Reusable ISNs report (SPM303) are produced to show which ISNs are available and to determine which part of the Address Converter is being searched for available ISNs. The ISNREUSE option can be changed with the ADADBS utility.

Report SPM303 is produced with either the short or long report option, but only if ISN REUSAGE is YES on Report SPM301. This report lists the table of eight ISNs which are currently available for reuse and displays the ISN where the search for reusable ISNs resumes when the table must be refilled. Each time Adabas adds a record to the file and assigns one of these ISNs from the table, the position in the table is set to zeros.

If this report consistently shows all zeros, there may be no ISNs available to reuse. Examine Report SPM302 for the availability of any unused ISNs.

Adabas provides an option to set ISNREUSE=YES at load time. You should consider the impact on performance this option has until enough record DELETES have been done to make ISNREUSE effective. Use Unicenter CA-SpaceMan reports to determine when the ISNREUSE option should be YES or NO.

Large, Lightly Used
Secondary Extents

When a logical extent (Data, Address Converter or Index) of a large file is exhausted, Adabas allocates another extent of approximately 25 per cent the size of the original extent. The immediate result is a large area of disk storage which contains little information. The remainder of the space is wasted if the file is fully loaded or experiences very little additional growth.

Use the Unicenter CA-SpaceMan Summary of Extents report (SPM301) to quickly see how much of each extent is used/unused. Secondary extents may be eliminated by unloading and reloading the file or by reordering the Associator for the file using Adabas utilities.

Report SPM301 is produced with either the short or long report option. The File Control Block (FCB) for each file is read to obtain the information on extents allocated to the Address Converter. Note the TOP ISN and MAX ISN values.

When the TOP ISN reaches MAX ISN, another Address Converter extent is allocated and the MAX ISN is automatically increased, unless NOACEXTENSION has been specified. The RABN range and number of blocks allocated are shown, along with the number of blocks and percentage of blocks actually used. This information should be used to monitor the growth of a file's Address Converter.

The number of ISNs per block is shown for each device type and blocksize. The VOLSER displayed is the pack on which the extent begins. This information is obtained from the Adabas volume serial table. A warning message is issued when any file component reaches three or more extents. The file analysis job completes normally, but with a return code of 4.

Work Analysis

Work Part 1 Too Big

The size of Work Part 1 is determined by the ADARUN LP parameter. Adabas writes to this area sequentially and wraps around when the last block has been reached. Many installations have set this value so high that each LP block is used only once or twice a day. Blocks become available for reuse after the ET has occurred for an update transaction and an I/O buffer flush has taken place. It is best to strive for short logical transactions and multiple wrap-arounds of the LP area.

An over-allocated Work Part 1 results in:

- Excessive head movement within the Work data set
- Wasted DASD space
- Longer time required for a restart/recovery
- Increased likelihood of a User Queue overflow (RSP072) during restart/recovery

The size of Work Part 1 is shown on the Database by Pack Report (SPM401). Unicenter CA-SpaceMan requests can be submitted online or in batch to report the number of I/O to Work Part 1. This assists in setting the optimum value for the LP parameter. Report SPM401 is produced when MAP=YES is coded on the Unicenter CA-SpaceMan GLOBALS statement. The General Control Block (GCB), all File Control Blocks (FCBs), and the Free Space Table (FST) of a selected database are read to produce a map of the physical Adabas database.

The map is displayed in sequence by volume serial. All file components and free spaces are mapped by their from/to RABNs and physical location on the pack. File components and free spaces which span multiple packs are marked as (PARTIAL) components.

The LP value supplied on the Unicenter CA-SpaceMan GLOBALS statement is used in computing the size of Work Part 1. It should be equal to the LP value in the ADARUN cards for this database. The LP may be omitted since its value is obtained directly from the work header.

2002-11-06

COMPUTER ASSOCIATES INTERNATIONAL, INC.
PHYSICAL DATABASE MAPPING UTILITY

17:37:43

REPORT SPM401 : DATABASE BY PACK

DATABASE NAME : DATABASE-71
DATABASE NUMBER : 00071
DATABASE VERSION: 712

VOLUME SERIAL : GSW003 (CONT.)

I	-----	I
I	2,396 - 2,415 FILE 00015 EXTENT 1 UPPER INDEX	I
I	-----	I
I	2,416 - 2,437 FILE 00018 EXTENT 1 ADDRESS CONVERTOR	I
I	-----	I
I	2,438 - 2,840 FILE 00018 EXTENT 1 NORMAL INDEX	I
I	-----	I
I	2,841 - 2,862 FILE 00018 EXTENT 1 UPPER INDEX	I
I	-----	I
I	2,863 - 2,866 FILE 00019 EXTENT 1 ADDRESS CONVERTOR	I
I	-----	I
I	2,867 - 2,966 FILE 00019 EXTENT 1 NORMAL INDEX	I
I	-----	I
I	2,967 - 3,266 FILE 00019 EXTENT 1 UPPER INDEX	I
I	-----	I
I	3,267 - 3,269 FILE 00213 EXTENT 1 ADDRESS CONVERTOR	I
I	-----	I
I	3,270 - 3,293 FILE 00213 EXTENT 1 NORMAL INDEX	I
I	-----	I
I	3,294 - 3,308 FILE 00213 EXTENT 1 UPPER INDEX	I
I	-----	I
I	3,309 - 3,312 FILE 00246 EXTENT 1 ADDRESS CONVERTOR	I
I	-----	I
I	3,313 - 3,351 FILE 00246 EXTENT 1 NORMAL INDEX	I
I	-----	I
I	3,352 - 3,375 FILE 00246 EXTENT 1 UPPER INDEX	I
I	-----	I
I	3,376 - 3,379 FILE 00247 EXTENT 1 ADDRESS CONVERTOR	I
I	-----	I
I	3,380 - 3,419 FILE 00247 EXTENT 1 NORMAL INDEX	I
I	-----	I
I	3,420 - 3,444 FILE 00247 EXTENT 1 UPPER INDEX	I
I	-----	I
I	3,445 - 3,446 FILE 00302 EXTENT 1 ADDRESS CONVERTOR	I
I	-----	I
I	3,447 - 3,454 FILE 00302 EXTENT 1 NORMAL INDEX	I
I	-----	I
I	3,455 - 3,457 FILE 00302 EXTENT 1 UPPER INDEX	I
I	-----	I
I	3,458 - 3,530 FILE 00004 EXTENT 1 NORMAL INDEX	I
I	-----	I
I	3,531 - 3,545 FILE 00001 EXTENT 1 UPPER INDEX	I
I	-----	I
I	3,546 - 3,588 FILE 00005 EXTENT 1 NORMAL INDEX	I
I	-----	I

Work Part 2 Too Big Or
Too Small

Use the SPM401 report to view the entire database layout, noting concentrations of files or unused space on a pack. The breakdown of the Work data set helps in allocating and managing its separate parts.

The default size of Work Part 2 is determined at Adabas nucleus start-up by a computation based on the highest ISN in use in the database. This value may be overridden by ADARUN parameter LWKP2.

The allocated size of Work Part 2 must be at least twice the highest number of blocks required from the area by any one command, due to the check for space made by Adabas before beginning a complex search. The default computation often results in under-allocation of Work Part 2, so that complex searches appear to be single threaded due to the lack of available blocks. On the other hand, if complex searches have been minimized or eliminated at your installation, even the default computation may result in an over-allocation of space.

The size of Work Part 2 is shown on the Database by Pack Report (SPM401). Unicenter CA-SpaceMan requests can be submitted online or in batch to report the number of I/Os to Work Part 2. Use this information to determine whether LWKP2 parameter is necessary, and what its value should be.

If your installation uses the LWKP2 parameter to set the size of Work Part 2, this value must also be entered as LZ on the Unicenter CA-SpaceMan GLOBALS statement. With this parameter omitted, Unicenter CA-SpaceMan reads all the FCBs to compute the size of Work Part 2, just as Adabas does, based on the highest ISN in the database. Note that the computation is different for Adabas V6, which assumes ISNSIZE=4.

The proper size for Work Part 2 results in greater throughput of complex searches and decreased head travel between the three parts of work.

Work Part 3 Too Big

Work Part 3 is usually over-allocated simply because it is the remainder of the Work data set. Much of the space is unused throughout the Adabas session.

The Database by Pack report (SPM401) shows the size of each Work part. A SUMMARIZE request by work component displays the number of I/O to each part as well as which blocks, or ranges of blocks, are actually being used. A more efficiently allocated Work data set results in less DASD used and less head travel on the volume.

Disk Device Analysis

Imbalanced I/O Loads Across Volumes

When a high percentage of Adabas I/O activity occurs on just a few of the physical volumes, Adabas processing may be delayed to wait for I/O processing. If this same I/O load were spread evenly across multiple volumes, processing times could be reduced.

Use a Unicenter CA-SpaceMan request to SUMMARIZE I/O counts by volume and identify those devices which may be causing delayed throughput. Then use Unicenter CA-SpaceMan historical data of I/O activity by file components to reconfigure your database so that all available volumes on which Adabas resides have an equal I/O load.

Excessive Head Movement

The number of cylinders which the disk read/write heads must move between successive I/Os on the same volume is a factor in the total time required for the I/O. Adabas file components with heavy I/O can be placed in such a way that activity is concentrated around one portion of the disk.

Use the I/O by Cylinder report (SPM402) or a Unicenter CA-SpaceMan request to identify those file components and physical cylinders which are most frequently accessed. The Unicenter CA-SpaceMan IOLOG facility can be used to report the amount of head movement between successive I/O.

Report SPM402 is produced when LOGIO=YES is coded on the Unicenter CA-SpaceMan GLOBALS statement. A map is generated similar to report SPM401, sorted by volume serial, and divided into cylinder increments. The size of the increments is set by the CYLINDER-INC parameter of the Unicenter CA-SpaceMan GLOBALS statement. Each increment is displayed by its from-to RABNs.

The Adabas Command Log is read from DD statement DDLOG. This file is required only when LOGIO=YES is coded on the GLOBALS statement. Not all physical I/Os for each command appear on the Command Log due to record length restrictions.

For those periods of time when Adabas I/O logging is turned ON, the component and RABN is used to accumulate READ and WRITE counts. The total READS and WRITES are then reported for each cylinder increment. Use this report to identify areas of the physical pack which contain high levels of Adabas I/O activity.

Do not try to tune head movement too finely. Each head movement consists of three time intervals: acceleration of the head, travel, and braking. For movements of relatively few cylinders the acceleration and braking times are greater than the travel time. Therefore, little improvement is gained by reducing head movement by only a few cylinders. Address situations where the head travels across many cylinders.

Allocating and
Placing Adabas Data
Sets

Allocating Adabas data sets across multiple volumes and mixing components on the same volume has always been a time-consuming and error-prone task.

OS JCL limits the allocation of Adabas data sets to one size for the primary extent and one size for all the subsequent secondary extents. Dummy data sets must be allocated in order to give the appearance of a full volume, thus forcing secondary extents onto the desired volumes.

With the Unicenter CA-SpaceMan OS Allocator, data sets may have extents of different sizes on different volumes. This makes I/O load balancing possible across volumes. When supplied with the data set name, volume, and space requirements, a single execution of the Unicenter CA-SpaceMan OS Allocator allocates and catalogs all the Adabas data sets into any desired configuration.

Before Getting Started Online

Before you use the online components of Unicenter CA-SpaceMan File Management for Adabas (Unicenter CA-SpaceMan), you must first gather database layout and I/O load data for Unicenter CA-SpaceMan to use. You place this data into the Unicenter CA-SpaceMan Repository file for use by the various Unicenter CA-SpaceMan functions. It is as important to gather accurate and appropriate I/O samples and to gather current figures for the Repository file if it is to properly perform the online steps.

The Unicenter CA-SpaceMan Repository

Unicenter CA-SpaceMan maintains all of its information on databases in an Adabas file, the Unicenter CA-SpaceMan Repository. The Repository file is the primary source from which the program creates a database image or projection for you to modify. This file contains records for the following types of data:

- Existing databases and data set names
- Existing files, extent allocations and load parameters
- DASD volumes
- Owner data for accounting reports
- I/O sample data for existing databases
- History data for Adabas file sizes
- Configurations for possible reorganization of the database

Purposes of the Repository

The Unicenter CA-SpaceMan Repository file data serves a number of functions:

- Displaying the layout of existing databases
- Maintaining a history of space usage by file
- Storing of I/O load samples showing which parts of the database were most frequently accessed during an interval
- Storing input parameters (projections) for the Unicenter CA-SpaceMan database Configurator
- Storing output database configurations from the Unicenter CA-SpaceMan Configurator. These are used to generate control cards for various types of database maintenance

Types of Data in the Repository

In order to support the above functions, the following types of data are maintained in the Repository file:

- Current database, which contains information about existing databases
- File history that contains historical data on file sizes
- Ownership contains data on who owns files and Natural logons for charge-back
- I/O samples, that is, I/O at the file and component level for a period of time
- Projections, that is, input to the Unicenter CA-SpaceMan Configurator program, Current Database information modified by the DBA to show expected file growth and other projected changes
- Configurations, that is, output from the Configurator program showing the layout of the database optimized for a given I/O load and set of volumes

Record Types

A set of records is kept for each of the above types of data. These records can be of the following types:

- Database record: One record for general database information.
- Data set record: The data set names for the database OS files.
- File record: One record for each file on the database.
- Component record: One record for each component of each file, at least four per file.
- Volume record: The volumes on which the database is or will be stored.
- Index record: Various records describing owners, projections, configurations, etc.

The diagram below shows the record types that may be present for each type of data.

	DB	DS	FI	CM	VO	Notes:
CURRENT DB	x	x	x	x	x	
IO SAMPLE HISTORY	x	x	x	x	x	
PROJECTION (1)	x	x	x		x	Reconfiguration option
PROJECTION (2)					x	Reallocation option
PROJECTION (3)			x			File Add/Reload
CONFIGURATION	x	x	x	x	x	

Repository File Sources

Information for maintaining the Repository file comes from several sources:

- Database component extent information is read directly from the Adabas Associator by the SPMUPD program. You must add this data to the Repository file using the SPMDIRUP Natural program.
- The Data Collector collects I/O sample information from the I/O Log generated by DBGIOR5.
- Volume information is user-specified. Information on I/O loads to non-Adabas files can be obtained from RMF statistics or other performance monitoring tools.
- File history is maintained automatically as extent data is added to the database. You specify the amount of detail maintained.
- The Unicenter CA-SpaceMan online Configurator program generates configurations. Configurations are possible layouts for the database given a set of volumes with available space.
- Accounting data is entered at the terminal by the user.

The Space Management Process

The space management process as defined by Unicenter CA-SpaceMan can be put simply into a few separate steps. At nearly every point along the way, however, you have the opportunity to view and sometimes modify the various database records available. In this section we first define the steps in the space management process, then explain in detail how to perform each step.

The general steps in the space management process are described below:

1. Add Current Database information to the Repository.

Use the SPMDIRUP job to add data to the Repository from the database information collected by the SPMUPD job. We expect that you will run this update regularly, perhaps daily, but you must update the Current Database in the Repository before using Unicenter CA-SpaceMan after any changes have been made to the database. Running Unicenter CA-SpaceMan with inaccurate database space information can lead to unpredictable results. The database history information is processed automatically on a regular basis.

2. Add I/O Samples to the Repository.

Collect I/O samples online using a standard Data Collector request (DGIOSUM), and add them to the Repository using SPMDIRUP as in the previous step. An alternate source of I/O summary data is the Unicenter CA-SpaceMan IOLOG, processed in batch using the SPMNENSU job. I/O summary data is then added to the Repository file by matching it to the Current Database information.

3. Create a Projection from Repository data.

The online Unicenter CA-SpaceMan system allows you to create a new projection from the Current Database, an I/O sample, or a previous configuration. The projection is the modification stage of the Unicenter CA-SpaceMan database maintenance process, and it contains all the information necessary for executing the Unicenter CA-SpaceMan Configurator. When creating a projection, you must specify the type of database maintenance option that it will drive. After a projection is created, you can modify it to allow for database file growth, new systems, growth in I/O loads, etc.

4. Create a configuration from a projection.

Using the Unicenter CA-SpaceMan Database Configurator, take a projection and, for some options, an I/O sample and create a new database layout that optimizes I/O across available packs. Browse the results. If the resulting configuration is not to your liking, re-modify the projection to produce another configuration.

5. Generate Adabas utility control cards from a configuration.

After you have created a satisfactory configuration, Unicenter CA-SpaceMan generates appropriate Adabas utility control cards and JCL to perform the database maintenance originally requested when you created the projection. These cards are used to drive Adabas and Unicenter CA-SpaceMan database maintenance utilities.

Adding Current Database Information to the Repository File

The Unicenter CA-SpaceMan Repository file is the central repository for information about all databases. This section assumes the Repository file has already been allocated.

Updating Current Database Information

The Repository maintains information on the Current Database space configuration. You will want to update this information on a regular basis, possibly daily, before using online Unicenter CA-SpaceMan, or after any database modification such as a file load or file delete. The update is a two-step process.

1. The first step is to execute program SPMUPD from the Unicenter CA-APAS LOAD library.

This is a standard batch program. Access is required to the Adabas LOAD library, the Unicenter CA-APAS LOAD library, and the data sets for the database being accessed. The program reads the database disk data sets directly. It does not access the Adabas nucleus I/O buffer pool, so it can be run with Adabas either active or inactive. If an Adabas update utility is in progress, for example, a File Load, SPMUPD does not continue execution, but terminates with a condition code 12.

The JCL for executing the first step of the update process is contained in source library member XSPMUPD. Note the previous discussion of the LP and LWKP2 parameters.

The output of SPMUPD is a sequential file written to HISTRECS. This file contains database, file, extent, volume, and data set information that describes the Current Database configuration. When updating the Repository Current Database, this file becomes input to the second step of the update process.

2. The second step is to run the batch Natural program SPMDIRUP from the Natural SPACEMAN library.

The JCL to run this program is found in the Unicenter CA-APAS SOURCE library member XDIRUPD. File HISTRECS from SPMUPD is read as a Natural work file, CMWKF01. SPMDIRUP then adds or replaces the current configuration name ****CURRENTnnnnn**** where *nnnnn* is the database-id.

This database update information becomes the basis for all subsequent I/O samples for this database. The Current Database is now available to online or batch Unicenter CA-SpaceMan for use in building a new projection or configuration.

Adding I/O Samples to the Repository

I/O samples can also be used as input to the Unicenter CA-SpaceMan Configurator functions. They are also of interest in themselves for reviewing where I/O activity is occurring. Properly timing, selecting, and combining I/O samples can influence the accuracy of your database reconfiguration efforts. We include tips on how to manage this process efficiently in the following paragraphs.

I/O Sample Timing

You should determine periods of the day which are critical to performance and measure I/O during these periods. Considering that I/O bottlenecks usually occur over relatively brief periods, the duration of an I/O sample is extremely important. Too long a sample tends to average and flatten peak measurements, while too short a sample may not reflect the full I/O load. You may also wish to measure during these peak periods on a number of different days to establish the average daily I/O peak load, and to build a standard I/O sample that can be used to drive database maintenance functions.

I/O Sampling Procedure

Collecting the I/O samples for the Unicenter CA-SpaceMan configuration is a two-step process.

1. Collect I/O samples using a standard Data Collector request, DGIOSUM.

This request is found in the Unicenter CA-APAS SOURCE library member SPMDFLTS, and also in the INSIGHT Natural library, member DGIOSUM. Do not modify this request except for the time interval of the sample period. The output from this request should be written to a sequential file, which is then input to the SPMDIRUP maintenance program.

To collect more than one sample during an MPM session, the Unicenter CA-APAS user can ADD the DGIOSUM request, collect data for a specified interval, DELETE the request, copy the output to another file, and then ADD the request again when another interval is desired. It is also possible to MOD several samples together on the output data set.

2. To update the Repository file with the I/O sample data, execute the batch Natural program SPMDIRUP from the SPACEMAN Natural library.

The JCL is in the Unicenter CA-APAS SOURCE library member XDIRUPD. An I/O sample that was previously collected by the DGIOSUM request is read as a Natural work file, CMWKF02. If there are several samples in the work file, each is added separately.

It is permissible to execute SPMDIRUP with input for both CMWKF01, a current configuration, and CMWKF02, an I/O sample, in the same run. If only an I/O is being read, make CMWKF01 a dummy data set.

All counts in the I/O sample are normalized to a rate of I/O per hour so that the counts from samples of different time durations may be compared. However, when viewing the available I/O samples through online Unicenter CA-SpaceMan, the counts displayed are actual counts. This is so that the true size of the sample may be seen.

After you execute SPMDIRUP, the I/O sample is available to online or batch Unicenter CA-SpaceMan for use in building a new projection or configuration that optimizes physical I/O for the sample period.

Online Maintenance

To add, update, and delete records in the Repository file, use the Unicenter CA-SpaceMan online system. This is a standard Natural application. Installation instructions are found in the chapter “Unicenter CA-SpaceMan Installation” of this manual. Log on to the Natural SPACEMAN library and execute the command “SPACEMAN”. Full online help is included with this system.

You may wish to choose the GUIDED option to take advantage of the Unicenter CA-SpaceMan menu system.

Creating Projections, Configurations, and Utility Control Cards

To create a projection from a Current Database or I/O sample data you have populated the Repository file with, create a configuration from a Current Database or a projection you created, and generate utility control cards from a configuration you created, you must use the online Unicenter CA-SpaceMan facility.

These activities are discussed in more detail in the next chapter “Getting Started Using Online Unicenter CA-SpaceMan.”

Getting Started Using Online Unicenter CA-SpaceMan

This chapter provides a sample Unicenter CA-SpaceMan session intended to walk you through the online portion of the Unicenter CA-SpaceMan space management process. We assume the Repository file is allocated and has been populated by at least one Current Database and I/O sample as described in the chapter “Before Getting Started Online.” After the sample session are in-depth discussions of the three Configurator options and the OS Space Allocator utility and how they perform.

Online Menu System

Once you have invoked the online facility, you can use the Unicenter CA-SpaceMan menu system to proceed

From selecting a desired Current Database or I/O sample to using that selection to create a database projection

- From viewing and modifying that projection to using it as input to create a configuration
- From viewing the results of the configuration to generating utility control cards and JCL to use in reconfiguring the database

Sample Online Session

The Unicenter CA-SpaceMan online system has many options in order to handle the various requirements for reporting and configuring a database. This section gives a script for working through the major Unicenter CA-SpaceMan database maintenance functions. We suggest that you first read through the script, then use it as a guide as you execute your own online Unicenter CA-SpaceMan sessions. Once the basic process is understood, you should have few problems performing functions to meet the actual requirements of your site.

Prior to working with the online system, load the directory file with database information and an I/O sample. This procedure is covered in the chapter "Before Getting Started Online."

Getting Started Online

Signon to Natural, logon to the SPACEMAN Natural library, and execute the command "SPACEMAN".

You are initially presented with the Unicenter CA-SpaceMan copyright notice and logo screen.

```

                COPYRIGHT 2003 COMPUTER ASSOCIATES INTERNATIONAL, INC.
          ALL RIGHTS RESERVED. CONTAINS TRADE SECRETS AND CONFIDENTIAL PROPRIETARY
                INFORMATION OF COMPUTER ASSOCIATES INTERNATIONAL, INC.
    REVERSE ENGINEERING PROHIBITED. COPYRIGHT NOTICE DOES NOT IMPLY PUBLICATION.

SSSSSSS  PPPPPP  AAAAAAA  CCCCCC  EEEEEEE  MM      MM  AAAAAAA  NN      NN
SS       PP  PP  AA   AA  CC      EE      MMM   MMM  AA   AA  NNN   NN
SS       PP  PP  AA   AA  CC      EE      MMMM  MMMM AA   AA  NN  N  NN
SSSSSSS  PPPPPP  AAAAAAA  CC      EEEEE   MM  M  M  MM  AAAAAAA  NN  N  NN
        SS  PP      AA   AA  CC      EE      MM  MM  MM  AA   AA  NN   N  NN
        SS  PP      AA   AA  CC      EE      MM      MM  AA   AA  NN   NNN
SSSSSSS  PP       AA   AA  CCCCCC  EEEEEEE  MM      MM  AA   AA  NN   NN

*****
*          S P A C E  M A N A G E M E N T   F O R   A D A B A S          *
*****

    CA-SPACEMAN VERSION 4.01.01 FOR ADABAS V7 & NATURAL V3
CHANGE SESSION OPTIONS BELOW, IF DESIRED:
    NAVIGATION MODE: G F=FREE,  G=GUIDED
    LOWERCASE READOUT: Y Y=YES,  N=NO

                                PRESS < ENTER > FOR LIFTOFF
    
```

Notice the two options at the bottom of the screen. You can select the navigation mode GUIDED or FREE. Guided navigation provides menus arranged by the portion of the Unicenter CA-SpaceMan space management process, as described earlier.

These menus make Unicenter CA-SpaceMan easier for the novice or infrequent user to use, while the free navigation option allows an experienced user to access Unicenter CA-SpaceMan functions by typing in a four-character code. The following sample walk-through follows the guided navigation mode. Type the letter G into the NAVIGATION MODE field.

The other option available on the Unicenter CA-SpaceMan logo panel is the Lowercase Readout option. If your terminal does not display lowercase characters, set this option to NO by typing the letter N. Press ENTER.

The Unicenter CA-SpaceMan Main Menu

The next display you see is the Unicenter CA-SpaceMan Main Menu. The first four selections on the Main Menu correspond to the space management process. The next three selections are additional history reporting and analysis functions Unicenter CA-SpaceMan provides.

```

2002-11-21          CA-SPACEMAN GUIDED NAVIGATION          GUIDMAIN
21:40:51.2                MAIN MENU

CODE  FUNCTION
-----
 1    CURRENT DATABASES
 2    I/O SAMPLES
 3    PROJECTIONS
 4    CONFIGURATIONS

 5    SPACE USAGE HISTORY FUNCTIONS
 6    SPACE USAGE FORECASTING FUNCTIONS
 7    NATURAL FUSER FILE SPACE USAGE

 8    REPOSITORY MAINTENANCE FUNCTIONS

 9    START A NEW CA-SPACEMAN SESSION
 .    TERMINATE CA-SPACEMAN

 A    CA-APAS HISTORY APPLICATION
 I    CA-INSIGHT      APPLICATION
-----
ENTER CODE: _

```

Under the guided navigation option, many Unicenter CA-SpaceMan screens contain an entry field at the bottom labeled '! TO RETURN. On a Guided Menu panel, like the Main Menu shown above, the only entry field available is the ENTER CODE field. Typing the code corresponding to your menu selection into the ENTER CODE field invokes that function.

Type the number 1 in the ENTER CODE field and press ENTER.

Each time you type a code in the ENTER CODE or '! TO RETURN field, you must press ENTER before any action is taken. From this point, we assume that after you have typed in a code you will press ENTER.

The Current Databases Menu

You are now presented with the Current Databases Menu. Notice that the line CURRENT SELECTIONS at the bottom of the screen shows that no Current Database has yet been selected.

```

2002-11-21          CA-SPACEMAN GUIDED NAVIGATION          CDBMENU
21:47:50.5          CURRENT DATABASES MENU

CODE  FUNCTION
-----
A    SELECT A DATABASE FROM A LIST
      USE BELOW --AFTER-- SELECTION
B    UPDATE DESCRIPTION
C    VIEW DATABASE RECORD
D    BROWSE FILE RECORDS
E    VIEW FILE RECORDS
F    BROWSE COMPONENTS
G    VIEW 'WORK' INFO
H    VIEW DATASET INFO
I    BROWSE DATASET RABN MAPS
J    BROWSE DATASET UNUSED SPACE
K    BROWSE DATASET ALTERNATE RABNS
L    BROWSE VOLUMES INFO
M    VIEW VOLUMES INFO
.    RETURN TO MAIN MENU
-----
ENTER CODE:
CURRENT SELECTIONS =      DBID:      0          FILE:      0

```

You must select a Current Database on which to operate, so type the letter A in the ENTER CODE field.

The Select a Current Database Panel

The Select A Current Database panel presents a list of Current Database configurations from the Repository file. You should see at least one Current Database listed as ****CURRENTnnnnn****, where *nnnnn* is the DBID of the database being monitored by Unicenter CA-SpaceMan.

Each of the selection screens (Current Database, I/O Sample, Projection, and Configuration) displays a set of Repository records that, if more than one page, can be displayed page by page by pressing ENTER.

If you found an item you wanted to select, you would type its list sequence number on the page in the SELECT field. Once selected, the item is then used for all subsequent report and configuration functions until you select a different item.

You might have only one configuration record to display at this point.

2002-11-21 21:59:37.8	CA-SPACEMAN / CURRENT DATABASES SELECT A CURRENT DATABASE	SCDB ABEJU01	
CURRENT DATABASE	DBID VER	DATE/TIME	DESCRIPTION
1 **CURRENT00003**	3 7.1	2002-06-06 11:53:14	THIS IS A VERSION 7.1 DATABASE
2 **CURRENT00007**	7 7.4	2002-11-18 10:11:30	THIS IS A VERSION 7.4 DATABASE

SELECT: 0_
'.' TO RETURN:
WHEN SELECTING, ALSO ENTER '.' TO RETURN

Type the letter of a Current Database in the SELECT field and a period (.) in the '.' TO RETURN field.

Notice that your selection is now displayed at the bottom of the Current Databases Menu.

Displaying Current Database Information

The View DB Record Panel The Current Databases Menu shows all of the reports that are available for directory records. Type the letter C in the ENTER CODE field to select the VIEW DATABASE RECORD function.

The VIEW DATABASE RECORD function shows general database information.

2002-11-21 22:13:27.3	CA-SPACEMAN / CURRENT DATABASE **CURRENT00007** VIEW DB RECORD	UDBA ABEJU01
--------------------------	---	-----------------

DBID: 7	DB NAME: TECH-DBID007	MAX FILES: 255
	LOAD DATE: 19940719 201720	RABNSIZE : 3

TOTAL DATABASE SIZE (KB)	ASSO	DATA	WORK
MINIMUM ALLOCATION	54904	303789	15403
ACTUAL USED	17528	153601	

DATASET	DEV	FROMRABN	TO RABN	DATASET	DEV	FROMRABN	TO RABN
ASSOR1	3390	1	21582	DATAR1	3390	1	59990
ASSOR2		0	0	DATAR2		0	0
ASSOR3		0	0	DATAR3		0	0
ASSOR4		0	0	DATAR4		0	0
ASSOR5		0	0	DATAR5		0	0

'.' TO RETURN: _____

Type a period to return to the Current Databases Menu.

At the Current Databases Menu, type the letter D in the ENTER CODE field. The Browse Files panel is displayed.

The Current Database Browse Files Panel

The Browse Files panel lists all the files of the selected Current Database. You can position a given file at the top of the display by putting its number in the STARTING FILE field. File 0, the generated file, is the space used by the GCB, FCB, FDT, and other control blocks.

2002-11-21		CA-SPACEMAN / CURRENT DATABASE						BFIL	
22:15:23.7		**CURRENT00007** BROWSE FILES						ABEJU01	
FILE	FILE-NAME	LOAD DATE	ACISEXUC	AP	DP	#EXT UNAD!	A ALLOC KB USED	D ALLOC KB USED	RD/HR WR/HR
0	GENERATED FILE	2002-10-28	NNNNNNNN	0	0	0100	40755	150187	0
1	EMPLOYEES	2002-10-24	NNNSNNNC	10	10	1111	249	759	0
2	VEHICLES	2002-10-24	NNNSNNNN	10	10	1111	178	233	0
3	MISCELLANEOUS	2002-10-24	NNNSNNNN	10	10	1111	75	759	0
11	NAT-USER	2002-10-21	NNISNNNN	10	10	1212	71	61	0
12	NAT-SYSTEM	2002-10-21	NNISNNUN	10	10	1112	97	759	0
19	CHECKPOINT	1994-07-19	NNNSNNNN	1	1	1111	86	269	0
							1011	22788	0
							547	11774	0
							1726	121536	0
							1272	58019	0
							93	506	0
							19	92	0

STARTING FILE: 19_____ DBID: 7
 '.' TO RETURN: _____

Tab to the '.' TO RETURN field and type a period to return to the Current Databases Menu. Type the letter E in the ENTER CODE field to View File Records from the Current Database.

The Current Database View Files Panel

You get an error message at the top of the panel because the VIEW FILES function does not yet know which file to display.

Type the number of your EMPLOYEES Adabas demo file in the NEXT FILE field.

You can invoke other browse and display functions from the Database Menu for the Current Database.

```

2002-11-21          CA-SPACEMAN / CURRENT DATABASE          UFIL
22:16:45.8          **CURRENT00007** VIEW FILES           ABEJU01

FILE:      1 FILE NAME: EMPLOYEES                          DBID      7
           LOAD DATE: 20021024 142529

           PADDING ASSO: 10 % DATA: 10 %                OPTIONS I: N A: N X: N
           MINISN :          1                          S: S C: N U: N
           MAXISN :          1695                        E: N C: C
           SORTSEQ:          ISNSIZE: 4
           MAXRECL: 5060  MAXDS:          0  MAXUI:      0  MAXNI:      0

COMPONENTS  TYPE #EXT  KB ALLOC  KB USED  IO READS  IO WRITES
          AUI   1     33      33      33
          ANI   1    211     140
          AAC   1     5       5
          DDA   1    759     233

NEXT FILE: _____
'. ' TO RETURN: _____
    
```

Tab to the '. ' TO RETURN field and type a period to return to the Database Menu to again view all the possible display functions. Then type a period in the ENTER CODE field to return to the Main Menu.

Displaying I/O Sample Information

The I/O Samples
Menu

From the Main Menu, type the number 2 to display the I/O Samples Menu.

```
2002-11-21          CA-SPACEMAN GUIDED NAVIGATION          IOSMENU
22:19:17.1          I/O SAMPLES MENU

CODE  FUNCTION
-----
A     COMBINE SAMPLES
B     SELECT A SAMPLE FROM A LIST
      USE BELOW --AFTER-- SELECTION
C     VIEW/UPDATE/DELETE SAMPLE
D     VIEW DATABASE RECORD
E     BROWSE FILE RECORDS
F     VIEW FILE RECORDS
G     BROWSE COMPONENTS
H     VIEW 'WORK' INFO
I     VIEW DATASET INFO
J     BROWSE VOLUMES INFO
K     VIEW VOLUMES INFO

      . RETURN TO MAIN MENU
-----
ENTER CODE: _

CURRENT SELECTIONS = DBID:      0  START DATE TIME: 0000-00-00 00:00:00
```

Type the letter B in the ENTER CODE field. The I/O Sample Selection screen is displayed.

The Select a Sample Panel

You may now select an I/O sample for reporting and configuration. An I/O sample has structure similar to the current database configuration, but with I/O counts added. The number of reads and writes in each sample is displayed. When selecting an I/O sample, you may browse through a multi-page list of I/O samples in the selection panel by pressing ENTER, or by putting the date-time identifier of the item you want to start the list with in the FULL START DATE-TIME TO SCAN FROM field near the bottom of the panel.

2002-11-21 22:22:01.5	CA-SPACEMAN / IO SAMPLES SELECT A SAMPLE	SIOS ABEJU01
--------------------------	---	-----------------

##	START DATE-TIME END DATE-TIME	DBID ADA SMPID VER	READ IOS WRITE IOS	DESCRIPTION
1	2002-06-05 18:20:57 2002-06-05 19:14:10	3 7.1	356 223	THIS IS A V 7.1 DATABASE SAMPLE
2	2002-11-18 11:04:34 2002-11-18 12:03:44	7 7.4	185 35	THIS IS A V 7.4 DATABASE SAMPLE

0_ <--SELECT ## _____ <--FULL START DATE-TIME TO SCAN FROM
'.' TO RETURN:
WHEN SELECTING, ALSO ENTER '.' TO RETURN

Type the number 1 in the SELECT ## field. Tab to the '.' TO RETURN field and type a period to return to the I/O Samples Menu.

Type 'E' in the ENTER CODE field. The I/O Sample Browse Files panel is displayed.

The I/O Sample
Browse Files Panel

The reads and writes are expressed as a rate per hour so that you can compare or combine samples of different duration.

You should now be displaying records similar to the current database file record, but now I/O counts are displayed for the I/O sample period. File 0, the generated file, represents activity against the GCB, FCB, FDT, and other control blocks.

2002-11-21 22:31:10.5		CA-SPACEMAN / BROWSE FILES					BFIL ABEJU01		
FILE	FILE-NAME	LOAD DATE	ACISEXUC	AP	DP	#EXT UNAD!	A ALLOC KB USED	D ALLOC KB USED	RD/HR WR/HR
0	GENERATED FILE	2000-06-03	NNNNNNN	0	0	0100	3537	0	21
							3537	0	53
1	EMPLOYEES	1995-08-28	NNISNNN	10	5	1111	388	650	24
							213	222	5
3	EMPLOYEE-ADAM	1996-01-04	ANISNNN	5	20	1111	137	506	0
							75	429	0
14	NAT228-SYSTEM	1995-09-25	NNISNNN	3	3	1111	831	25064	128
							459	18124	9
15	NAT228-FUSER	1995-09-25	NNISNNN	10	15	1111	1596	20244	194
							176	8787	27
18	SPACEMAN-V330	1996-05-09	NNISNNN	10	5	1111	131	1301	25
							90	25	61
19	CHECKPOINT	1995-08-28	NNNSNNN	10	10	1111	39	482	3
							23	15	2

STARTING FILE: 19____ DBID: 3
'.' TO RETURN: _____

Tab to the '.' TO RETURN field and type a period to return to the I/O Samples Menu. Type the letter G in the ENTER CODE field. The Browse Components panel is displayed.

The I/O Sample
Browse Components
Panel

This shows I/O counts for individual database components. The file component level is the lowest level of data maintained in the Repository file, and the level at which the database is configured by Unicenter CA-SpaceMan.

2002-11-21 22:32:06.1			CA-SPACEMAN / BROWSE DB COMPONENTS					BCMP ABEJU01	
FILE	CMP	NEW	FROM RABN	TO RABN	VOLUME	KB ALLOC	KB USED	IO RPH READS	IO RPH WRITES
0	AL1		8079	8082	GSW003	10	10	0	0
0	DS1		1381	1391	GSW003	27	27	3	23
0	FV0		1	1380	GSW003	3510	3510	18	30
0	DL1		26977	26991	GSW005	72	72	0	0
0	WL1		2008	2016	GSW003	51	51	0	0
0	W10		1	201	GSW003	1150	1150	5	94
0	W20		202	301	GSW003	572	572	0	0
0	W30		302	2016	GSW003	9816	9816	1	0

STARTING FILE: 1____ DBID: 3
'.' TO RETURN: _____

If you wish to look at the components of a particular file, you can type the file number in the STARTING FILE field. If you press ENTER without specifying a file number, you can browse the files in sequence, displaying the next file report each time you press ENTER.

Tab to the '.' TO RETURN field and type a period to return to the I/O Samples Menu, then type a period into the ENTER CODE field to return to the Main Menu.

Creating a Projection to Reload/Add Files

The Projections Menu From the Main Menu, type the number 3 to display the Projections Menu.

```
2002-11-21          CA-SPACEMAN GUIDED NAVIGATION          PRJMENU
22:33:54.3          PROJECTIONS MENU

CODE  FUNCTION
-----
A    CREATE A PROJECTION
B    SELECT A PROJECTION FROM A LIST
      USE BELOW --AFTER-- SELECTION
C    ADD/UPDATE/DELETE PROJECTION
D    ADD/UPDATE/DELETE DATABASE RECORD
E    BROWSE FILE RECORDS
F    ADD/UPDATE/DELETE FILE RECORDS
G    VIEW/UPDATE 'WORK' INFO
H    VIEW/UPDATE DATASET INFO
I    BROWSE VOLUMES INFO
J    ADD/UPDATE/DELETE VOLUMES

.    RETURN TO MAIN MENU
-----
ENTER CODE: _
CURRENT SELECTIONS =  DBID:      0          FILE:      0
```

Type the letter A in the ENTER CODE field to select the Create a Projection option.

The Create New Projection Panel

You will now create a projection. A projection is basically a copy of database or I/O sample data which you can modify to reflect projected growth or changes in the database before you do any manipulation of the database.

The Create New Projection panel shows an INPUT CONFIG with the name of the Current Database you selected, and an INPUT I/O SAMPLE with the date-time identifier of the I/O sample you selected.

```

E264 - MUST SPECIFY PROJECTION NAME
2002-11-21          CA-SPACEMAN DATABASE PROJECTIONS          CPRO
22:34:28.7          CREATE NEW PROJECTION                    ABEJU01

PROJECTION NAME: _____ DBID 3__
DESCRIPTION: _____

INPUT CONFIG: _____ <====
OR
I/O SAMPLE: 20020605182057 <==== |-- ENTER ONE ONLY

OPTION: 0      1 = RECONFIGURATION  ALL FILES COPIED
              2 = REALLOCATION      VOLUME RECORDS ONLY
              3 = ADD/RELOAD       ADD/MODIFY FILES WITH UFIL
                                   SFIL INVOKED AUTOMATICALLY
                                   FOR FILE SELECTION

CANCEL: _____ 'CAN' CANCELS CREATION

'. ' TO RETURN: _____
    
```

Creating a projection consists of copying records from either an input Current Database record or an input I/O sample. The latter has I/O loads to aid in balancing I/O across packs.

Type PROJECT 1 in the PROJECTION NAME field, and type RELOAD 1 AND 2 in the DESCRIPTION field. Erase the name in the INPUT CONFIG field.

Type the number 3 in the OPTION field and press ENTER.

By specifying option 3, you are asking for a Reload/Add File type of projection. This projection uses the I/O sample to place components or reloaded files in a way that balances I/O across disks. If you had selected the Current Database configuration, no I/O data would be available, and unload/reload JCL would still be generated, but the reloaded file would not necessarily improve the I/O balance.

The Select a Projection Panel

The projection PROJECT 1 is listed as a projection. You may have to scroll through the list by pressing ENTER until you see PROJECT 1 displayed.

2002-11-21 22:45:02.3	CA-SPACEMAN / PROJECTIONS SELECT A PROJECTION				SPRO ABEJU01
PROJECTION	OPT	DBID	DATE-TIME	DESCRIPTION	
1 PROJECT 1	3	3	2002-11-18 22:44:52	RELOAD 1 AND 2	
2 PROJ4DB5	3	5	2000-06-06 17:49:11		

SELECT: 0_ NAME: _____
'.' TO RETURN: _____
WHEN SELECTING, ALSO ENTER '.' TO RETURN

Type the PROJECT 1 projection number, in this case number 1, in the SELECT field and a period in the '.' TO RETURN field. The Projections Menu is displayed.

At the Projections Menu, type the letter E to select the BROWSE FILE RECORDS Option.

The I/O Sample
Browse Files Panel

A file list for PROJECT 1 with files 1 and 2 listed is displayed.

2002-11-21 22:48:54.3		CA-SPACEMAN / PROJECTION PROJECT 1 BROWSE FILES					BFIL ABEJU01		
FILE	FILE-NAME	LOAD DATE	ACISEXUC	AP	DP	#EXT UNAD!	A ALLOC KB USED	D ALLOC KB USED	RD/HR WR/HR
1	EMPLOYEES	0000-00-00	NNNSNNNC	10	10	1111	249 178	759 233	0 0
2	VEHICLES	0000-00-00	NNNSNNNN	10	10	1111	75 71	759 61	0 0

STARTING FILE: 1____ DBID: 3
'.' TO RETURN: _____

Type a period in the '.' TO RETURN field to return to the Projections Menu. Type the letter F in the ENTER CODE field.

The Projection
Update Files Panel

At the Update Files panel, type the number 1 in the NEXT FILE field and press ENTER.

```

2002-11-21          CA-SPACEMAN / PROJECTION          UFIL
22:50:42.6          PROJFNRI2 UPDATE FILES          ABEJU01

FILE:  ___1  FILE NAME: EMPLOYEES_____  DBID  3
LOAD DATE:

      PADDING ASSO: 10 %  DATA: 10 %          OPTIONS I: N  A: N  X: N
      MINISN  : _____1                    S: S  C: N  U: N
      MAXISN  : _____1695                  E: N  C: C
      SORTSEQ: _____ISNSIZE: 4
      MAXRECL: _5060  MAXDS:  ___0  MAXUI:  ___0  MAXNI:  ___0

COMPONENTS  TYPE  #EXT  KB ALLOC  KB USED  IO READS  IO WRITES
      AUI      1      33      33      _____
      ANI      1     211     140      _____
      AAC      1      5       5      _____
      DDA      1     759     233      _____

NEXT FILE:  _____  'CAN' CANCELS CHANGES
              'ADD' ADD NEW FILE  'DEL' DELETE FILE
'. ' TO RETURN:  _____
    
```

You may now update the data for file 1. This allows you to modify the sizes of components, the padding factors, and projected I/O loads before the file is reloaded.

Type the number 10 in both PADDING FACTOR fields, ASSO and DATA, type the number 2000 in the MAXISNS field, and double each number in the component KB ALLOC fields. Press ENTER.

You have now updated the file in PROJECT 1 to allow for approximately twice the number of records currently loaded. You may also adjust I/O load figures if you think a file may become more active.

You could also use this function to add a new file record. The UPDATE FILES program generates JCL to add the file. To do this, you would type values in all the fields of the UPDATE FILES screen and enter the ADD command in the NEXT FILE field. If you forgot to enter the ADD command, the current record would be updated instead, including the file number.

Type a period in the '. ' TO RETURN field to return to the Projections Menu. Then type a period in the ENTER CODE field to return to the Main Menu.

Creating a Configuration From a Projection

The Configurations Menu From the Main Menu, type the number 4 to display the Configurations Menu.

```

2002-11-21          CA-SPACEMAN GUIDED NAVIGATION          CNFMENU
22:53:15.5          CONFIGURATIONS MENU

CODE  FUNCTION
-----
A    CREATE A CONFIGURATION
B    SELECT A CONFIGURATION FROM A LIST
      USE BELOW --AFTER-- SELECTION
C    VIEW/UPDATE/DELETE CONFIGURATION
D    VIEW DATABASE RECORD
E    BROWSE FILE RECORDS
F    VIEW FILE RECORDS
G    BROWSE COMPONENTS
H    VIEW 'WORK' INFO
I    VIEW DATASET INFO
J    BROWSE DATASET RABN MAPS
K    BROWSE VOLUMES INFO
L    VIEW VOLUMES INFO
M    PREVIEW UTILITY CONTROL CARDS
.    RETURN TO MAIN MENU
-----

ENTER CODE:
CURRENT SELECTIONS =  DBID:    3  **CURRENT00003**  FILE:    1

```

Type the letter A in the ENTER CODE field.

```

2002-11-21          CA-SPACEMAN / CONFIGURATOR          CCNF
22:53:59.4          CREATE NEW CONFIGURATION          ABEJU01

PROJECTION NAME: PROJECT 1
(PROJECTION HAS OPTION INDICATING CONFIGURATION TYPE TO RUN)

INPUT CONFIG: _____ <====
OR
I/O SAMPLE: 20020605182057 <==== |-- ENTER ONE ONLY
(OPTIONS 2 AND 3 ONLY)

OUTPUT CONFIG: _____

```

'.' TO RETURN: _____

Verify that PROJECT 1 is in the PROJECTION NAME field. Erase any name in the INPUT CONFIG field, but keep the selected I/O sample.

Type RELOAD1-2 in the OUTPUT CONFIG field and press ENTER.

The Configurator now generates a complete configuration. If the Create Configuration operation is successful, the Save Configuration screen is displayed. Press ENTER to complete the configuration. If an error occurs, you can use the Help function to examine the cause of the error by typing the error number in the NEXT FUNCTION field.

Type a period in the '!' TO RETURN field to return to the Configurations Menu. Type the letter B in the ENTER CODE field. The configuration selection list is displayed.

The Select A Configuration Panel You are now shown the Configuration Selection panel.

```

2002-11-21          CA-SPACEMAN / CONFIGURATIONS          SCNF
22:56:23.8          SELECT A CONFIGURATION              ABEJU01

  CONFIGURATION    OPT DBID VER DATE/TIME                DESCRIPTION
1 RELOAD1-2       3   3  7 2002-11-21
                   22:55:55

```

```

SELECT: 0_        NAME: _____
'. ' TO RETURN: _____
WHEN SELECTING, ALSO ENTER '. ' TO RETURN

```

If the configuration RELOAD1-2 is not displayed on the panel, type RELOAD1-2 in the NAME field and press ENTER.

Once the RELOAD1-2 configuration is shown, type the corresponding configuration number in the SELECT field and a period in the '. ' TO RETURN field. The Configurations Menu is displayed.

Type the letter G in the ENTER CODE field to browse the file components.

The Configuration
Browse Components
Panel

At the Browse Components panel, press ENTER to display the file 1 components. This shows the new placement of components for file 1. Press ENTER to see the new placement for file 2.

2002-11-21 22:57:39.7			CA-SPACEMAN / CONFIGURATION RELOAD1-2 BROWSE COMPONENTS					BCMP ABEJU01	
FILE	CMP	NEW	FROM RABN	TO RABN	VOLUME	KB ALLOC	KB USED	IO RPH READS	IO RPH WRITES
1	AC1	0	7089	7090	GSW002	5	0	0	0
1	NI1	0	7006	7088	GSW002	211	0	0	0
1	UI1	0	7143	7155	GSW002	33	0	0	0
1	DA1	0	29983	30132	GSW002	759	0	0	0

STARTING FILE: 2_____ DBID: 3
'.' TO RETURN: _____

Type a period in the '.' TO RETURN field to return to the Configurations Menu. Type the letter M in the ENTER CODE field. The Generate Utility Control Cards panel is displayed.

The Generate Utility
Control Cards Entry
Panel

Generate and display the control cards that are used to reorganize files 1 and 2.

2002-11-21
22:58:55.6

CA-SPACEMAN / DATABASE MAINTENANCE
GENERATE UTILITY CONTROL CARDS

GCRD
ABEJU01

CONFIGURATION NAME: RELOAD1-2_____

NOTE: THE GENERATE CARD FUNCTION WILL GENERATE CONTROL CARDS
BASED ON THE CONFIGURATION OPTION FOR THE CONFIGURATION.
IT MAY BE INVOKED ONLINE TO DISPLAY GENERATED CARDS, BUT
IT MUST BE EXECUTED IN BATCH TO SAVE GENERATED CARDS.

TYPE '?' BELOW FOR MORE INFORMATION.

'.' TO RETURN: _____ <ENTER> TO CONTINUE

Type RELOAD1-2 in the CONFIGURATION NAME field, if it is not already displayed. Press ENTER. The following screen is displayed.

2002-11-21 22:59:55.1	CA-SPACEMAN / DATABASE MAINTENANCE GENERATE UTILITY CONTROL CARDS	GCRD ABEJU01
--------------------------	--	-----------------

```
ERRLIM : ____0
LIP    : ____0
LPB    : ____0
NUMREC : _____
UTYPE  : _____
TEMPSIZE: ____0
SORTSIZE: ____0
TEST   : _ (Y/N)

SELECT (X) ONE: ADALOD LOAD
                ADAORD REORFILE _
```

ENTER ADDITIONAL VALUES FOR THE UTILITY, AND PRESS <ENTER>

When generating the utility output from an Option-3 configuration, you have a choice of two different Adabas utilities. You can choose to generate the necessary statements to unload the file, if present, delete the file, if present, and load the new file. Alternatively you can choose to generate the necessary statements to invoke the ADAORD REORFILE utility, which may be more efficient.

This screen also contains several parameters that may be included with the generated utilities. A parameter is only included if a value for that parameter is entered on the screen. See the *Adabas Utilities Manual* for the appropriate utility to determine the possible values for each parameter and the effect it has on execution of the utility.

The Generated
Control Cards Panel

The Generate Utility Control Card (GCRD) function allows you to view all the JCL the program would generate in order to unload and reload files 1 and 2. The actual JCL generated is based on source in Natural's SPACEMAN library. For information on modifying this JCL, see the chapter "Unicenter CA-SpaceMan Installation" in this manual.

WWW

Page 1

JCL FOR CONFIGURATION RELOAD1-2
WRITTEN TO WORK FILE 2

```
//RLD00001 JOB (ACCOUNT,FILEREORG),,CLASS=A,MSGCLASS=X,TYPRUN=HOLD
//*
/** THIS JCL IS USED BY THE GCRD FUNCTION TO GENERATE JCL TO
/** REORDER BOTH ASSOCIATOR AND DATA FOR A SINGLE FILE. THE
/** LAST 5 CHARACTERS OF THE JOBCARD WILL BE SET TO THE FILE NO.
/**
/** YOU MUST MODIFY THIS MEMBER TO CONFORM TO YOUR LOCAL JCL
/** STANDARDS AND YOUR ADABAS PROCS.
/**
/** THE FOLLOWING IS A FAIRLY TYPICAL PROC FOR EXECUTING THE ADAORD
/** UTILITY. IT IS ASSUMED THAT THERE IS A PARMLIB CONTAINING A
/** SELECTION OF ADARUN CARDS.
/**
//ADAREF PROC DBID=DDDDD,
// FILE=NNNNN,
// MODE=MULTI
/**
```

Continue pressing ENTER until you return to the Generate Utility Control Cards panel. A message is displayed indicating that control cards have been generated for this configuration.

To actually generate this JCL, you must execute GCRD in batch. The JCL generated is for Adabas utilities depending on the version of the Current Database used. The JCL is output to work files which may be PDS members.

Type a period in the '!' TO RETURN field and press ENTER. Do this again to return to the Main Menu.

Creating a Projection to Reallocate the Database

Reallocating the database allows you to spread a database across multiple disk volumes without reorganizing the internal structure or the database. This is done to equalize and spread I/O across as many disk devices as possible.

From the Main Menu, type the letter 3 to reach the Projections Menu.

```

2002-11-21          CA-SPACEMAN GUIDED NAVIGATION          PRJMENU
23:05:37.8          PROJECTIONS MENU

CODE  FUNCTION
-----
A    CREATE A PROJECTION
B    SELECT A PROJECTION FROM A LIST
      USE BELOW --AFTER-- SELECTION
C    ADD/UPDATE/DELETE PROJECTION
D    ADD/UPDATE/DELETE DATABASE RECORD
E    BROWSE FILE RECORDS
F    ADD/UPDATE/DELETE FILE RECORDS
G    VIEW/UPDATE 'WORK' INFO
H    VIEW/UPDATE DATASET INFO
I    BROWSE VOLUMES INFO
J    ADD/UPDATE/DELETE VOLUMES

.    RETURN TO MAIN MENU
-----
ENTER CODE: _

CURRENT SELECTIONS =  DBID:    3  PROJECT 1          FILE:    1
  
```

Type the letter A in the ENTER CODE field to create a new projection.

The Create New
Projection Panel

Type REALC1 in the PROJECTION NAME field and REALLOCATE DATABASE in the DESCRIPTION field. Erase the INPUT CONFIG field, leaving the ID in IO SAMPLE. If an I/O sample is not displayed, go to the Select I/O Sample (SIOS) function, select a sample, and return to the create new projection display.

Type the number 2 in OPTION and press ENTER. The confirmation message should appear on the panel like this.

```

2002-11-21          CA-SPACEMAN DATABASE PROJECTIONS          CPRO
23:06:25.0                CREATE NEW PROJECTION                ABEJU01

PROJECTION NAME: _____ DBID 3____
DESCRIPTION: REALLOCATE DATABASE_____

INPUT CONFIG: _____ <====
OR
I/O SAMPLE: 20020605182057 <==== |-- ENTER ONE ONLY

OPTION: 0      1 = RECONFIGURATION  ALL FILES COPIED
              2 = REALLOCATION      VOLUME RECORDS ONLY
              3 = ADD/RELOAD       ADD/MODIFY FILES WITH UFIL
                                   SFIL INVOKED AUTOMATICALLY
                                   FOR FILE SELECTION

THE NEW PROJECTION FOR REALC1 HAS BEEN CREATED.

CANCEL: _____ 'CAN' CANCELS CREATION

'. ' TO RETURN: _____

```

You have now created a reallocation projection. This consists only of volume records. You may update, add or delete volumes onto which the database can be allocated.

Return to the Projections Menu and type the letter B in the ENTER CODE field to select the new projection from a list.

The Select a
Projection Panel

If REALC1 is not displayed on the initial screen, type REALC1 in the NAME field and press ENTER. The projection REALC1 is listed as a projection.

2002-11-21 23:15:52.4	CA-SPACEMAN / PROJECTIONS SELECT A PROJECTION			SPRO ABEJU01	
	PROJECTION	OPT	DBID	DATE-TIME	DESCRIPTION
	1 REALC1	2	3	2002-11-21 23:08:01	REALLOCATE DATABASE
	2 PROJECT 1	3	3	2002-11-21 22:44:52	

SELECT: 0_ NAME: _____
'.' TO RETURN:
WHEN SELECTING, ALSO ENTER '.' TO RETURN

Type the REALC1 projection number, in this case the number 1, in the SELECT field, a period in the '.' TO RETURN field, and press ENTER. The Projections Menu is displayed.

Type the letter I for the Browse Volumes panel.

The Browse Volumes
Panel

The Browse Volume function displays the information from the volume records
contained in the projection.

2002-11-21 23:18:46.7		CA-SPACEMAN / PROJECTION REALC1 BROWSE VOLUMES						BVOL ABEJU01	
VOLSER	DEV	ASSO CYLS	DATA CYLS	WORK CYLS	AVAIL CYLS	ASSO IO RPH	DATA IO RPH	WORK IO RPH	NONADA IO RPH
GSW002	3390	0	0	0	500	0	0	0	0

STARTING VOL: _____ DBID: 3

'.' TO RETURN: _____

Return to the Projections Menu and type the letter J in the ENTER CODE field to
display the volume's information.

The Update Volumes Panel Type in a volume name from the BROWSE VOLUMES list in the NEXT VOLUME field and press ENTER. The volume's configuration information is displayed.

```

2002-11-21          CA-SPACEMAN / PROJECTION          UVOL
23:20:59.8          REALC1 UPDATE VOLUMES            ABEJU01

VOLUME: GSW002      DEVICE: 3390                      DBID:    3

CONFIGURATION INPUT:
  AVAILABLE SPACE:  __500 CYLS
  NON ADABAS I-O LOAD: _____ 0

CONFIGURATION OUTPUT:
  DATASETS  SEQ#  SPACE(CYLS)  IO RPH
  ASSO      0    _____  0
  DATA     0    _____  0
  WORK      0    _____  0

NEXT VOLUME: _____ 'CAN' CANCELS CHANGES
                  'ADD' ADD NEW VOLUME 'DEL' DELETE VOLUME
'. ' TO RETURN: _____
    
```

You can now update this volume record. You may increase or decrease the number of cylinders available and modify the NON-ADABAS I/O LOAD value which is taken into account in I/O load balancing.

Now create two new volume records. Enter ADNEW1 in the VOLUME field, your standard device type in the DEVICE field, the number 5 in the AVAILABLE SPACE field, ADD in the NEXT VOLUME field, and press ENTER. A message is displayed indicating that the new record was added.

Type ADNEW2 in the VOLUME field, your standard device type in the DEVICE field, the number 5 in the AVAILABLE SPACE field, ADD in the NEXT VOLUME field, and press ENTER. You have now added two new volumes to the volumes usable for Adabas files.

Type a period in the '! ' TO RETURN field and press ENTER. Once again, type the letter I to browse the volume information.

The Projection Browse Volumes Panel The Browse Volumes panel now displays the new volumes together with the old ones. If necessary, enter a value for the starting volume to reposition the volume display.

2002-11-21 23:24:05.8		CA-SPACEMAN / PROJECTION REALC1 BROWSE VOLUMES					BVOL ABEJU01			
VOLSER	DEV	ASSO CYLS	DATA CYLS	WORK CYLS	AVAIL CYLS	ASSO IO RPH	DATA IO RPH	WORK IO RPH	NONADA IO RPH	
ADNEW1	3390	0	0	0	5	0	0	0	0	
ADNEW2	3390	0	0	0	5	0	0	0	0	
GSW002	3390	0	0	0	500	0	0	0	0	

STARTING VOL: _____ DBID: 3
'.' TO RETURN: _____

Update the available space for the current volumes if you wish, again from the Update Volumes panel. Type a period in the '.' TO RETURN field to return to the Projections Menu, then type a period in the ENTER CODE field to return to the Main Menu.

At the Main Menu, type the number 4 to display the Configurations Menu. Type the letter A in the ENTER CODE field.

The Save
Configuration Panel

Type REALC1 in the PROJECTION NAME field, if it is not already displayed.

Erase the name in the INPUT CONFIG field. Type REALC1 in the OUTPUT CONFIG field. If an I/O sample is not displayed, go to the I/O Samples Menu, select a sample, and return to the Create New Configuration panel. Press ENTER after all required information is supplied. If the Create Configuration operation was successful, the Save Configuration panel is displayed.

2002-11-21
23:25:17.8

CA-SPACEMAN / CONFIGURATOR
SAVE CONFIGURATION

CSLC
ABEJU01

CONFIGURATION SAVED WITH THE NAME: REALC1

PRESS ENTER TO CONTINUE

Press ENTER to complete the configuration process.

You have now generated a configuration which has the database spread across the volume set based on I/O load balancing without changing the internal structure.

Return to the Configurations Menu and type the letter B in the ENTER CODE field.

At the Configuration Selection screen, select the REALC1 configuration and return to the Configurations Menu.

Type the letter K in the ENTER CODE field.

The Configuration
Browse Volumes
Panel

The Browse Volumes function shows the I/O load for each volume for the configuration. These values should be moderately well-balanced, given the constraint that the internal structure of the database has not been altered.

2002-11-21 23:26:56.3		CA-SPACEMAN / CONFIGURATION REALC1 BROWSE VOLUMES					BVOL ABEJU01		
VOLSER	DEV	ASSO CYLS	DATA CYLS	WORK CYLS	AVAIL CYLS	ASSO IO RPH	DATA IO RPH	WORK IO RPH	NONADA IO RPH
ADNEW1	3390	5	0	0	5	26	0	0	0
ADNEW2	3390	5	0	0	5	62	0	0	0
GSW003	3390	20	0	16	45	0	0	17	0
GSW005	3380	0	200	0	200	0	116	0	0

STARTING VOL: _____ DBID: 3
'.' TO RETURN: _____

Return to the Configurations Menu.

If you select Browse Components from the Configurations Menu, you can view the components in file order. The Browse Dataset RABN Maps display shows the components in RABN order.

Return to the Configurations Menu and type the letter M in the ENTER CODE field.

The Generate Utility
Control Cards Panels

The Generate Utility Control Card (GCRD) function allows you to view the allocator control cards on your terminal. The actual JCL is generated in batch based on source in Natural's SPACEMAN library. For information on modifying this JCL, see the chapter "Unicenter CA-SpaceMan Installation" in this manual .

2002-11-21
23:28:50.5

CA-SPACEMAN / DATABASE MAINTENANCE
GENERATE UTILITY CONTROL CARDS

GCRD
ABEJU01

CONFIGURATION NAME: REALC1_____

**NOTE: THE GENERATE CARD FUNCTION WILL GENERATE CONTROL CARDS
BASED ON THE CONFIGURATION OPTION FOR THE CONFIGURATION.
IT MAY BE INVOKED ONLINE TO DISPLAY GENERATED CARDS, BUT
IT MUST BE EXECUTED IN BATCH TO SAVE GENERATED CARDS.**

TYPE '?' BELOW FOR MORE INFORMATION.

'.' TO RETURN: _____ <ENTER> TO CONTINUE

Type REALC1 in the CONFIGURATION NAME field and press ENTER.

Continue pressing ENTER until you return to the panel where you entered the REALC1 configuration name.

Type a period in the '.' TO RETURN field and press ENTER to return to the Configurations Menu. Repeat this process to return to the Main Menu.

Creating a Projection to Reorganize the Database

Reorganization refers to a total restructuring of all internal file components on the database in addition to reallocating the database on volumes.

From the Main Menu, type the letter 3 to reach the Projections Menu.

```
2002-11-21          CA-SPACEMAN GUIDED NAVIGATION          PRJMENU
23:35:22.4          PROJECTIONS MENU

CODE  FUNCTION
-----
A    CREATE A PROJECTION
B    SELECT A PROJECTION FROM A LIST
      USE BELOW --AFTER-- SELECTION
C    ADD/UPDATE/DELETE PROJECTION
D    ADD/UPDATE/DELETE DATABASE RECORD
E    BROWSE FILE RECORDS
F    ADD/UPDATE/DELETE FILE RECORDS
G    VIEW/UPDATE 'WORK' INFO
H    VIEW/UPDATE DATASET INFO
I    BROWSE VOLUMES INFO
J    ADD/UPDATE/DELETE VOLUMES

.    RETURN TO MAIN MENU
-----
ENTER CODE: _

CURRENT SELECTIONS =  DBID:      3  REALC1          FILE:      0
```

Type the letter A in the ENTER CODE field to create a new projection.

The Create New Projection Panel

Type REORG1 in the PROJECTION NAME field and REORGANIZE THE DATABASE in the DESCRIPTION field.

Erase the INPUT CONFIG field, leaving the identifier in the I/O SAMPLE. If an I/O sample is not displayed, go to the I/O Samples Menu, select a sample, and return to this Create New Projection panel.

Type the number 1 in the OPTION field and press ENTER. If successful, a confirmation message is displayed on the panel as shown here.

```

2002-11-21          CA-SPACEMAN DATABASE PROJECTIONS          CPRO
23:36:21.5                CREATE NEW PROJECTION                ABEJU01

PROJECTION NAME: _____ DBID 3__
DESCRIPTION: REORGANIZE THE DATABASE_____

INPUT CONFIG: _____ <====
OR
I/O SAMPLE: 20020605182057 <==== |-- ENTER ONE ONLY

OPTION: 0      1 = RECONFIGURATION ALL FILES COPIED
              2 = REALLOCATION      VOLUME RECORDS ONLY
              3 = ADD/RELOAD       ADD/MODIFY FILES WITH UFIL
                                   SFIL INVOKED AUTOMATICALLY
                                   FOR FILE SELECTION

THE NEW PROJECTION FOR REORG1 HAS BEEN CREATED.

CANCEL: _____ 'CAN' CANCELS CREATION

'. ' TO RETURN: _____
    
```

You have now generated a complete copy of the I/O sample data input. At this point, you can adjust file sizes with the Add/Update/Delete Files function from the Projections Menu to allow for growth. You can also adjust volume allocations and add volumes as with the reallocation option. Try adding the same volumes as in the previous projection to compare the results of reallocation and reorganization.

Return to the Projections Menu and type letter J in the ENTER CODE field to select the Add/Update/Delete Volumes function.

The Update Volumes Panel

Type ADNEW1 in the VOLUME field, your standard device type in the DEVICE field, the number 300 in the AVAILABLE SPACE field, ADD in the NEXT VOLUME field and press ENTER. If successful, a confirmation message is displayed.

```

2002-11-21          CA-SPACEMAN / CONFIGURATION          UVOL
23:38:40.3         REORG1 UPDATE VOLUMES                ABEJU01

VOLUME: ADNEW1      DEVICE: 3390                        DBID:   3
CONFIGURATION INPUT:
  AVAILABLE SPACE:  ___300 CYLS
  NON ADABAS I-O LOAD: _____0

CONFIGURATION OUTPUT:
  DATA SETS  SEQ#  SPACE(CYLS)  IO RPH
  ASSO       0     _____  0
  DATA      0     _____  0
  WORK       0     _____  0

      THE NEW RECORD FOR FILE ADNEW1 HAS BEEN ADDED.
NEXT VOLUME: ADD ___      'CAN' CANCELS CHANGES
                          'ADD' ADD NEW VOLUME  'DEL' DELETE VOLUME
'. ' TO RETURN: _____

```

Type ADNEW2 in the VOLUME field, your standard device type in the DEVICE field, the number 300 in the AVAILABLE SPACE field, ADD in the NEXT VOLUME field and press ENTER. Again, if successful, a confirmation message is displayed.

Now create a configuration from the Option 1 projection you just modified. Return to the Main Menu. Type the number 4 to reach the Configurations Menu.

Type the letter A and press ENTER to create a configuration.

The Create New
Configuration Panel

Type REORG1 in the PROJECTION NAME field, if it is not already displayed.

2002-11-21
23:53:38.4

CA-SPACEMAN / CONFIGURATOR
CREATE NEW CONFIGURATION

CCNF
ABEJU01

PROJECTION NAME: REORG1
(PROJECTION HAS OPTION INDICATING CONFIGURATION TYPE TO RUN)

INPUT CONFIG: _____ <====
OR |-- ENTER ONE ONLY
I/O SAMPLE: 20020605182057 <==== (OPTIONS 2 AND 3 ONLY)

OUTPUT CONFIG: _____

'.' TO RETURN: _____

Erase the name in the INPUT CONFIG field. Also erase the I/O sample name. Type REORG1 in the OUTPUT CONFIG field and press ENTER. If the Create Configuration operation was successful, the Save Configuration screen is displayed. Press ENTER to complete the configuration process.

You have now generated a configuration which has Adabas file components allocated in an optimal fashion for I/O load balancing.

Return to the Configurations Menu and type the letter B in the ENTER CODE field to select the newly-created configuration.

The Select A
Configuration Panel

If the configuration is not displayed on the initial screen, type REORG1 into the NAME field and press ENTER.

```

2002-11-21          CA-SPACEMAN / CONFIGURATIONS          SCNF
23:55:30.7          SELECT A CONFIGURATION              ABEJU01

  CONFIGURATION    OPT DBID VER DATE/TIME          DESCRIPTION
 1 REALC1          2   3  7 2002-11-21
                   23:14:54
 2 RELOAD1-2      3   3  7 2002-11-21
                   22:55:55
 3 REORG1          1   3  7 2002-11-21
                   23:30:42

```

```

          SELECT: 0_          NAME: _____
'. ' TO RETURN: _____
          WHEN SELECTING, ALSO ENTER '. ' TO RETURN

```

Type the REORG1 configuration's selection number, in this case the number 3, into the SELECT field and a period in the '. ' TO RETURN field to return to the Configurations Menu.

Now that you have created and selected the configuration, the Browse Components option on the Configurations Menu shows you the components in file order. The Browse Dataset RABN Maps option shows the components in RABN order. But for now, we'll just browse the volume information.

Type the letter K in the ENTER CODE field to select the BROWSE VOLUMES INFO option.

The Browse Volumes Panel

This function shows the I/O load for each volume for the configuration. These values should be better balanced than those for the REALC1 configuration. If they are not significantly better, then the rather complex reorganization process is not worth doing.

2002-11-21 23:57:09.8		CA-SPACEMAN / CONFIGURATION REORG1 BROWSE VOLUMES					BVOL ABEJU01		
VOLSER	DEV	ASSO CYLS	DATA CYLS	WORK CYLS	AVAIL CYLS	ASSO IO RPH	DATA IO RPH	WORK IO RPH	NONADA IO RPH
ADNEW1	3390	2	298	0	300	14	44	0	0
ADNEW2	3390	300	0	0	300	55	0	0	0
GSW003	3390	0	30	15	45	0	37	17	0
GSW005	3380	17	183	0	200	19	35	0	0

STARTING VOL: _____ DBID: 3
'.' TO RETURN: _____

Return to the Configurations Menu and type the letter M in the ENTER CODE field.

The Generate Utility
Control Cards Panel

Type REORG1 in the CONFIGURATION NAME field, if it is not already displayed and press ENTER.

2002-11-21
23:58:15.8

CA-SPACEMAN / DATABASE MAINTENANCE
GENERATE UTILITY CONTROL CARDS

GCRD
ABEJU01

CONFIGURATION NAME: REORG1_____

NOTE: THE GENERATE CARD FUNCTION WILL GENERATE CONTROL CARDS
BASED ON THE CONFIGURATION OPTION FOR THE CONFIGURATION.
IT MAY BE INVOKED ONLINE TO DISPLAY GENERATED CARDS, BUT
IT MUST BE EXECUTED IN BATCH TO SAVE GENERATED CARDS.

TYPE '?' BELOW FOR MORE INFORMATION.

'.' TO RETURN: _____ <ENTER> TO CONTINUE

When generating the utility output from an Option-1 configuration, you have a choice of two different Adabas utilities. You may choose to generate the ADAORD REORASSO and ADAORD REORDATA as separate utilities to reorder each component independently. Alternatively, you can choose to generate the single ADAORD REORDB utility to reorder both Associator and Data components in a single execution.

2002-11-21
23:58:54.6

CA-SPACEMAN / DATABASE MAINTENANCE
GENERATE UTILITY CONTROL CARDS

GCRD
ABEJU01

ERRLIM : _____0
LIP : _____0
LPB : _____0
NUMREC : _____
UTYPE : _____
TEMPSIZE: _____0
SORTSIZE: _____0
TEST : _ (Y/N)

SELECT (X) ONE: ADALOD LOAD _
ADAORD REORFILE _

ENTER ADDITIONAL VALUES FOR THE UTILITY, AND PRESS <ENTER>

This screen also contains several parameters that may be included with the generated utilities. A parameter is only included if a value for that parameter is entered on the screen. See the *Adabas Utilities Manual* for the appropriate utility to determine the possible values for each parameter and the effect it has on the execution of the utility.

The control cards for the allocator and the Adabas REORDER utilities are displayed on your terminal.

MORE
Page 1 2002-11-21 23:58:54.6

SPACEMAN ALLOCATOR CONTROL CARDS FOR REORG1
WRITTEN TO WORK FILE 1

```
ADNEW2 000300 02544 GSA.DBID003.ASSOR1
GSW005 000017 02004 GSA.DBID003.ASSOR1
ADNEW1 000002 02544 GSA.DBID003.ASSOR1
ADNEW1 000298 05064 GSA.DBID003.DATAR1
GSW003 000030 05064 GSA.DBID003.DATAR1
GSW005 000183 04820 GSA.DBID003.DATAR1
GSW003 000015 05724 GSA.DBID003.WORKR1
```

Continue pressing ENTER until you return to the panel where you entered the configuration name. Type a period into the field labeled '.' TO RETURN to return to the Configurations Menu.

To actually perform the reorganization, you must run the GCRD program in batch to generate and save the necessary utility control cards.

Type a period and press ENTER to return to the Main Menu.

Note: This script has taken you through the main functions of the Unicenter CA-SpaceMan online system. We hope it has given you a general idea of how it fits together. We appreciate your feedback on all aspects of the system and are always ready to answer your questions.

Unicenter CA-SpaceMan Configurator

The function of the Unicenter CA-SpaceMan File Management for Adabas (Unicenter CA-SpaceMan) Configurator is to optimize the placement of Adabas files and/or data sets based on averaging the I/O load across disk volumes. The Configurator provides three separate options for reorganizing a database:

- Option 1 – complete reconfiguration of the database
- Option 2 – reallocation of the database
- Option 3 – individual file maintenance

Configurator Overview

Once you have used the SPMREP reports to analyze your space management problems, you can select one or more of these configuration options as your space management solution. You can compare the results of one option to another online, and use that comparison to choose your optimal configuration.

Configurator Input

The Unicenter CA-SpaceMan Configurator requires as input, for any of its options, an input projection name and an output configuration name. For options 2 and 3 you can also supply either a configuration name or an I/O sample selection as additional input. The input projection contains an indication of the Configurator option to be performed and other input parameters unique to this Configurator run. The output configuration name is the name under which the resultant configuration records are saved for later display and utility control card generation for implementation of the configuration. The input configuration name or the I/O sample selected provides the Configurator with the Current Database configuration of files and data sets and the associated I/O loads of each of these components.

Configurator Execution

The Configurator is invoked under Natural either in batch or interactively online. Currently, interactive execution of the Configurator is limited to TSO, CICS or COM-LETE. In interactive mode, an initial panel is displayed on which the required input names are specified. When the Configurator completes, another panel is displayed indicating the resultant configuration was saved and then the initial panel is displayed again.

Each of the Unicenter CA-SpaceMan Configurator options is separately detailed in the following sections, including descriptions of the input and output.

Option 1: Completely Reconfiguring the Database

Option 1 makes the most extreme changes of any of the Configurator options. This option provides for both reallocating the database data set and reconfiguring the internal mapping of database components. The DBA enters volumes with available space and selects an I/O sample with an I/O load to be balanced.

The Configurator then generates a new configuration, placing each file component within volumes and determining the data set allocations to balance the I/O. A complete reorganization can require a good deal of processor time, but it results in the greatest improvements in database performance.

Input to the Configurator

The Configurator takes as input the following projection record types:

- The MAXFILE setting for the database from the database record.
- Minimum sizes for ASSO and DATA data sets from the database record.
- A set of volume records created in the projection which describe the volumes to be configured containing:
 - Volume serial and device type
 - The available space on the volume in cylinders
 - Any non-Adabas I/O load for the volume (I/Os per hour)
- A set of file records which specify each file component's size and I/O load.

Gathering Data

As the input records are passed to the Configurator, it accumulates information regarding the total size of the individual components and the amount of space available for allocation on the given volume set. It also receives the minimum sizes required for the Associator and Data data sets so the current database can be reorganized successfully. A check is made after receiving all the input to insure that the total size requested for the components is less than the available volume space and that the available volume space is equal or greater than the minimum database size.

Sorting Data

Next, the volumes are sorted into order of decreasing available space. Two sorted orders of the individual components are made, one in order of decreasing size and the other in order of decreasing I/O load.

Configurator Processing

Given this input, the Configurator determines the data set sizes to be allocated on each volume, the volume order within each data set, the component placement in each data set as specified by its RABNs, and the I/O load allocated to each volume by the placement of the files.

Configurator Processing Algorithm

1. The total I/O load is calculated from the individual component I/O loads and the volume non-Adabas loads. Average I/O load per volume is then determined.
2. From this information, the number of packs per data set type (ASSO, DATA, WORK) based on the I/O load for each type is determined. A proposed order of pack usage for each data set type is made based on: volume available space sizes; number of volumes required for I/O load; separation of data set types.
3. Before allocation of the individual component entries to a volume, the first volume for a data set type is initialized to remove the first track from being available due to Adabas Siemens support. For the Associator data set, the fixed blocks required at the front of the data set are removed from availability. If Work components have been specified, they are allocated next due to the required order of the parts of the Work data set.
4. The sorted size order of components is interpreted to determine whether any components are larger than the available space on any one volume. Such components are split among as many volumes as required to handle the size. The I/O load for such components is split proportionally.
5. Once the components requiring immediate size splitting are allocated, the I/O load sorted order of components is used for remaining component allocations. Unallocated components are taken in turn. The component I/O load is checked to see whether it is greater than volume average; if so, the component is split in half and allocated to adjacent volumes of the appropriate data set type. Its I/O load is split between the volumes.
6. For other components, it is determined whether they are Associator components. An attempt is made to allocate Associator components that have another Associator component of the same file already allocated to a volume to the same volume if its I/O load fits within that volume's average load.

7. All other components and those above that won't fit on the tested volume are tried on each volume in order of proposed usage for the data set type until a volume is found where the component's I/O load fits.
8. Once a volume is selected, it is checked to see whether space is available for the component. If the size test fails, a new volume selection is made until the component is allocated or fails due to either I/O load greater than volume average or size not available.
9. For an I/O load fit failure, a list of volumes is made in increasing order of current I/O load. These volumes are tried in order to see if the component fits and is allocated to the first one it fits.
10. For size fit failures, the component is split between two adjacent volumes where sufficient space is available and a split has not already occurred. The I/O load for such a split is split between the volumes proportional to the size split.
11. After all components have been allocated, any remaining available space not allocated to a data set on a volume is parceled out in proportion to the allocated space on the volume. The RABN ranges for each volume of a data set are determined. Then the starting and ending RABN for each component is determined, where the components of the same file are placed in adjacent RABNs on a volume.

Output from the Configurator

All the volume allocations, allocated I/O loads, and component RABNs are given back to be entered into the Repository file for control card generation if a database reorganization is desired based on the new allocations.

The Configurator produces as output:

- A set of volume records containing, in addition to the input parameters, for each type of data set:
 - The number of cylinders allocated
 - The volume I/O load
 - Relative volume number within the data set
- A set of component records containing, in addition to the input parameters, the volume serial to which the component is allocated and the starting and ending RABNs.

These Configurator output records are stored, along with the database record and file records copied from the projection, into the Unicenter CA-SpaceMan Repository file under the output configuration name. From there they are available for display and subsequent utility control card generation if desired.

Natural program GCRD for Option 1 creates control cards for the OS Space Allocator. Data set names for this function are drawn from the data set records for the Current Database or for the configuration.

Option 2: Reallocating the Database

This option of the Unicenter CA-SpaceMan Configurator achieves the best possible load balance across physical DASD volumes without changing the internal location of database components. The DBA specifies a set of volumes and the space available on the volumes. A sampled I/O load is selected and then the Configurator partitions the Adabas data sets in such a way as to equalize the I/O load for this sample as much as possible.

This option has the advantage of minimal resource usage and complexity for the benefits gained, but does not gain the benefits of optimizing the internal structure of the database.

Input to the Configurator

Option 2 of the Configurator takes as input the following projection record types:

- The current number of cylinders allocated for each Adabas data set type for example ASSO, DATA, or WORK, from the input configuration volume records.
- A set of volume records created in the input projection which describe the volumes to be configured containing:
 - Volume serial and device type
 - The available space on the volume
 - Any non-Adabas I/O load for the volume (I/Os per hour)
- A set of component records taken from the specified input configuration or I/O sample containing:
 - File number and component-id
 - I/O load for component
 - Size in K-bytes
 - From and to RABNs

Gathering Data

As the input records are passed to the Configurator, it accumulates the amount of space available for allocation on the given volume set. It also receives the number of cylinders currently allocated for each data set type. The component records are checked to insure they have a valid RABN range specified. A check is made after receiving all input to insure the available volume space is equal to or greater than the required number of cylinders for reallocation of the database.

Sorting Data

Next, the volumes are sorted into order of decreasing available space. The total I/O load is calculated from the individual component I/O loads and the volume non-Adabas loads, from which Unicenter CA-SpaceMan determines the average I/O load per volume. From this information the number of packs per data set type (ASSO, DATA, WORK) based on the I/O load for each type is determined. A proposed order of pack usage for each data set type is made based on: volume available space sizes; number of volumes required for I/O load and separation of data set types.

Configurator Processing

With this set of input, the Configurator determines the data set sizes to be allocated on each volume and the volume order within each data set.

Natural program GCRD for Option 2 creates control cards for the OS Space Allocator. Data set names for this function are drawn from the data set records for the Current Database or for the configuration.

Configurator Processing Algorithm

1. If Work components have been specified, they are allocated to the volume set first, due to the required order of the parts of the Work data set. The remaining two data set types (ASSO and DATA) are taken separately and processed as described in the following steps. The volume set is processed in the proposed order for the data set type.
2. The components, ordered in RABN sequence, are taken one by one until either their cumulative I/O load is greater than the volume average or the end of available space on the volume is reached.
3. The data set size on the volume is truncated to a cylinder boundary, and the RABN range on the volume is calculated.
4. If a component is split by the end-of-volume, the Configurator splits the component I/O load proportional to the size of the component part on the volume.

5. The next volume is taken in order and the processing continues in like manner until all components have been processed. Additional space is allocated for the data set type as required to reach the number of cylinders allocated for the current data set.
6. The relative volume number for each data set type is determined by the order of the RABN range allocated to each volume in ascending sequence.

Output from the Configurator

The Configurator produces as output a set of volume records containing, in addition to the input parameters, for each type of data set:

- The number of cylinders allocated
- The volume I/O load
- Relative volume number within the data set

The set of component records input to the Configurator are returned to be saved in the output configuration. These Configurator output records, along with the database record and file records copied from the projection, are stored into the Unicenter CA-SpaceMan Repository file under the output configuration name. Subsequent displays and allocation control card generation can be made from the configuration.

Option 3: Adding/Reloading Individual Files

Option 3 of the Unicenter CA-SpaceMan Configurator allows the DBA to optimally place new files into the database to maintain a good balance of I/O across DASD volumes. The DBA specifies the new file's characteristics in the input projection, including each component's size and projected I/O load. In addition, files already present in the database may be specified in the projection. Such files are repositioned to best advantage in the output configuration.

A sample I/O load is selected and then the Configurator uses the Current Database configuration to position the specified files within the database. This option can be used to compress files which have components in multiple extents.

Input to the Configurator

Option 3 of the Configurator takes as input:

- A set of file records for the new files to be allocated containing component size and I/O load.
- A set of volume records from the input configuration or I/O sample containing for each volume:
 - Volume serial and device type
 - Any non-Adabas I/O load for the volume (I/Os per hour)
 - The number of cylinders allocated and relative volume number for each data set type on the volume
- A set of component records taken from the input configuration or I/O sample specifying the current file configuration and containing:
 - File number and component-id
 - I/O load
 - Size in K-bytes
 - From and to RABNs

Gathering Data

As the input records are passed to the Configurator, the component records are checked and flagged as being a new file, a reallocation of an existing file, or an unchanged existing file.

Sorting Data

Next, the volumes are ordered into relative volume order for each data set type based on the relative volume number. The beginning and ending RABNs for a volume are initialized by type.

The existing components for each data set type are ordered in ascending RABN sequence for each volume.

Two sorted orders of the individual components are created, one in order of decreasing I/O load and the second in order of decreasing size.

Configurator Processing

Given this set of input, the Configurator allocates the specified new files within the free space in the current configuration while maintaining an I/O load balance across volumes.

Configurator Processing Algorithm

The average volume I/O load is calculated, and the individual volume I/O load of already existing components is determined.

1. A chain of the free space elements for each volume of each data set type is created taking into account an element that spans volumes. The components to be allocated are then taken in turn from the I/O sorted order of components.
2. A volume is selected by testing to see whether the component I/O load sends the volume's total I/O load above the I/O average. For Associator components, the first volume tried is one that already has Associator components for the same file allocated on it. Otherwise, the first volume tried is the first relative volume for the data set type. The volumes are tried in order until the component fits on the volume within the average I/O and in a free space element on the volume, or it fails either the I/O test or fails because of insufficient free space on the volume.
3. For an I/O load test failure, the volumes are retried in order to see whether the component fits sizewise without regard to I/O loading; for a size fit failure, all previously allocated components are backed out and then the sorted size order of components is used to attempt allocation of new components. Any size failure here causes the Configurator to terminate for insufficient free space.

Output from the Configurator

The Configurator produces as output:

- The set of volume records input to the Configurator are returned to be saved in the output configuration.
- The input set of component records are returned to be saved.
- The set of component records for the newly allocated files are returned with the following additional information:
 - From and to RABNs allocated for the component
 - A flag indicating whether this is a new component or a reallocated, existing component

These output records, together with the database and file records from the input configuration or I/O sample, are stored into the Unicenter CA-SpaceMan Repository file under the output configuration name. They are available for generation of the file unload/reload utility control cards.

Natural program GCRD for Option 3 generates JCL based on model JCL held as source in the Unicenter CA-SpaceMan Natural library. These members contain online JCL PROCs which should be changed to conform to your local standards.

The JCL is written to a Natural workfile, CMWKF02, in batch. One job is generated for each file to be loaded or unload/reloaded. The output should be reviewed and the jobs submitted either as a group or released based on available resources.

This version of Unicenter CA-SpaceMan does not include JCL procedures for unloads/reloads of large files using tape for intermediate data sets. You should carefully review jobs generated for large files to make certain that they do not exceed space for intermediate files or TEMP and SORT.

Understanding the Unicenter CA-SpaceMan OS Allocation Utility

The Unicenter CA-SpaceMan OS Allocation Utility, SPMALC, allocates multi-volume OS physical sequential data sets as specified by a set of input data statements. The data sets are allocated to the volumes indicated with the specified size, blocksize and data set name. Each data set is cataloged with all the volumes specified in the order of the input data statements.

Each input data statement specifies the attributes of a data set on a single volume. Subsequent data statements indicate the parameters for an additional portion of the same data set on another volume, or a segment of another data set on a volume. The order of the input statements is only important within statements specifying the same data set name. The data set is allocated and cataloged with the volume order the same as the order of the input data statements for the data set name.

The Unicenter CA-SpaceMan OS Allocation Utility is an authorized program which must be executed from an authorized load library. Authorization is required for two reasons:

- Uses UCB pointer option for some of the CVAF macro functions. UCB option is required for update function of the CVAFDIR macro.
- Enqueues on the SYSVTOC major name when reading format 5 DSCBs for a non-indexed VTOC.

Note: The SPMALC utility cannot be used on SMS controlled volumes.

Reporting and Forecasting DASD Space Usage

Part of your overall space management task for Adabas is to account for actual space usage and to forecast future space usage. It is important to be able to do both things at the Adabas file level, particularly for databases whose logical files are owned by different organizational entities. In the case of a Natural FUSER system file, it can be important to account for its space by LOGON (library or application) ID.

Reporting actual space usage is often required for billing purposes. Past space usage also becomes the basis for forecasting future space usage.

The ability to predict future space usage is important in a number of ways. It supports capacity planning for ensuring that enough total DASD space is in place when it is needed. At the individual Adabas file level, it helps you stay ahead of the game and in control regarding allocation of new logical extents for each file.

This can be critical as files approach the Adabas limit of five extents for the different types of areas, and where the size of the next extent of some type for a file is a significant factor within a database.

Unicenter CA-SpaceMan Space Usage History

Unicenter CA-SpaceMan File Management for Adabas (Unicenter CA-SpaceMan) provides a number of facilities you can use to aid you in this area of space management:

- Process space usage data from the HISTRECS output file created by the SPMUPD program directly, using your own programs
- Post space usage data from the HISTRECS output file into the Unicenter CA-SpaceMan Repository file using the SPMHISUP program
- Create and maintain space usage Accounting Control records in the Repository that specify which entities “own” Adabas files and what the cost of space is for each file
- Generate reports based on the space usage data in the repository, including cost and ownership information from the Accounting Control records
- Forecast future space usage, using a file's historical average rate of growth in actual space usage as a basis for looking ahead
- Generate reports on the space used in Natural FUSER system files by LOGON (library or application) ID, for source and/or compiled members

Each of these facilities is explained in more detail in this chapter.

User Processing of HISTRECS Space Usage Data

The most direct and efficient way to use Unicenter CA-SpaceMan facilities to capture and process space usage data is by executing a combination of SPMUPD and a user-provided program. These facilities are described below.

This method does not incur the additional overhead of having space usage data added to the Repository for later retrieval and eventual deletion. However, if you do not put historical space usage data into the Repository, then you cannot use the space usage accounting and forecasting facilities of Unicenter CA-SpaceMan. This process is described in Posting Space Usage Data to the Repository in this chapter.

HISTRECS File

Each time the SPMUPD program is executed, it produces a HISTRECS output file. Among the various types of records in the HISTRECS file are records which give space usage data for each component of the logical files of the database.

Sample User Processing

In the Unicenter CA-SpaceMan Natural library is a sample skeleton program, SPMCHARG, which reads the HISTRECS file and selects only the component records for further processing. You can add code to the program to provide this processing. By referencing the space usage data items in these records, the user-provided code can either generate billing data at that point, or it can write an output Work File which can later be used as input to a billing system.

If you adopt this method of accounting for space usage, then you should determine the frequency of executing SPMUPD and other related Unicenter CA-SpaceMan programs. If space usage data is to be generated on a daily basis, then SPMUPD should be executed daily to capture the space usage data for each day.

The output files from SPMUPD can then be processed daily by SPMCHARG or another user-provided program, or they can be saved for later processing on some other cycle.

Posting Space Usage Data to the Repository

In addition to, or instead of, processing HISTRECS data by user-provided programs, you can put space usage data into the Repository using the SPMHISUP program. The data put into the Repository by SPMHISUP can then be used by the Unicenter CA-SpaceMan space usage and forecasting facilities which are described below.

The Unicenter CA-SpaceMan programs which report space usage assume that space usage data has been added to the Repository on a daily basis; for example, that SPMUPD and SPMHISUP have been executed once each day for each database. In this version of Unicenter CA-SpaceMan there is no logic to compensate for days for which data was not added to the Repository. The total usage reported for a file is the sum of the daily postings of space usage.

Sample JCL for executing SPMHISUP is in member XHISUPD of the Unicenter CA-APAS SOURCE library.

Retention Control Records

SPMHISUP is influenced by the Retention Control record for a given database if such a record is present in the Repository when SPMHISUP is executed. The SPMHISUP logic is as follows:

- If no Retention Control record is present for the database, then SPMHISUP adds posting records which contain daily space usage data for reporting and accounting purposes and adds and/or updates base records which support calculation of file space usage growth rates and forecasting of future space usage.
- If a Retention Control record is present for the database and specifies a zero value retention period for posting records, then posting records are not added to the Repository.
- If a Retention Control record is present for the database and specifies a zero value retention period for base records, then base records are not added or updated.

By setting appropriate values in the Retention Control record for a database, the DBA can suppress the use of either posting records, base records or both. An alternative way to suppress both kinds of records is to simply never run SPMHISUP against the HISTRECS records for the database.

Adding Retention Control Records

Retention Control records can be added and updated through selections that are available from the guided navigation menus of the Unicenter CA-SpaceMan online Natural application. Select REPOSITORY MAINTENANCE FUNCTIONS from the Main Menu. Then select RETENTION CONTROL RECORD MAINTENANCE from the Repository Maintenance Functions Menu.

Using Accounting Control Records

The purpose of accounting control records is to assign ownership of Adabas files to specified accounts, owners and/or applications, and to specify a cost factor for space usage. If none of these features is needed, it is not necessary to have accounting control records in the Repository. Without them, space usage can still be reported by Adabas file within database. With them, space usage and space usage cost can be reported in various combinations by account, owner and application.

Creating Accounting Control Records

Accounting control records can be added and updated through selections that are available from the guided navigation menus of the Unicenter CA-SpaceMan online Natural application. Select SPACE USAGE HISTORY FUNCTIONS from the Main Menu; then select ACCOUNTING CONTROL RECORD MAINTENANCE from the Space Usage History Functions Menu.

Applying Accounting Control

A given accounting control record is applied at a database level or at an individual file level. If the file number in the record is zero, then the control information in the record applies to any files within that database which do not have their own accounting control records. With a non-zero file number, an accounting control record applies only to the specified file.

Accounting Control Effective Period

In order to accommodate changes in ownership or cost values for a database or file, each accounting control record is effective for a specified time period. Effective time periods of two different accounting control records for the same database with file number zero or file may not overlap. Thus, when a new accounting control record is being added for a database or file, it may be necessary to first update an existing accounting control record to set a new ending date that is less than the starting date in the record that will be added.

Reporting Space Usage History

Unicenter CA-SpaceMan programs use the daily space usage posting records to report actual past usage of DASD space by Adabas files. Each posting record contains the amount of space allocated to each of the files within a given database at the time the SPMUPD program was executed.

Selecting Reports

Online reporting can be obtained through selections that are available from the guided navigation menus of the Unicenter CA-SpaceMan online Natural application. Select SPACE USAGE HISTORY FUNCTIONS from the Main Menu; then select the type of report desired from the Space Usage History Functions Menu. Before pressing ENTER to select a report, you must type in values which define the scope of the report.

Batch Reporting

This release does not include any batch reporting of space usage.

Forecasting Future Space Usage

Unicenter CA-SpaceMan programs use the space usage base records to estimate future usage of DASD space by Adabas files. Only one base record can exist in the Repository for a given file.

Base Record Contents

The base record contains a number of file status items. In addition, it records the amount of space allocated and the amount used for each extent of the file. In each base record this extent data references two points in time:

- The from point, which is usually the time the file was last loaded, but may be a later time when a significant change that altered the file's space usage growth rate occurred
- The through point, which is always the time of the most recent execution of SPMUPD and SPMHISUP

The time period between the from point and the through point is used as a base period. When forecasting, the Unicenter CA-SpaceMan programs calculate the average daily space usage growth rate during the base period for each type of extent within a file.

Space Forecasting Process

The forecast for each extent type of the file is made for the time period defined by the through point reflected in the base record through the forecast period ending date provided by the user when invoking the program. The number of days in the forecast period is determined, then the average daily growth rate is multiplied by that number of days to estimate total expected space usage growth for the extent type during the forecast period. The space forecast for an extent type of a file is further refined by taking into consideration the sizes of all existing extents of that type and any that are expected to be allocated during the forecast period.

SPMHISUP automatically resets the from point data for a file when it detects that certain changes have occurred in the file. This is done on the assumption that the change has possibly caused any past growth history to become invalid as a basis for forecasting future growth. The reset consists of using the current, that is, the latest, space usage data from the HISTRECS file being processed as both the from and through data for the file. The effect of the reset operation is to delete past growth history for the file and to re-commence accumulating growth history.

The Space Usage Forecasting Functions Menu of the Unicenter CA-SpaceMan online guided navigation system provides an option for you to review and, if desired, cause selected base records to be reset as described above.

Selecting Reports

Online forecasts can be obtained through selections that are available from the guided navigation menus of the Unicenter CA-SpaceMan online Natural application. Select SPACE USAGE FORECASTING FUNCTIONS from the Main Menu; then select FORECAST SPACE GROWTH from the Space Usage Forecasting Functions Menu. Before pressing ENTER to select a report, you must type in values which specify the database, files and ending date of the forecast period.

Batch Reporting

This release does not include any batch forecasting of space usage. As batch programs are added in later releases, they will be described in the release notes which accompany the new release tapes.

Reporting Space Usage for Natural FUSER System Files

Unicenter CA-SpaceMan has online and batch facilities for reporting space usage by library or application-id within Natural FUSER system files. Space usage is calculated separately for source members and for compiled (object) members.

Selecting Reports

Online reporting can be obtained through selections that are available from the guided navigation menus of the Unicenter CA-SpaceMan online Natural application. Select Natural FUSER FILE SPACE USAGE from the Main Menu. At the next screen, indicate the kind of member(s) to be the subject of the report and specify a range of one or more LOGON (library or application) IDs to be included in the report.

Please bear in mind that the response time for this reporting is a function of how many records exist in the system file for the range of IDs you specify. Response time can be several minutes for libraries with very large numbers of members. It may be prudent to limit each online report to a single LOGON ID, specifying it as both the from value and the through value.

To report space usage for a greater range of LOGON-IDs, execute the batch Natural program NATSPCBT using sample JCL XNATSPC. This program produces a printed report and/or a sequential file of used source and object space by LOGON-ID. The format of each output work file is:

LOGON-ID (A8)	/* Natural LOGON
RECORD-COUNT (N10)	/* Number of records with this LOGON
AC-KBYTES (N9.2)	/* Kilobytes of Address Converter used by this LOGON
ASSO-KBYTES (N9.2)	/* Kilobytes of Associator used by this LOGON
DATA-KBYTES (N9.2)	/* Kilobytes of Data Storage used by this LOGON

IOLOG Record Processing

This chapter describes control card syntax for the Data Collector unique to its use with Unicenter CA-SpaceMan File Management for Adabas (Unicenter CA-SpaceMan) in analyzing Adabas physical I/O. Full descriptions of use of the Data Collector, including installation, operation, input, output, use with Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS), and general control card syntax are all contained in the appropriate Unicenter CA-APAS manuals.

Generating IOLOG Records

The Data Collector can process more than one type of record. Specifically:

- CMDLOG records – the Adabas Command Log records from the Adabas nucleus; this is the default.
- IOLOG records – these log records are generated by the DBGIOR5 exit from ADAIOR, supplied with Unicenter CA-SpaceMan. One record is generated for every call to ADAIOR. This section describes all the data fields available in this record.

In order to generate IOLOG records, the Unicenter CA-SpaceMan DBGIOR5 module must be loaded in to the Adabas region. For further details, see the chapter “Unicenter CA-SpaceMan Installation” in this guide .

With the Data Collector active (ENSU=YES), IOLOG records may be processed in one of two ways:

1. EXTRACT or SUMMARIZE IOLOG requests submitted from batch or through Unicenter CA-APAS with output data directed to INSIGHT-LINES, PRINT TO, or an OUTPUT-FILE.
2. A COPY IOLOG request to write the entire IOLOG record to either a single or dual OUTPUT-FILE.

IOLOG Record Layout

The IOLOG record layout is shown below.

IORFIELD	DSECT		LOG RECORD FOR ADAIOR CALLS
IORRECLN	DS	H	VARIABLE RECORD PREFIX.
	DS	H	
IORPROGN	DS	XL1	PROGRAM NUMBER
IORCTYPE	DS	XL1	CALL TYPE
IORFUNCT	DS	XL1	FUNCTION INDICATOR
IORRESP	DS	XL1	RESPONSE
IORSPCID	DS	0F	SPECIAL INDICATOR (CLOG/PLOG)
IODURAT	DS	F	CALL DURATION IN 16 MSECS.
IORRABN	DS	F	RABN
IORCMDCD	DS	CL2	ADABAS COMMAND CODE.
IORFILE	DS	XL2	ADABAS FILE NUMBER.
IORIOSEQ	DS	F	I/O NUMBER
IORIOCMP	DS	F	I/O NUMBER OF COMPLETED CALL(TYPE M).
IORVOLSR	DS	CL6	VOL SER OF I/O.
IODASDC	DS	XL2	CYLINDER NUMBER OF I/O.
IORSTCK	DS	F	STCK OF CALL END.
IORPAMBL	DS	F	PAM BLOCK FOR BS2000. (RELATIVE)
IORCMSQ	DS	F	COMMAND SEQUENCE NUMBER FROM CQE
IORDESC	DS	CL2	DESCRIPTOR OF ASSOCIATOR ACCESS.
IORCOMP	DS	XL1	COMPONENT OF ASSOCIATOR ACCESS?
IORSMPID	DS	XL1	SMPID
	DS	0H	
IORFLDLN	EQU	*-IORFIELD	
IODATLN	DS	H	LENGTH OF IOR DATA. USUALLY 0.
IORDATA	DS	C	DATA; PRINT-LINE, WTO, ENQ

Data Collector Control Card Syntax

The following pages explain control card syntax and describe the parameters which are new or different from those described in the latest Unicenter CA-APAS manuals.

GLOBALS: Definition of Global Environment

Each Data Collector statement input stream should include a GLOBALS statement. It defines the processing environment and values to be used in developing data for reports and output files. The GLOBALS statement does not require a statement label.

Syntax

```

GLOBALS
[ COLSPC = number ]           /* default column spacing
[ CPU-ID = name ]             /* cpu identifier
[ DBID = number ]             /* database number
[ DBNAME = name ]             /* database name
[ DUAL-COPY-EXIT = module_name ]
[ GRAND-TOTAL-TABLE = {YES|NO} ] /* affects SUMMARIZE statements
[ IGNORE-INIT-ERRORS = {YES|NO} ]
[ IOLOG-SWITCH = { ON|OFF} ]    /* controls I/O logging
[ LINESIZE = number ]         /* default chars/line
[ MAXRECS = number ]         /* limit on number of input recs
[ MAXTSIZE = {number in k|800} ] /* limit TSIZE of SUMMARIZE requests
[ PAGESIZE = number ]         /* default lines/page
[ PRINT-REC-HEAD = {YES|NO} ]
[ RECTYPE = {CMDLOG |IOLOG} ]  /* record type
[ USER-EXIT = module_name ]
[ WHERE logical-expression ] /* global selection

```

The following options are available only with batch Unicenter CA-SpaceMan (SPMNENSU):

```

[ DEVICE = device type ]
[ LP = number-of-blocks ]    /* size of Work Part 1
[ LWKP2 = number-of-blocks ] /* size of Work Part 2

```

Example

```

GLOBALS PAGESIZE=23
IOLOG-SWITCH=ON
CPU-ID=SYSA
DBID=2 DBNAME=PRODUCTION-DB;

```

Parameters

Selected parameters of the GLOBALS statement are described below. Each parameter is optional unless otherwise stated.

DEVICE

This must be the same as the device type specified in ADARUN cards for Adabas execution. The DEVICE parameter, together with the LP and LWKP2 parameters, is required by Unicenter CA-SpaceMan to build its RABN-RANGE table. The Unicenter CA-APAS RABN-RANGE parameter should not be used because Unicenter CA-SpaceMan builds its own RABN-RANGE table.

IOLOG-SWITCH

Specifies whether I/O Log records are to be created in DBGIOR5. If the default value ON is used, DBGIOR5 passes I/O Log records to the Data Collector, providing there are active IOLOG requests. If the value is set to OFF, it does not. The IOLOG-SWITCH can also be controlled dynamically with Unicenter CA-APAS Insight using the IOLOGSW command.

When IOLOG-SWITCH is set to OFF, DBGIOR5 performs no work and thus imposes no overhead on ADAIOR. This allows you to limit sampling overhead on ADAIOR to only the times you want I/O data. To do this you must:

- Install DBGIOR5
- Set IOLOG-SWITCH to OFF in the GLOBALS statement
- Type IOLOGSW on the Unicenter CA-APAS Insight command line and ON or OFF on the request line to start or stop I/O Log record collection

LP

This must be the same as the LP parameter specified in ADARUN cards for the MPM execution. If this parameter is omitted, the LP value of the most recent MPM session is obtained from the Work data set. If specified, the LP parameter overrides the corresponding Adabas parameter when Unicenter CA-SpaceMan builds its RABN-RANGE table. The Unicenter CA-APAS RABN-RANGE parameter should not be used because Unicenter CA-SpaceMan builds its own RABN-RANGE table.

LWKP2

If your installation uses the LWKP2 parameter to define the size of Work Part 2, the value should be entered with this parameter. If specified, the LWKP2 parameter overrides the corresponding Adabas parameters when Unicenter CA-SpaceMan builds its RABN-RANGE table. The Unicenter CA-APAS RABN-RANGE parameter should not be used because Unicenter CA-SpaceMan builds its own RABN-RANGE table.

RECTYPE

Defines the default record-type for all requests. If this parameter is not specified, batch Unicenter CA-APAS defaults to CMDLOG processing. Batch Unicenter CA-SpaceMan (SPMNENSU) defaults to IOLOG processing.

DECLARE: The Declare A User-Defined Field

The DECLARE statement defines a new field and the means of computing its value from existing fields. Two forms of definition are supported: a static redefinition of an existing field and an assignment of values based on logical criteria.

To redefine or reference another user-defined field, the redefined or referenced field must be declared first. In other words, a field referenced in a DECLARE must have already been declared. The DECLARE statement does not require a statement label.

Bear the following in mind when using DECLARE statements:

- Declared fields are global in nature; once defined in a single DECLARE statement, a field may be referenced in any subsequent statements.
- DECLARE statements may be added but not deleted from an Unicenter CA-APAS terminal. Once in effect, the field definition from a DECLARE continues to be available through the rest of the current execution of the Data Collector.
- A given field must not be declared in more than one DECLARE statement in any given execution of the Data Collector.

Syntax

```

DECLARE
[ CMDLOG | IOLOG ]
field_name ( {A|B|H|N|P} length [.decimal_places [/scale] ]
            [ HD 'header_string[,...]' ]
            )
{ REDEFINES field_name [(length [,offset])]
| VALOF
  IF logical_expression
  [ ELSE {conditional_statement | RESULTIS constant} ]
  [ ELSEIF {conditional_statement | RESULTIS constant} ]
  [ ELSEIF {conditional_statement | RESULTIS constant} ]... ]
  [ THEN {statement | RESULTIS constant} ]
  IFEND
}

```

Example

```
DECLARE IOLOG DSTYPE(A1) REDEFINES DATASET;
```

Parameters

CMDLOG | IOLOG

The record type, either CMDLOG or IOLOG, declares to which type of input record and request a declared field is applicable.

If no record-type is specified or if CMDLOG is coded, the field may be used only in EXTRACT and SUMMARY requests processing CMDLOG fields. If IOLOG is specified, the field may be used only in processing Unicenter CA-SpaceMan I/O Log records.

EXTRACT: Report/Output Individual Command Data

The EXTRACT statement requests detailed output from individual ADAIOR commands for exception reporting, debugging, or other purposes. As many EXTRACT statements as desired may be used in a given execution of the Data Collector.

EXTRACT statements may be added, deleted, paused, and resumed from Unicenter CA-APAS Insight terminals. However, appropriate JCL for the ddnames must have been included when starting the Adabas nucleus session. Review the discussion of Output files in *Unicenter CA-APAS Systems Guide* before using the EXTRACT statement.

The EXTRACT statement specifies the following:

- Output fields and their report/output format
- Report titles
- Selection criteria for accepting commands to be processed
- Output destinations

Syntax

```
label: EXTRACT
  [ ( format_parameter_list ) ]
  [ CMDLOG | IOLOG ]
  extract field list
  [ INSIGHT-LINES = number ]
  [ OUTPUT-EXIT = module_name ]
  [ OUTPUT-FILE {ddname}(DUAL,ddname) ]
  [ OUTPUT-INSIGHT-LINES = {YES|NO} ]
  [ OWNER = {userid|'*'} ]
  [ PRINT TO {ddname}(DUAL,ddname) ]
  [ PRINT-INSIGHT-LINES = {YES|NO} ]
  [ SECURITY = {DISPLAY|MODIFY|NONE} ]
  [ TITLE = 'character_string' ]
  [ WHERE logical_expression ]
;
```

Example

```
IOTRACE: EXTRACT (MAXLINES=300) IOLOG
IOR-FUNC FUNCTION R-W
FILE COMPONENT EXTENT CMD DESCRIPTOR DURATION HEAD-TRAVEL
VOLSER CYLINDER PRINT-LINE(PF=00)
TITLE='TRACE OF ADAIOR ACTIVITY'
PRINT TO GENPRT;
```

Parameters

Selected parameters of the EXTRACT statement are described below.

Label

A label is required and must be unique. It is written both to the report generated and in the record header on the output file.

CMDLOG | IOLOG

The record type, either CMDLOG or IOLOG, specifies the type of log record that the request processes. The fields named in the request must all apply to this record type. For more information about IOLOG data fields, see the chapter “IOLOG Data Fields” in this manual. If no record type is specified, CMDLOG, the default, is applied.

OUTPUT-EXIT

An output-exit may be specified in addition to or instead of an output file. All output records generated by the request are passed to the specified module. Standard linkage conventions are used. On entry to the module, Register 15 contains the entry point of the module, Register 14 contains the return address, Register 13 points to a standard 18-word OS save area which can be used by the module. Register 1 points to a two-word parameter list:

- Offset 0: Address of the output record. It is set to 0 on the final call.
- Offset 4: Address of a word that may be used by the exit module. The contents of this word is preserved between calls to the exit module, so it can be used to store the address of a work area for the exit module. The value of the word is 0 on the first call to the exit module. The exit is called once for each output record and a final time when Unicenter CA-APAS ends or the MPM shuts down.

A return code may be inserted into Register 15 before returning from the exit module. A return code of 16 (decimal) bypasses writing of the output record to the output file. Any other value results in the output record being written if an OUTPUT-FILE was specified.

OUTPUT-INSIGHT-LINES

When you specify an INSIGHT-LINES parameter and either an OUTPUT-FILE or OUTPUT-EXIT parameter or both, the OUTPUT-INSIGHT-LINES parameter specifies how the Data Collector processes a DISPLAY command from Unicenter CA-APAS for this request. The default value NO specifies that the record generated for the DISPLAY command are not written to the OUTPUT-FILE data set or passed to the OUTPUT-EXIT.

A value of YES instructs Unicenter CA-APAS to write a record to the OUTPUT-FILE data set and/or passed to the OUTPUT-EXIT whenever a Unicenter CA-APAS DISPLAY command is processed for this request. Note that this parameter has no effect on calls to the OUTPUT-EXIT or records written to the OUTPUT-FILE data set at the end of an interval or at the end of an MPM session.

In addition, setting the value of this parameter to YES has no meaning and is treated as a syntax error when the request does not contain an INSIGHT-LINES and either or both the OUTPUT-FILE or OUTPUT-EXIT parameters.

OWNER

The OWNER parameter specifies the 8-character identifier of the owner of the request. It is used to control who is authorized to use Unicenter CA-APAS commands to modify the request, that is, to use the DELETE, PAUSE, RESUME, or RESET commands against the request, or to issue a DISPLAY command to display its INSIGHT-LINES. Typically, the OWNER value is the userid of the person who creates the request.

An asterisk, '*', which is the default value indicates a public request, which can be accessed by all Unicenter CA-APAS users. Note that the quotation marks must enclose the asterisk, which is not an alphanumeric character. However, the quotation marks are not part of the value.

PRINT-INSIGHT-LINES

When you specify both a PRINT TO parameter and an INSIGHT-LINES parameter, the PRINT-INSIGHT-LINES parameter specifies how the Data Collector processes a DISPLAY command from Unicenter CA-APAS for this request. The default value NO specifies that the lines generated for the DISPLAY command are not written to the PRINT TO data set also.

A value of YES specifies that APAS writes all request data lines to the PRINT TO data set whenever a Unicenter CA-APAS DISPLAY command is processed for this request. Note that this parameter has no effect on the records written to the PRINT TO data set at the end of an interval or at the end of an MPM session. In addition, setting the value of this parameter to YES when the request does not contain both an INSIGHT-LINES and a PRINT TO parameter has no meaning and is treated as a syntax error.

SECURITY

The SECURITY parameter defines the level of request security for the request. See the *Unicenter CA-APAS User Guide* for a complete description of Unicenter CA-APAS request security. The default value, NONE, allows all users complete access to the request. A value of MODIFY allows all users to display data collected by the request but allows only authorized user to modify it, that is, to use Unicenter CA-APAS DELETE, PAUSE, RESUME, or RESET commands against this request. A value of DISPLAY allows only authorized users to display or modify the request.

SUMMARIZE: Report/Output Interval and Total Summary Data

The SUMMARIZE statement summarizes information over multiple ADAIOR commands. Summarization is based on control break fields. If an INTERVAL parameter is supplied, summary tables are output at breaks of the specified time interval. If no INTERVAL parameter is coded, a summary table is output at the end of processing. Any number of SUMMARIZE requests may be included in an execution of the Data Collector.

SUMMARIZE statements may be added, deleted, paused, and resumed from Unicenter CA-APAS terminals. However, appropriate JCL for the ddnames must have been included when starting the Adabas nucleus session. Review the discussion of Output files in *Unicenter CA-APAS Systems Guide* before using the SUMMARIZE statement.

The SUMMARIZE statement specifies the following:

- The fields which are to be summarized and their report/output formats
- The control break fields for summarization
- Any special sort order for the report
- Optionally, the periodic time interval at which summary reports are to be created
- Titles for the report
- Selection criteria for including command records in the summarization
- Output destinations

Syntax

```
label: SUMMARIZE
  [ ( format_parameter_list ) ]
  [ CMDLOG | IOLOG ]
  summary_function_list
  [ BY control_break_field [ (BY number) ]
    [ BY control_break_field [ (BY number) ] ]... ]
  [ INSIGHT-LINES = number ]
  [ INTERVAL = duration_value ]
  [ OUTPUT-EXIT = module_name ]
  [ OUTPUT-FILE {ddname| (DUAL, ddname)} ]
  [ OUTPUT-INSIGHT-LINES = {YES|NO} ]
  [ OWNER = {userid|'*'} ]
  [ PRINT TO {ddname| (DUAL, ddname)} ]
  [ PRINT-INSIGHT-LINES = {YES|NO} ]
  [ SECURITY = {DISPLAY|MODIFY|NONE} ]
  [ TITLE = 'character_string' ]
  [ WHERE logical-expression ]
;
```

Example

```
IOSUMINS: SUMMARIZE (SUBTOT=OFF) IOLOG
COUNT (HD='I/O', 'COUNT') SUM(DURATION) MEAN(DURATION)
BY FILE BY COMPONENT BY R-W
WHERE IOR-FUNC <= H'0A'
PRINT TO GENPRT;
```

Parameters

Selected parameters of the SUMMARIZE request are described below.

label

The label specified for the SUMMARIZE request must be unique. It is written both to the report generated from the request and in the record header on the output file.

CMDLOG | IOLOG

The record type, CMDLOG or IOLOG, specifies the type of log record the request processes. The fields named in the request must all apply to this record type. For more information about IOLOG data fields, see the chapter “IOLOG Data Fields” in this manual. If no record type is specified, CMDLOG, the default, is applied.

BY *control_break_field*

A control break field is specified after the keyword BY. Several control break fields can be listed from major to minor break level. These fields are always printed in front of the list of summary functions. Field-formats may be given for the fields, but slashes are not allowed. Report subtotals and subtotal records for each specified break level are automatically generated, unless suppressed by SUBTOT=OFF or a SORT specification.

SUBTOT=OFF in the format parameter list immediately following the SUMMARIZE keyword suppresses all levels of subtotals. SUBTOT=OFF may be used following one or more of the control break fields to selectively suppress individual levels of subtotalling.

(BY *number*)

The (BY *number*) parameter is applied only to numeric control-break fields. It allows grouping of numeric values by rounding them down to the next lower multiple of the number specified.

Format

BY *control_break_field* (BY *number*)

Syntax

```
IOCYL: SUMMARIZE IOLOG COUNT (HD='I/O', 'COUNT')
BY VOLUME BY CYLINDER (BY 50)
PRINT TO GENPRT;
```

This reports I/O by CYLINDER ranges 0-49, 50-99, etc.

OUTPUT-EXIT

An output-exit may be specified in addition to or instead of an output file. All output records generated by the request are passed to the specified module. Standard linkage conventions are used. On entry to the module, Register 15 contains the entry point of the module, Register 14 contains the return address, Register 13 points to a standard 18-word OS save area which can be used by the module. Register 1 points to a two-word parameter list:

- Offset 0: Address of the output record. It is set to 0 on the final call.
- Offset 4: Address of a word that may be used by the exit module. The contents of this word is preserved between calls to the exit module, so it can be used to store the address of a work area for the exit module. The value of the word is 0 on the first call to the exit module.

The exit is called once for each output record and a final time when Unicenter CA-APAS ends or the MPM shuts down.

A return code may be inserted into Register 15 before returning from the exit module. A return code of 16 (decimal) bypasses writing of the output record to the output file. Any other value results in the output record being written if an OUTPUT-FILE was specified.

OUTPUT-INSIGHT-LINES

When you specify an INSIGHT-LINES parameter and either an OUTPUT-FILE or OUTPUT-EXIT parameter or both, the OUTPUT-INSIGHT-LINES parameter specifies how the Data Collector processes a DISPLAY command from Unicenter CA-APAS for this request. The default value NO specifies that the record generated for the DISPLAY command are not written to the OUTPUT-FILE data set or passed to the OUTPUT-EXIT.

A value of YES specifies that Unicenter CA-APAS writes a record to the OUTPUT-FILE data set and/or passed to the OUTPUT-EXIT whenever an Unicenter CA-APAS DISPLAY command is processed for this request.

Note that this parameter has no effect on calls to the OUTPUT-EXIT or records written to the OUTPUT-FILE data set at the end of an interval or at the end of an MPM session. In addition, setting the value of this parameter to YES has no meaning and is treated as a syntax error if the request does not contain an INSIGHT-LINES and either or both the OUTPUT-FILE or OUTPUT-EXIT parameters.

OWNER

The OWNER parameter specifies the 8-character identifier of the owner of the request. It is used to control who is authorized to use Unicenter CA-APAS commands to modify the request, that is, to use the DELETE, PAUSE, RESUME, or RESET commands against the request, or to issue a DISPLAY command to display its INSIGHT-LINES. Typically, the OWNER value is the userid of the person who creates the request.

An asterisk, '*', which is the default value indicates a public request, which can be accessed by all Unicenter CA-APAS users. Note that the quotation marks must enclose the asterisk, which is not an alphanumeric character. However, the quotation marks are not part of the value.

PRINT-INSIGHT-LINES

When you specify both a PRINT TO parameter and an INSIGHT-LINES parameter, the PRINT-INSIGHT-LINES parameter specifies how the Data Collector processes a DISPLAY command from Unicenter CA-APAS for this request. The default value NO specifies that the lines generated for the DISPLAY command are not written to the PRINT TO data set also.

A value of YES specifies that Unicenter CA-APAS writes all request data lines to the PRINT TO data set whenever a Unicenter CA-APAS DISPLAY command is processed for this request. Note that this parameter has no effect on the records written to the PRINT TO data set at the end of an interval or at the end of an MPM session. In addition, setting the value of this parameter to YES when the request does not contain both an INSIGHT-LINES and a PRINT TO parameter has no meaning and is treated as a syntax error.

SECURITY

The SECURITY parameter defines the level of Unicenter CA-APAS request security for the request. See the *Unicenter CA-APAS User Guide* for a complete description of Unicenter CA-APAS request security. The default value, NONE, allows all users complete access to the request. A value of MODIFY allows all users to display data collected by the request but allows only authorized user to modify it, that is, to use Unicenter CA-APAS DELETE, PAUSE, RESUME, or RESET commands against this request. A value of DISPLAY allows only authorized users to display or modify the request.

COPY: Copy Command Log Records for Later Batch Processing

The COPY statement writes Command Log records to a sequential file for later batch processing by the Data Collector. As many COPY statements as desired may be coded for a given execution of the Data Collector.

COPY statements may be added, deleted, paused, and resumed from Unicenter CA-APAS Insight terminals. However, appropriate JCL for the ddnames must have been included when starting the Adabas nucleus session. Review the discussion of COPY files in *Unicenter CA-APAS Systems Guide* before using the COPY statement.

The COPY statement specifies the following:

- Selection criteria for command records to be output
- The ddname for sequential file output

Syntax

```
label: COPY  
[ CMDLOG | IOLOG ]  
[ Buffer_List ]  
OUTPUT-FILE {ddname|(DUAL,ddname )}  
[ OUTPUT-EXIT = module_name ]  
[ OWNER = {userid | '*' } ]  
[ SECURITY = {DISPLAY|MODIFY|NONE} ]  
[ WHERE logical-expression ]  
;
```

Example

```
COPYVOL: COPY IOLOG  
WHERE VOLSER = 'ADA001'  
OUTPUT-FILE = (DUAL,DCCOPY1) ; /* X/Y file
```

Parameters

Selected parameters of the COPY statement are described below.

label

The *label* specified for the COPY request must be unique. It may be referenced from Unicenter CA-APAS to ADD, PAUSE, RESUME and DELETE the request.

CMDLOG | IOLOG

The record type, CMDLOG or IOLOG, specifies the type of log record the request processes. The fields named in the requests must all apply to this record type. For more information about IOLOG data fields, see the chapter "IOLOG Data Fields" in this manual. If no record type is specified the default, CMDLOG, is applied.

OWNER

The OWNER parameter specifies the 8-character identifier of the owner of the request. It is used to control who is authorized to use Unicenter CA-APAS commands to modify the request, that is, to apply the DELETE, PAUSE, RESUME, or RESET commands to the request or to issue a DISPLAY command to display its INSIGHT-LINES. Typically, the OWNER value is the userid of the person who creates the request.

An asterisk, '*', which is the default value indicates a public request, which can be accessed by all Unicenter CA-APAS users. Note that the quotation marks must enclose the asterisk, which is not an alphanumeric character. However, the quotation marks are not part of the value.

SECURITY

The SECURITY parameter defines the level of Unicenter CA-APAS request security for the request. The default value, NONE, allows all users complete access to the request. The value MODIFY allows all users to display data collected by the request but allows only authorized users to modify it by apply the DELETE, PAUSE, RESUME or RESET commands to the request. The value DISPLAY allows only authorized users to display or modify the request.

See the *Unicenter CA-APAS User Guide* for a complete description of request security.

IOLOG Data Fields

This chapter describes the fields that can be reported when the Unicenter CA-SpaceMan File Management for Adabas (Unicenter CA-SpaceMan) IOLOG capture facility has been installed. These fields can be reported either interactively with Unicenter CA-APAS Insight Monitor for Adabas (Unicenter CA-APAS) or with a batch execution of the SPMNENSU program processing an IOLOG. These fields cannot be reported together with Command Log fields within a given request; each request can reference the fields of only one of its record types. The RECORD-TYPE must be set either explicitly or by default to IOLOG to access these fields.

CMDLOG Fields

The following DATE, TIME and general summary fields are available as defined in the CMDLOG field definitions.

CMD-RATE	Rate of ADAIOR calls
COUNT	Count of calls
DATE	
DATN	
DATE-TIME	
DATN-TIME	
DATEJ	
DAT4J	
DAY	
HOUR	
MONTH	
PERCENT	Percent of calls
QUARTER-HOUR	
TIME	

WEEK
WEEKDAY

IOLOG Data Fields

Field	Description												
CMD	See COMMAND-CODE.												
CMD-SEQ-NO	See IORCMDSQ.												
COMMAND-CODE	<p>The two-character Adabas command code of the command that caused the I/O if the command was issued from an Adabas thread.</p> <table border="1"> <thead> <tr> <th>File</th> <th>Rept</th> <th>Sum</th> <th>Acc</th> </tr> <tr> <th>Fmt</th> <th>Fmt</th> <th>Fld</th> <th>Size</th> </tr> </thead> <tbody> <tr> <td>A2</td> <td>A2</td> <td>N</td> <td></td> </tr> </tbody> </table> <p>Alias CMD</p> <p>Column Heading CMD</p>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	A2	A2	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
A2	A2	N											

Field	Description
COMPONENT	<p>File component being accessed on Data or Associator. COMPONENT differentiates between different parts of the Associator data set such as Address Converter, Upper Index, etc.</p> <p>File Rept Sum Acc Fmt Fmt Fld Size A2 A2 N</p> <p>Values</p> <ul style="list-style-type: none"> AC – Address Converter AL – Alternate Associator Blocks CL – Command Log DA – Data Storage DL – Alternate Data Blocks DS – Data Storage Space Table FC – File Control Block FD – Field Description Table GC – General Control Block NI – Normal Index PL – Protection Log PT – PPT SF – Delta Save UA – Unused Associator UD – Unused Data UI – Upper Index UT – Utility Blocks (RABN 2-30) W1 – Work Part 1 W2 – Work Part 2 W3 – Work Part 3 W4 – Work Part 4 <p>Column Heading FILE/COMPT</p>

Field	Description												
CYLINDER	<p>Cylinder number accessed by I/O, that is, the cylinder number on the volume where the RABN is located that is being accessed.</p> <table><thead><tr><th>File</th><th>Rept</th><th>Sum</th><th>Acc</th></tr><tr><th>Fmt</th><th>Fmt</th><th>Fld</th><th>Size</th></tr></thead><tbody><tr><td>B2</td><td>N4</td><td>N</td><td></td></tr></tbody></table> <p>Column Heading CYLN</p>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	B2	N4	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
B2	N4	N											
DATASET	<p>The ASSO, DATA, WORK, CLOG, or PLOG data set being accessed by ADAIOR call.</p> <table><thead><tr><th>File</th><th>Rept</th><th>Sum</th><th>Acc</th></tr><tr><th>Fmt</th><th>Fmt</th><th>Fld</th><th>Size</th></tr></thead><tbody><tr><td>A4</td><td>A4</td><td>N</td><td></td></tr></tbody></table> <p>Usage</p> <p>Same field as FUNCTION, but indicative of ASSO, DATA, WORK, CLOG, PLOG, SORT, TEMP values.</p> <p>Column Heading FUNCT</p>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	A4	A4	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
A4	A4	N											
DESCRIPTOR	<p>The two-character Adabas field name of the index being accessed by Adabas search commands during index access.</p> <table><thead><tr><th>File</th><th>Rept</th><th>Sum</th><th>Acc</th></tr><tr><th>Fmt</th><th>Fmt</th><th>Fld</th><th>Size</th></tr></thead><tbody><tr><td>A2</td><td>A2</td><td>N</td><td></td></tr></tbody></table> <p>Usage</p> <p>When Adabas accesses a particular index while executing an Adabas command, this field will contain the descriptor name.</p> <p>Column Heading DESC</p>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	A2	A2	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
A2	A2	N											
DUR	See DURATION												

Field	Description
DURATION	<p>Time to execute ADAIOR call, that is, the time from initiation of the ADAIOR function until ADAIOR returns. This is the time required to complete the I/O if this is a synchronous I/O call; otherwise it only includes the time required to initiate the I/O.</p> <p>File Rept Sum Acc Fmt Fmt Fld Size B4 N7 Y F</p> <p>Values</p> <p>Expressed in seconds.</p> <p>Usage</p> <p>The total time to execute all I/O is the duration of all I/O calls plus the duration of the following WAIT function calls, when the I/O is asynchronous.</p> <p>Alias DUR</p> <p>Column Heading DURAT/SECS</p>
EXTENT	<p>The Adabas extent of the component being accessed. Most Adabas components can have up to five extents allocated to them on DATA or ASSO. May also distinguish between CLOG R1/R2 and PLOG R1/R2.</p> <p>File Rept Sum Acc Fmt Fmt Fld Size A1 A1 N</p> <p>Column Heading COMP/EXT</p>
FILE	<p>Adabas file number, that is, the file number of the Adabas file which was being accessed by the call.</p> <p>File Rept Sum Acc Fmt Fmt Fld Size B2 N5 N</p> <p>Column Heading FILE</p>

Field	Description												
FUNCTION	<p>ADAIOR function in readable form. In general you will be interested in the READ/WRITE calls to the data sets, but data on all functions is available.</p>												
	<table border="1"> <thead> <tr> <th data-bbox="574 443 634 470">File</th> <th data-bbox="634 443 695 470">Rept</th> <th data-bbox="695 443 755 470">Sum</th> <th data-bbox="755 443 813 470">Acc</th> </tr> <tr> <th data-bbox="574 470 634 497">Fmt</th> <th data-bbox="634 470 695 497">Fmt</th> <th data-bbox="695 470 755 497">Fld</th> <th data-bbox="755 470 813 497">Size</th> </tr> </thead> <tbody> <tr> <td data-bbox="574 497 634 520">A4</td> <td data-bbox="634 497 695 520">A4</td> <td data-bbox="695 497 755 520">N</td> <td></td> </tr> </tbody> </table>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	A4	A4	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
A4	A4	N											
	<p>Together with R-W gives a readable translation of IOR-FUNC. What follows under the heading Values is a partial list of functions. Refer to the <i>Adabas Internals Manual</i> for a complete list.</p>												
	Values												
	8IOS – return up to eight completed I/O sequence numbers												
	ASSO – access to Associator data set												
	CARD – read a control card												
	CLOG – read/write to the Command Log												
	CPUI – CPU timing initialized												
	CPUU – CPU time used												
	DATA – access to the Data data set												
	DATE – get the date												
	DEQ – issue an OS dequeue, IOR-DATA gives value												
	ENQ – issue an OS enqueue IOR-DATA gives value												
	FREM – freemain storage												
	GETM – getmain storage												
	JOBN – return job name												
	PLOG – read/write to the protection log												
	PRNT – print a line, PRINT-LINE contains the line												
	SEQF – I/O to a sequential file, IOR-CTYPE tells which one												
	SORT – access to the Sort data set (utilities only)												
	STCK – get the STCK time stamp												
	TEMP – access to the Temp data set (utilities only)												
	WAIT – wait for completion of an I/O sequence number in IOR-SEQ-NO												

Field	Description
	<p>WORK— access to the Work data set</p> <p>WTO— write to operator, PRINT-LINE contains the message</p> <p>WTOR— write to operator with reply, PRINT-LINE contains the message</p> <p>Column Heading FUNCT</p>
HEAD-TRAVEL	<p>Head movement caused by this I/O, that is, the number of cylinders across which the head has to move to execute this I/O.</p> <p>File Rept Sum Acc Fmt Fmt Fld Size B4 N4 Y F</p> <p>Derivation</p> <p>This is computed by maintaining a record of the last cylinder accessed on the volume by Adabas. This value does not take into account access to non-Adabas files on the same volume.</p> <p>Column Heading HEAD/TRAVEL</p>
IO-COMPLETED	<p>Sequence number of I/O that has completed. This is the I/O sequence number of completed asynchronous disk I/O. Every asynchronous I/O call returns the sequence number of a previous call for which I/O has completed, avoiding redundant wait calls.</p> <p>File Rept Sum Acc Fmt Fmt Fld Size B4 N8 N</p> <p>Usage</p> <p>The actual time to complete a disk I/O for an asynchronous call is the time from the initiation of the I/O until it is reported as completed, either in this field during a later call or following a WAIT function for the I/O. This field is thus necessary for tracing of detailed I/O sequences.</p> <p>Column Heading IO-SEQ-NO/COMPLETED</p>

Field	Description												
IO-SEQ-NO	<p>Internal ADAIOR I/O sequence number. ADAIOR assigns a unique sequence number to each I/O that it initiates. This number is used to report the completion of asynchronous I/O.</p>												
	<table border="0"> <tr> <td>File</td> <td>Rept</td> <td>Sum</td> <td>Acc</td> </tr> <tr> <td>Fmt</td> <td>Fmt</td> <td>Fld</td> <td>Size</td> </tr> <tr> <td>B4</td> <td>N8</td> <td>N</td> <td></td> </tr> </table>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	B4	N8	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
B4	N8	N											
	Values												
	<p>Note that this number is increased by 2 on every call and is approximately twice the count of total I/O.</p>												
	Column Heading												
	IO-SEQ-NO												
IO-TYPE	<p>Synchronous vs asynchronous I/O indicator.</p>												
	<table border="0"> <tr> <td>File</td> <td>Rept</td> <td>Sum</td> <td>Acc</td> </tr> <tr> <td>Fmt</td> <td>Fmt</td> <td>Fld</td> <td>Size</td> </tr> <tr> <td>A1</td> <td>A1</td> <td>N</td> <td></td> </tr> </table>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	A1	A1	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
A1	A1	N											
	<p>Indicates whether the ADAIOR read or write function should wait for completion of the I/O or return to Adabas immediately.</p>												
	Values												
	<p>W – wait for completion of the I/O; the duration reflects the total time to read the record.</p>												
	<p>A – asynchronous I/O; return to the caller as soon as the I/O is initiated; IO-COMPLETED may indicate a previous asynchronous I/O which has completed.</p>												
	Column Heading												
	W/A												

Field	Description												
IORCMDSQ	<p>Command Log sequence number. This field reports the sequence number of the Adabas command responsible for an I/O.</p> <table border="1"> <thead> <tr> <th>File</th> <th>Rept</th> <th>Sum</th> <th>Acc</th> </tr> <tr> <th>Fmt</th> <th>Fmt</th> <th>Fld</th> <th>Size</th> </tr> </thead> <tbody> <tr> <td>B4</td> <td>N8</td> <td>N</td> <td></td> </tr> </tbody> </table> <p>Usage</p> <p>Use this field to relate commands to their resulting I/Os.</p> <p>The Command Log sequence number reported in the number assigned by Adabas when the command is placed in the command queue and not the sequential number assigned as records are written to the Command Log file. There will be gaps in the sequence, for example, in the ALLTRACE request. The missing numbers are sequence numbers assigned to internal or utility commands that are never written to the Command Log file.</p> <p>Alias CMDSEQ/CMD-SEQ-NO</p> <p>Column Heading CMD-SEQ-NO</p>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	B4	N8	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
B4	N8	N											
IOR-CTYPE	<p>This is a sub-function indicator for the ADAIOR call. Some documentation of values can be found in the <i>Adabas Internals Manual</i>.</p> <table border="1"> <thead> <tr> <th>File</th> <th>Rept</th> <th>Sum</th> <th>Acc</th> </tr> <tr> <th>Fmt</th> <th>Fmt</th> <th>Fld</th> <th>Size</th> </tr> </thead> <tbody> <tr> <td>A1</td> <td>A1</td> <td>N</td> <td></td> </tr> </tbody> </table> <p>Usage</p> <p>Distinguishes synchronous from asynchronous I/O.</p> <p>Values</p> <ul style="list-style-type: none"> C—synchronous I/O M—asynchronous I/O <p>Column Heading CALL/TYPE</p>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	A1	A1	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
A1	A1	N											

Field	Description												
IOR-DATA	<p>Textual data for ADAIOR call. This is the textual data for ENQ/DEQ, PRNT and WTO/WTOR calls.</p> <table border="1"> <thead> <tr> <th>File</th> <th>Rept</th> <th>Sum</th> <th>Acc</th> </tr> <tr> <th>Fmt</th> <th>Fmt</th> <th>Fld</th> <th>Size</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>O</td> <td>N</td> <td></td> </tr> </tbody> </table> <p>Alias PRINT-LINE</p> <p>Column Heading PRINT-LINE</p>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	A	O	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
A	O	N											
IOR-FUNC	<p>Internal ADAIOR function. This field is documented in the <i>Adabas Internals Manual</i>. It is translated to a readable function in the FUNCTION field.</p> <table border="1"> <thead> <tr> <th>File</th> <th>Rept</th> <th>Sum</th> <th>Acc</th> </tr> <tr> <th>Fmt</th> <th>Fmt</th> <th>Fld</th> <th>Size</th> </tr> </thead> <tbody> <tr> <td>B1</td> <td>H2</td> <td>N</td> <td></td> </tr> </tbody> </table> <p>Usage</p> <p>To trace the sequence of ADAIOR calls.</p> <p>Column Heading IOR/FUNC</p>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	B1	H2	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
B1	H2	N											
IOR-PROGRAM	<p>Internal Adabas program-id. This field is the internal Adabas program number for the Adabas load module that called ADAIOR. A partial list of these IDs can be found in the documentation of the SYNC checkpoint entries in the Adabas documentation of ADAREP.</p> <table border="1"> <thead> <tr> <th>File</th> <th>Rept</th> <th>Sum</th> <th>Acc</th> </tr> <tr> <th>Fmt</th> <th>Fmt</th> <th>Fld</th> <th>Size</th> </tr> </thead> <tbody> <tr> <td>B1</td> <td>H2</td> <td>N</td> <td></td> </tr> </tbody> </table> <p>Column Heading IOR/PN</p>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	B1	H2	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
B1	H2	N											

Field	Description												
IOR-RESP	<p>Internal ADAIOR response code. The response code returned by ADAIOR to the caller.</p> <table> <thead> <tr> <th>File</th> <th>Rept</th> <th>Sum</th> <th>Acc</th> </tr> <tr> <th>Fmt</th> <th>Fmt</th> <th>Fld</th> <th>Size</th> </tr> </thead> <tbody> <tr> <td>B1</td> <td>H2</td> <td>N</td> <td></td> </tr> </tbody> </table> <p>Values</p> <p>Documented in the <i>Adabas Internals Manual</i>.</p> <p>Column Heading IOR/RSP</p>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	B1	H2	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
B1	H2	N											
PRINT-LINE	<p>The PRINT-LINE field applies to the ADAIOR PRINT, WTO and WTOR functions and contains the textual data written to the DDPRINT data set from Adabas for ADAIOR call. Same as IOR-DATA. The line being printed to DDPRINT or DDDRUCK data set.</p> <table> <thead> <tr> <th>File</th> <th>Rept</th> <th>Sum</th> <th>Acc</th> </tr> <tr> <th>Fmt</th> <th>Fmt</th> <th>Fld</th> <th>Size</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>O</td> <td>N</td> <td></td> </tr> </tbody> </table> <p>Column Heading PRINT-LINE</p>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	A	O	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
A	O	N											
RABN	<p>Relative Adabas Block Number of the block being accessed. Relative Adabas Block Number being accessed by the call.</p> <table> <thead> <tr> <th>File</th> <th>Rept</th> <th>Sum</th> <th>Acc</th> </tr> <tr> <th>Fmt</th> <th>Fmt</th> <th>Fld</th> <th>Size</th> </tr> </thead> <tbody> <tr> <td>B4</td> <td>N11</td> <td>N</td> <td></td> </tr> </tbody> </table> <p>Column Heading RABN</p>	File	Rept	Sum	Acc	Fmt	Fmt	Fld	Size	B4	N11	N	
File	Rept	Sum	Acc										
Fmt	Fmt	Fld	Size										
B4	N11	N											

Field	Description
R-W	<p>Read/Write/Open/Close indicator. Type of access performed. Use together with DATASET field.</p> <p>File Rept Sum Acc Fmt Fmt Fld Size A1 A1 N</p> <p>Values</p> <p>C – Close</p> <p>O – Open</p> <p>R – Read</p> <p>W – Write</p> <p>Column Heading R/W</p>
STCK	<p>Store Clock Time I/O Call Completed. Time of completion of the ADAIOR function.</p> <p>File Rept Sum Acc Fmt Fmt Fld Size B4 H8 N</p> <p>Column Heading STCK/TIME</p>
VOLSER	<p>Volume number of disk being accessed.</p> <p>File Rept Sum Acc Fmt Fmt Fld Size A6 A6 N</p> <p>Alias VOLUME</p> <p>Column Heading VOLSER</p>
VOLUME	See VOLSER

Unicenter CA-SpaceMan Batch Messages

This chapter details messages issued by Unicenter CA-SpaceMan File Management for Adabas (Unicenter CA-SpaceMan) batch facilities.

All messages have the following format.

DBG*n*C YYYY-MM-DD HH:MM:SS text

where *n* is a five-digit number and *C* is one of the following letters:

I – Informational

W – Warning

E – Error

Unicenter CA-SpaceMan Batch Messages

DBG00001I CONTROL CARD INPUT

Explanation The Unicenter CA-SpaceMan control card input from data set DBGIN is listed. Processing continues.

Action None required.

DBG00002I SYNTAX PROCESSING SUCCESSFULLY COMPLETED

Explanation Syntax checking of the user-supplied control cards has been completed and no syntax errors were found. Processing continues.

Action None required.

DBG00003E ERROR DETECTED IN CA-SPACEMAN INPUT, PROCESSING TERMINATED

Explanation Processing was terminated due to one or more control card syntax errors. The program terminates with a return code 12.

Action Resubmit the job after correcting the errors.

DBG00004I THE CA-SPACEMAN DATABASE REPORT HAS COMPLETED WITH RETURN CODE *n*

Explanation Unicenter CA-SpaceMan has completed with the indicated return code. Processing terminates.

Action If the return code is other than zero, check the preceding messages for any errors.

DBG00005E THE CA-SPACEMAN OS ALLOCATION UTILITY HAS COMPLETED WITH RETURN CODE *n*

Explanation The Unicenter CA-SpaceMan OS Allocation Utility has completed with the indicated return code. Processing terminates.

Action If the return code is other than zero, check preceding messages for any errors.

DBG00006I THE CA-SPACEMAN UPDATE UTILITY HAS COMPLETED WITH RETURN CODE *n*

Explanation The Unicenter CA-SpaceMan Update Utility has completed with the indicated return code. Processing terminates.

Action If the return code is other than zero, check the preceding messages for any errors.

DBG01001E IDENTIFIER IS LONGER THAN 32 CHARACTERS

Explanation Keyword values longer than 32 characters are not allowed. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.

Action Correct the error and resubmit the request.

DBG01002E STRING IS LONGER THAN 127 CHARACTERS

Explanation	The maximum allowable length of a string is 127 characters. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.
Action	Correct the error and resubmit the request.

DBG01003E STRING NOT TERMINATED BEFORE END OF LINE

Explanation	Strings cannot be continued past the end of a control statement. Long strings can be generated through concatenation. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.
Action	Correct the error and resubmit the request.

DBG01007E CONTINUATION OF STRING NOT FOUND

Explanation	The control statement indicated that a string continuation should be expected, but none was found. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.
Action	Correct the error and resubmit the request.

DBG01010E NUMBER HAS TOO MANY DIGITS

Explanation	Numbers are limited to 15 digits plus sign and decimal point for decimal numbers. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.
Action	Correct the error and resubmit the request.

DBG01011E FIXED NUMBER IS TOO LARGE

Explanation	Fixed numbers are limited to a maximum value of 2,147,483,647. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.
Action	Correct the error and resubmit the request.

DBG01012E GETSPC FOR *n* BYTES FAILED

Explanation An attempt to obtain *n* bytes of working storage failed due to insufficient memory in the region or partition. The program is terminated or attempts recovery depending on the nature of the request.

Action Increase the size of the region or run in a larger partition.

DBG01013E GETMAIN FOR *n* BYTES FAILED

Explanation This message is the actual amount of space the system was attempting to obtain to satisfy the GETSPC requests.

Action Increase the region size or run in a larger partition.

DBG02001E MISSING STATEMENT IDENTIFIER

Explanation No statement identifier, GLOBALS or REPORT, was found at the start of a control statement sequence. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.

Action Correct the error and resubmit the request.

DBG02002E INVALID STATEMENT IDENTIFIER

Explanation This is an internal error, not a user error. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.

Action Contact Computer Associates with a description of the error.

DBG02003E INVALID PARAMETER IN GLOBALS STATEMENT

Explanation One of the keywords encountered while parsing the GLOBALS statement is not valid. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.

Action Correct the error and resubmit the request.

DBG02004E INVALID PARAMETER IN THE REPORT STATEMENT

Explanation	A parameter supplied on a REPORT statement is not valid. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.
Action	Correct the error and resubmit the request.

DBG02005E FILE MUST BE FIRST PARAMETER IN REPORT STATEMENT

Explanation	The REPORT statement must contain a FILE parameter. Execution is terminated with a condition code of 12 following analysis of the control cards.
Action	Correct the error and resubmit the request.

DBG02006E NO SVC VALUE PROVIDED IN GLOBALS

Explanation	The SVC keyword must be supplied as one of the parameters in a GLOBALS statement before Unicenter CA-SpaceMan batch processing can proceed.
Action	Supply a GLOBALS statement with the SVC number assigned to the Adabas database being accessed.

DBG02013E EXPECTING CHARACTER STRING

Explanation	The parameter requires a character string value. Some other type of symbol was found. Execution is terminated with a condition code of 12 following analysis of the control cards.
Action	Correct the error and resubmit the request.

DBG02021E EXPECTING NUMERIC PARAMETER

Explanation	The parameter requires a numeric value. Some other type of symbol was found. Execution is terminated with a condition code of 12 following analysis of the control cards.
Action	Correct the error and resubmit the request.

DBG02022E MISSING MATCHING PAREN ON PARAMETER

Explanation One or more parentheses have been omitted from the indicated statement. The number of right parentheses must equal the number of left parentheses. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.

Action Correct the error and resubmit the request.

DBG02023E INVALID ALPHABETIC PARAMETER

Explanation The indicated alphabetic parameter is not valid. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.

Action Correct the error and resubmit the request.

DBG02024E INVALID PARAMETER TYPE

Explanation The indicated parameter is not a valid type in the context of the control statement on which it was encountered. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.

Action Correct the error and resubmit the request.

DBG02028E INVALID NUMERIC PARAMETER

Explanation The indicated numeric parameter is not valid. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.

Action Correct the error and resubmit the request.

DBG02029E INVALID ON/OFF PARAMETER

Explanation The indicated ON or OFF parameter is not valid. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.

Action Correct the error and resubmit the request.

DBG02030E INVALID SHORT/LONG PARAMETER

Explanation	The indicated SHORT or LONG parameter is not valid. The message indicates a location following the error in the control statement. Execution is terminated with a condition code of 12 following analysis of the control cards.
Action	Correct the error and resubmit the request.

DBG02035E NO DDNAME FOR FILE x IN JCL

Explanation	No DD or DLBL card was specified corresponding to the filename given in the OUTPUT-FILE keyword for this request. If an error is encountered during initialization, execution terminates with a condition code of 12 following an analysis of the control cards.
Action	Correct the error and resubmit the request.

DBG02051E EXCEEDED ALLOWED SPACE FOR USER DEFINED SYMBOLS

Explanation	Too many user-defined fields have been declared and the allowed space has been exhausted. Execution terminates with a condition code of 12 following an analysis of the control cards.
Action	Correct the error and resubmit the request.

DBG05001E ERROR OPENING ASSOCIATOR *n*; PROBABLY INVALID DEVICE PARAMETER

Explanation	The open of the Associator data set received the indicated response code. Processing terminates.
Action	Verify the DEVICE parameter. Correct the error and resubmit the request.

DBG05002E ADAIOR FUNCTION *x* RECEIVED RESPONSE *y*

Explanation	A call to ADAIOR received the indicated response code. Processing terminates.
Action	Contact Computer Associates technical support.

DBG05003E DATABASE PARAMETER DOES NOT MATCH INTERNAL DATABASE NUMBER

Explanation The DATABASE parameter in the control card does not match the internal DBID. Processing terminates.

Action Verify the DATABASE parameter and the DDASSO statement in your job control language.

DBG05004E FILE NUMBER EXCEEDS 32766

Explanation A file number greater than 32766 was selected. Processing terminates.

Action Correct the FILE parameter and resubmit the request.

DBG05005E ERROR IN Adabas VOL/SER TABLE

Explanation A RABN could not be found on any volume in the Adabas VOL/SER table. Processing continues. VOL/SER is given a value '*****'.

Action Verify DD or DLBL statements for ASSO/DATA/WORK. Verify that the LWKP2 parameter is specified in the Unicenter CA-SpaceMan GLOBALS statement if the LWKP2 ADARUN parameter has been specified in the MPM. Contact Computer Associates.

DBG05006W ADAM FILE IS BEYOND PRIME AREA

Explanation Data Storage records for an ADAM file were found beyond the file's prime area. Processing continues.

Action Possible UNLOAD/RELOAD of the file with a larger DSSIZE.

DBG05007W ADAM FILE IS BEYOND FIRST EXTENT

Explanation Data Storage records for an ADAM file have expanded beyond the first Adabas extent. Processing continues.

Action Possible UNLOAD/RELOAD of the file with a larger DSSIZE.

DBG05008W FILE IS AT n EXTENTS (MAXIMUM IS 5)

Explanation An Adabas file has expanded to three (3) or more extents of Upper Index, Normal Index, Data Storage, or Address Convertor. Processing continues.

Action Possible UNLOAD/RELOAD of the file.

DBG05009W ADAM TECHNIQUE IS DISABLED FOR THIS FILE

Explanation	Due to the small size of the file, the randomization technique for record placement is not being used. Processing continues.
Action	None required.

DBG05010E COMMAND LOG OPEN ERROR

Explanation	An error occurred while trying to open an Adabas Command Log file. Processing terminates.
Action	Verify the DDLOG statement in your job control language.

DBG05011E COMMAND LOG READ ERROR

Explanation	An error occurred while trying to read an Adabas Command Log file. Processing terminates.
Action	Probable I/O error. Resubmit the request.

DBG05012E UPDATE UTILITY IN PROGRESS--DO NOT RUN CA-SPACEMAN

Explanation	A utility which modifies the database is currently executing, for example, ADALOD, ADAINV. Processing terminates.
Action	Resubmit the request after the utility has successfully completed.

DBG05013E HISTRECS OPEN ERROR

Explanation	An error occurred while trying to open the output file of history records. Processing terminates.
Action	Verify the HISTRECS DD statement in your job control language.

DBG05014W ALLOCATED SIZE OF s DATA SET EXCEEDS DATABASE COMPONENT SIZE

Explanation	The physical size of a data set, for example the Work data set, is greater than the size of its associated database component. Processing continues using the size of the database component.
Action	Compare the size of the component on a database report to a VTOC listing of the data set. Resolve the differences.

DBG05015E ERROR LOADING MODULE *x*

Explanation	An error was encountered when trying to load module <i>x</i> . Processing is terminated.
Action	Verify that ADAIOR is available in one of the load module libraries. Verify that the DDCARD input corresponds to that specified in the documentation.

DBG05016E INCORRECT WORK DATA SET FOR THIS DATABASE

Explanation	The DATABASE parameter in the control card does not match the internal DBID of the Work data set. Processing terminates.
Action	Verify the DATABASE parameter and the DDWORK statement in your job control language.

DBG05017E INTERNAL ERROR. SEND DUMP TO COMPUTER ASSOCIATES INTERNATIONAL INC.

Explanation	An internal logic error has occurred. Processing terminates.
Action	Send the dump to Computer Associates.

DBG05018E SIZE OF WORK PART 1 CANNOT BE DETERMINED

Explanation	The size of Work Part 1 is needed but cannot be determined from this version of Adabas. Processing terminates.
Action	Supply the LP parameter and resubmit the job.

DBG05019W LAST EXTENT IS ONLY *n* % USED. CONSIDER CHANGES TO THE *x* PARAMETER

Explanation	The last extent of an Adabas component is less than 10 per cent used.
Action	Consider changes to the indicated secondary allocation to reduce wasted disk space.

DBG05020E THIS VERSION OF Adabas IS NOT SUPPORTED BY THE CURRENT RELEASE OF CA-SPACEMAN

Explanation	You are attempting to run Unicenter CA-SpaceMan against a version of Adabas that is not supported.
Action	Use the appropriate version of Unicenter CA-SpaceMan that supports the version of Adabas being used.

DBG11006E INTERNAL ERROR IN VIRTUAL STORAGE MANAGEMENT ROUTINES

Explanation	The routines which manage virtual storage have detected an error in the control blocks which they use. The most likely cause of this is that the control blocks have been overlaid by other routines. The Data Collector will attempt to rebuild the control blocks.
Action	None required. However, if this situation recurs, please contact Computer Associates or your local distributor so that further diagnosis can be made.

DBG33001E SPECIFIED DDNAME LONGER THAN 7 CHARACTERS

Explanation	The filename specified in the VSE-FILES parameter is longer than the maximum allowed. Processing terminates with a condition code of 12.
Action	Correct the filename and resubmit the job.

DBG33002E DDNAME SPECIFIED FOR OVERRIDE IS NOT IN TABLE

Explanation	The specified filename is not present in the default VSE file table and filename additions are not allowed by this utility. Processing terminates with a condition code of 12.
Action	Correct the filename spelling or specify the correct filename and resubmit the job.

DBG33003E VSE LOGICAL UNIT SPECIFIED IS GREATER THAN 254

Explanation	The VSE logical unit number specified in the VSE-FILES parameter is greater than the maximum number allowed by VSE. Processing terminates with a condition code of 12.
Action	Correct the number specified and resubmit.

DBG33004E INVALID RECFM SPECIFIED FOR OVERRIDE

Explanation	The RECFM specified for a filename entry in the VSE-FILES parameter is not one of the valid formats of F, FB, V, VB. Processing terminates with a condition code of 12.
Action	Correct the record format specification and resubmit.

DBG33005E INVALID VALUE SPECIFIED FOR LABEL TYPE

Explanation	The label type specified for a filename entry in the VSE-FILES parameter is not one of the valid types, SL, NL, or BLP. Processing terminates with a condition code of 12.
Action	Correct the label type and resubmit.

DBG33026E INVALID VALUE SPECIFIED FOR TAPE DISPOSITION

Explanation	The tape disposition parameter specified for the filename indicated is not one of the valid responses of LEAVE, REWIND, or UNLOADS. Execution is terminated with a condition code of 12 following analysis of the control card.
Action	Correct the error and resubmit the request.

DBG33027E INVALID VALUE SPECIFIED FOR DEVICE TYPE

Explanation	The device type specified must be either TAPE or DISK. Execution is terminated with a condition code of 12 following analysis of the control cards.
Action	Correct the error and resubmit the request.

DBG60001E DDCARD *d* MISSING FOR VOLUME *v*

Explanation	The indicated DD card was missing for volume <i>v</i> during execute of the OS Allocation Utility. Processing terminates.
Action	Correct the error and resubmit the job.

DBG60002E REQUEST FOR *c* CYLINDERS GREATER THAN CAPACITY FOR VOLUME *v*

Explanation	The requested number of cylinders was larger than the capacity of the indicated volume. Processing terminates.
Action	Correct the error and resubmit the job.

DBG60003E DF/DS NOT INSTALLED ON SYSTEM. CA-SPACEMAN OS ALLOCATION NOT SUPPORTED WITHOUT DF/DS

Explanation Data Facility/Device Support function is required system option for the OS Allocation Utility. Processing terminates.

Action Contact your system programmer.

DBG60004E CVAF UNABLE TO DETERMINE VTOC TYPE ON VOLUME *v*

Explanation The indicated volume was not mounted or the VTOC index block is not initialized for the volume. Processing terminates.

Action Have the volume mounted.

DBG60005E UNIT SPECIFIED FOR DDCARD *d* IS NOT A DASD OR IS A VIO UCB

Explanation The unit specified on the indicated DD card was either not a DASD unit or a VIO device was specified. Processing terminates.

Action Correct the error and resubmit the job.

DBG60006E CVAFDSM CALL TERMINATED WITH RETURN CODE *n*, STATUS CODE *m*

Explanation The Common VTOC Access Facility (CVAF) call returned the indicated codes for a request for returning unallocated space. Processing terminates.

Action Check the *MVS/DFPDiagnosis Reference: DADSM/CVAF Diagnostic Aids* for code explanations, then take appropriate corrective action and resubmit the job.

DBG60007E NO CONTIGUOUS EXTENT AVAILABLE ON VOLUME *v* TO FILL *c* CYLINDER REQUEST

Explanation The indicated volume did not have enough contiguous space to satisfy the request. Processing terminates.

Action Insure enough contiguous space is available for the request or select another volume. Then resubmit the job.

DBG60008E CVAFDIR CALL TERMINATED WITH RETURN CODE *n*, STATUS CODE *m*, WHILE READING F5 DSCBS

- Explanation The CVAF routines terminated with the indicated codes while the OS Allocation Utility was finding the volume's unallocated space for a non-indexed VTOC volume. Processing terminates.
- Action Check the *MVS/DFPDiagnosis Reference: DADSM/CVAF Diagnostic Aids* for code explanations, then take appropriate corrective action and resubmit the job.

DBG60009E ALLOCATE SVC TERMINATED WITH HEX RETURN CODE *n*

- Explanation Allocation terminated with the code indicated. Processing terminates.
- Action See your system programmer for the return code explanations for the Allocate SVC.

DBG60010E SECOND ATTEMPT AT ABSOLUTE TRACK ALLOCATION FAILED

- Explanation A second attempt at allocating a request at an absolute location has failed. Processing terminates.
- Action Insure exclusive usage of the volume during job execution and resubmit the job.

DBG60011E CVAFDIR READ OF F1 DSCB FAILED WITH RETURN CODE *n*, STATUS CODE *m*

- Explanation The CVAF request to read a Format 1 DSCB failed with the indicated codes. Processing terminates.
- Action Check the *MVS/DFPDiagnosis Reference: DADSM/CVAF Diagnostic Aids* for code explanations, then take appropriate corrective action and resubmit the job.

DBG60012E CVAFDIR WRITE OF F1 DSCB FAILED WITH RETURN CODE *n*, STATUS CODE *m*

- Explanation The CVAF request to rewrite a Format 1 DSCB failed with the indicated codes. Processing terminates.
- Action Check the *MVS/DFPDiagnosis Reference: DADSM/CVAF Diagnostic Aids* for code explanations, then take appropriate corrective action and resubmit the job.

DBG60013E INVALID DSNAME SPECIFIED

Explanation	The data set name specified cannot be SYSCTLG or SYS1.VTOCIX.____. Processing terminates.
Action	Correct the error and resubmit the job.

DBG60014E THIS ALLOCATION FAILED FOR THE ABOVE ERROR. OS ALLOCATION UTILITY TERMINATED

Explanation	The current data set allocation is terminated for the error indicated. Processing terminates.
Action	Correct the indicated errors and resubmit the job.

DBG60015E SCRATCH SVC TERMINATED WITH RETURN CODE *n*, SCRATCH CODE *m*

Explanation	After an allocation error, the scratching of a previous allocated data set failed with the indicated codes. Scratch processing continues.
Action	See the OS message explanation for message IER283I for code explanations. Take corrective action specified in those explanations.

DBG60016I ALLOCATION FOR DATA SET *s* ON VOLUME *v*

Explanation	Information message indicating currently allocating specified data set on indicated volume. Processing continues.
Action	None required.

DBG60017I SUCCESSFUL ALLOCATION OF *c* CYLINDERS AT TRACK ADDRESS *aaa ddd*

Explanation	The previously indicated data set has been allocated for <i>c</i> cylinders starting at the absolute track location indicated. Processing continues.
Action	None required.

DBG60018I *s* SUCCESSFULLY SCRATCHED FROM VOLUME *v*

Explanation	The specified data set was scratched from the indicated volume following a previous error. Processing continues.
Action	None required.

DBG60019I VOLUME *v* MADE VOLUME NUMBER *n* OF MULTI-VOLUME DATA SET *s*

Explanation The specified volume is the *n*th in the volume sequence for the indicated data set. Processing continues.

Action None required.

DBG60020I *c* TOTAL CYLINDERS WERE ALLOCATED FOR DATA SET *s*

Explanation A total of *c* cylinders were allocated for indicated data set. Data set may be a multi-volume data set. Processing continues.

Action None required.

DBG60021E INPUT DATA SET DID NOT OPEN. CHECK FOR MISSING SYSIN DD STATEMENT

Explanation SYSIN DD missing for input control cards. Processing terminates.

Action Correct the error and resubmit the job.

DBG60022I DATA SET *s* IS CATALOGED ON THE FOLLOWING VOLUME/S

Explanation The specified data set has been cataloged to the volumes below in the order specified. Processing continues.

Action None required.

DBG60023W CATALOGING OF THE PRECEDING DATA SET FAILED WITH A RETURN CODE *n*

Explanation The cataloging of a data set failed with the return code indicated. Processing continues.

Action Check the OS message IEF287I for the return code explanation. Correct the error and catalog the data set specifying the correct volume order.

DBG60024I SUCCESSFUL ALLOCATION WITHOUT SPECIFYING ABSOLUTE TRACK LOCATION

Explanation The previous data set has been allocated successfully at a location and in the extents picked by the OS Allocation SVC. Processing continues.

Action None required.

DBG70001E INSUFFICIENT VIRTUAL STORAGE AVAILABLE. EXECUTION TERMINATED

Explanation	Not enough virtual storage was available for the necessary work areas and input/output buffers. Provide a larger amount of virtual storage by altering the SIZE parameter in the JCL. VSE users should increase the amount of virtual storage available for GETVIS storage. Execution is terminated.
Action	Correct the error and resubmit the request.

DBG70003E PERMANENT I/O ERROR ON OUTPUT FILE x FOR REQUEST y

Explanation	A permanent I/O error has occurred writing onto the indicated output file for the indicated request. No further output is attempted to this data set for any request. Other requests continue to process. The program terminates with a completion code of at least eight (8).
Action	After investigating the cause of the error and making any necessary corrections, re-execute the requests that wrote to this data set.

DBG70004E OUTPUT TERMINATED FOR FILE x

Explanation	This message accompanies message DBG70003E. No further output is attempted to this data set for any request. Other requests continue to process. The program terminates with a completion code of at least eight (8).
Action	After investigating the cause of the error and making any necessary corrections, re-execute the requests that wrote to this data set.

DBG70005E PERMANENT I/O ERROR FOR PRINT FILE x

Explanation	A permanent I/O error has occurred writing the indicated print file for the indicated request. No further writing is done to this data set for this or any other request. Processing continues for other output data sets for this and other requests. The program terminates with a completion code of at least eight (8).
Action	After making any necessary correction, re-execute the requests that wrote to this print file.

DBG70006E OUTPUT TERMINATED FOR PRINT FILE x

Explanation This message is associated with message DBG70005E. No further writing is done to this data set for this or any other request. Processing continues for other output data sets for this and other requests. The program terminates with a completion code of at least eight (8).

Action After making any necessary correction, re-execute the requests that wrote to this print file.

DBG70007E ERROR CONVERTING FIELD x y z

Explanation An error has occurred converting the format of the indicated field. Processing continues, but the value of this field in the output file is unpredictable and should not be used. The program terminates with a completion code of at least eight (8).

Action Verify that the requested output format is valid for the field indicated. Also, verify that the data is in the format expected. For example, one cause of this error is characters that are not digits occurring in a zoned field.

DBG70008E FOR REQUEST x FOR DDNAME y

Explanation This message accompanies message DBG70007E. It serves to further identify the conversion request that resulted in the error. Processing continues, but the value of this field in the output file is unpredictable and should not be used. The program terminates with a completion code of at least eight (8).

Action Verify that the requested output format is valid for the field indicated. Also, verify that the data is in the format expected. For example, one cause of this error is characters that are not digits occurring in a zoned field.

DBG70009E STCK VALUE OF RECORD IS h

Explanation This message accompanies message DBG70007E and provides the hexadecimal value of STCK. It identifies the Command Log record that contains the value that could not be converted. Processing continues, but the value of this field in the output file is unpredictable and should not be used. The program terminates with a completion code of at least eight (8).

Action Verify that the requested output format is valid for the field indicated. Also, verify that the data is in the format expected. For example, one cause of this error is characters that are not digits occurring in a zoned field.

DBG70025E DDNAME *filename* NOT FOUND FOR PRINT FILE

Explanation	No filename was found for a PRINT output file. The printed report is not created.
Action	Correct the JCL.

DBG70027E VIRTUAL STORAGE NOT AVAILABLE TO OPEN DDNAME *filename*. OPEN WAS NOT DONE

Explanation	A GETSPC failure occurred during opening of the indicated data set. The OPEN is not performed.
Action	Verify the blocksize for the file. Execute in a larger region.

DBG70028E DDNAME *filename* ALREADY OPEN AS AN INPUT FILE

Explanation	The same filename was specified for an output file as the input file name. The program terminates.
Action	Correct and resubmit.

DBG76005I *n* RECORDS WRITTEN ONTO OUTPUT FILE *x*

Explanation	The indicated number of records were written onto the output file whose filename is shown.
Action	None required.

DBG76006I *n* BLOCKS WRITTEN ONTO OUTPUT FILE *x*

Explanation	The indicated number of blocks were written onto the output file whose filename is shown.
Action	None required.

DBG76021I *n* RECORDS READ FROM INPUT FILE *filename*

Explanation	The indicated number of records were read from the input file with the indicated filename.
Action	None required.

DBG76022I *n* BLOCKS READ FROM INPUT FILE *filename*

Explanation The indicated number of blocks were read from the input file with the indicated filename.

Action None required.

DBG98003I CA-SPACEMAN USED *n* BYTES OF GETMAIN STORAGE

Explanation The maximum amount of storage gotten by the program during its execution. Note that this does not include space occupied by the program modules.

Action None required.

Unicenter CA-SpaceMan Online Messages

This chapter details messages issued by Unicenter CA-SpaceMan File Management for Adabas (Unicenter CA-SpaceMan) online facilities.

All messages have the following format.

***E*n text**

where *n* is a three-digit number.

Unicenter CA-SpaceMan Online Messages

E201 DUPLICATE VOLUME SERIAL SPECIFIED FOR VOLUME ENTRY

Explanation	The Unicenter CA-SpaceMan Configurator program encountered an input volume record that has the same volume serial as a previously entered volume record.
Action	Insure that the volume records created in the input projection have unique volume serials.

E202 WORK COMPONENT SPECIFIED WITHOUT ALL LOWER WORK ELEMENTS DEFINED

Explanation	The Unicenter CA-SpaceMan Configurator program encountered an input component record of W2 or W3 without the W1 and/or W2 component defined by an input component record. All previous work components must be defined before processing a work component.
Action	The file record for file 0 in the input projection, configuration or I/O sample is incomplete. Contact Computer Associates technical support.

E203 INTERNAL PROGRAM ERROR FINDING CORRECT VOLUME ENTRY

Explanation	The Unicenter CA-SpaceMan Configurator program found an internal volume number assigned to an allocated component which doesn't correspond to the internally calculated volume entry.
Action	Contact Computer Associates technical support for help in correcting the problem or obtaining further problem documentation.

E204 INVALID COMPONENT TYPE SPECIFIED FOR AN INPUT ENTRY

Explanation	The Unicenter CA-SpaceMan Configurator program encountered an input component entry that does not have a valid component id, for example, AC, UI, NI, DA, FV, DS, W1, W2 or W3.
Action	<ul style="list-style-type: none">■ If option 1 of the Configurator, select the input projection using the SPRO function and then display the files using the UFIL function to find the erroneous component ID and correct it.■ If option 2 or 3 of the Configurator, select the input I/O sample using function SIOS or select the input configuration using SCNF. Then display the component records using the BCMP function to find the erroneous component ID.■ Contact Computer Associates technical support for help in correcting the problem or obtaining further problem documentation.

E205 OVERLAP OF RABNS DURING RABN CALCULATION

Explanation	The Unicenter CA-SpaceMan Configurator program calculated a RABN range for an allocated component that overlapped a RABN range previously calculated for another component.
Action	Contact Computer Associates technical support for help in correcting the problem or obtaining further problem documentation.

E206 TOTAL COMPONENT SIZE REQUIRED GREATER THAN AVAILABLE SPACE

Explanation	The Unicenter CA-SpaceMan Configurator program calculated that the sum of all the input component sizes was greater than the amount of space made available on the input volumes or the available space was less than the minimums required for the ASSO or DATA data sets.
Action	Increase the amount of available space on the input volumes specified in the input projection, or add more volumes with available space to the projection.

E207 UNRECOGNIZED DEVICE TYPE FOR AN INPUT VOLUME ENTRY

Explanation	The Unicenter CA-SpaceMan Configurator program encountered an input volume entry that specified a device type that is not in the Configurator's valid device table. All documented Adabas device types are supported.
Action	Correct the device type specified in the volume entries in the input projection using the Update Volumes function. If this is a user-defined device which is not in the Configurator's valid device table, contact Computer Associates technical support for instructions on how to add a new device.

E208 INTERNAL PROGRAM ERROR DURING I/O SPLIT ALLOCATION

Explanation	The Unicenter CA-SpaceMan Configurator program found insufficient space left for the second half of a split component after it had already verified the space was available.
Action	Contact Computer Associates technical support for help in correcting the problem or obtaining further problem documentation.

E209 NO SPACE AVAILABLE TO ALLOCATE A COMPONENT ENTRY

Explanation	The Unicenter CA-SpaceMan Configurator program found that the required size to allocate a component was not available on a single volume or was not available on volumes eligible for splitting the component.
Action	Increase the amount of available space on the input volumes specified in the input projection or add more volumes with available space to the projection.

E210 INTERNAL PROGRAM ERROR DURING SPLIT SIZE ALLOCATION

Explanation	The Unicenter CA-SpaceMan Configurator program found the space unavailable for allocating a component being split by size after having checked that the size was available.
Action	Contact Computer Associates technical support for help in correcting the problem or obtaining further problem documentation.

E211 INTERNAL PROGRAM ERROR DURING FREESPACE ENTRY SEARCH

Explanation	The Unicenter CA-SpaceMan Configurator program failed to locate the desired freespace entry in an internal table while backing out an allocated component entry.
Action	Contact Computer Associates technical support for help in correcting the problem or obtaining further problem documentation.

E212 INTERNAL PROGRAM ERROR DURING I/O PLACEMENT OF COMPONENT

Explanation	The Unicenter CA-SpaceMan Configurator program found a flag set during allocation of a component that should not have been set at the point it was encountered.
Action	Contact Computer Associates technical support for help in correcting the problem or obtaining further problem documentation.

E213 NO GCB ENTRY FOR THE CONFIGURATOR DURING FILE PLACEMENT

Explanation	The Unicenter CA-SpaceMan Configurator program did not receive an FV component entry as input.
Action	<ul style="list-style-type: none">■ If option 1 of the Configurator, select the input projection using the SPRO function and then display the file 0 using the UFIL function to find if the FV component has been specified.■ If option 2 or 3 of the Configurator, select the input I/O sample using function SIOS or select the input configuration using SCNF. Then display the component records for file 0 using the BCMP function to see if the FV component has a record specified.■ Contact Computer Associates technical support for help in correcting the problem or obtaining further problem documentation.

E214 COMPONENT ENTRY VOLUME SERIAL NOT FOUND IN A VOLUME ENTRY

Explanation	The Unicenter CA-SpaceMan Configurator program encountered an existing component entry in Configurator option 3 processing that had a volume serial that had not been specified in a volume entry.
Action	Contact Computer Associates technical support for help in correcting the problem or obtaining further problem documentation.

E215 RABN OVERLAP DISCOVERED DURING RABN CHAIN ORDERING

Explanation	The Unicenter CA-SpaceMan Configurator program encountered an existing component entry in Configurator option 3 processing whose RABN range overlapped the RABN range for another component.
Action	Contact Computer Associates technical support for help in correcting the problem or obtaining further problem documentation.

E216 INTERNAL PROGRAM ERROR DURING VOLUME SPLITTING

Explanation	The Unicenter CA-SpaceMan Configurator program encountered an error during volume splitting.
Action	Contact Computer Associates technical support for help in correcting the problem or obtaining further problem documentation.

E217 INPUT COMPONENT ENTRY DOES NOT SPECIFY RABNS

Explanation	The Unicenter CA-SpaceMan Configurator program for option 2 encountered an input component entry that did not specify the RABN range for the existing component.
Action	Contact Computer Associates technical support for help in correcting the problem or obtaining further problem documentation.

E221 INVALID VOLSER

Explanation	The volume serial specified was invalid. The value entered was blank. The volume serial must be one to six alphanumeric or national characters.
Action	Enter the correct volume serial.

E222 DUPLICATE VOLUME

Explanation	The volume given in the VOLUME field as the new volume serial for the displayed volume record for an update is a duplicate to a volume record that already exists in the projection.
Action	Correct the volume serial entered in the VOLUME field and retry the update function. If the correct volume serial had been entered, select that volume record using UVOL and perform the volume updates on that record.

E223 INVALID DEVICE TYPE

Explanation The device type specified for a volume was invalid. All documented Adabas device types are supported.

Action Correct the device type. If this is a user-defined device which is not defined in the Configurator's valid device table, contact Computer Associates technical support for instructions on how to add a new device.

E224 CONFIGURATION DBID DOES NOT AGREE WITH PROJECTION

Explanation The DBID of the specified configuration or current database is not the same as the DBID of the projection being used for input to this configuration.

Action Use SCNF or SCDB check the DBID of the configuration or current database you have specified as additional input to the create configuration function. Select either a configuration or current database which has the same DBID as the projection. Then return to the Create Configuration (CCNF).

E225 IO SAMPLE DBID DOES NOT AGREE WITH PROJECTION

Explanation The DBID of the specified input/output sample is not the same as the DBID of the projection being used for input to this configuration.

Action Use SIOS to check the DBID of the input/output sample you have specified as additional input to the Create Configuration function. Select an input/output sample that has the same DBID as the projection. Then return to the Create Configuration (CCNF).

E226 CAN ONLY SELECT 20 SAMPLES

Explanation An attempt was made to select the 21st I/O sample to be combined into one output configuration. The SCOM function allows only 20 samples to be selected.

Action Terminate the SCOM function by entering a period in the NEXT FUNCTION field at the bottom of the screen to create the new output configuration using the 20 I/O samples already selected.

E227 SAMPLES CAN ONLY BE COMBINED FROM SAME DBID, RESELECT

Explanation An attempt was made to select an I/O sample to be combined that had a different DBID from those already selected in SCOM.

Action Reselect a new I/O sample from the set of available samples that has the same DBID. Otherwise terminate the selection process to include only those already selected by entering a period in the NEXT FUNCTION field.

E231 DUPLICATE FILE

Explanation The number given in the FILE field as the new file number for the displayed file record for an update is a duplicate to a file whose record already exists in the projection.

Action Correct the file number entered in the FILE field and retry the update function. If the correct file number had been entered, select that file record using UFIL and perform the file updates on that record.

E232 INVALID PADDING FACTOR

Explanation The padding factor specified for a file update was outside the allowable range. The allowable range is from one through 90 per cent.

Action Enter the correct padding factor.

E233 INVALID FILE OPTION

Explanation The file option specified for a file update was invalid. The values that can be specified are "N" for any of the options to negate the option; or "A" for ADAM files, "C" for coupled files, "I" for reuse ISNs, "E" for encrypted files, "X" for expanded files, or "U" for user-assigned ISNs to select the indicated option.

Action Enter the correct file option values.

E234 MUST SPECIFY AN ALLOCATION

Explanation A size value in Kbytes must be specified for the UI, NI, and DA components when generating a new file record in a projection. These values are used by the configurator to allocate a file within the database with the appropriate RABN range.

Action Enter the appropriate size values in the KB Alloc column for the indicated components.

E235 INVALID MAXISN

Explanation Either the MAXISN value is zero (0), or the value of MAXISN is less than MINISN.

Action Enter a non-zero MAXISN value or one that is greater than MINISN.

E236 INSUFFICIENT AVAILABLE SPACE

Explanation The available space on this volume is less than the sum of the spaces requested for Associator, Data, and Work.

Action Either increase the available space for this volume or decrease the space requested for Associator, Data, or Work.

E241 ONLY 1 DATABASE RECORD ALLOWED. ADD FAILED.

Explanation The UDBA function was requested to add a database record, and one already existed in the selected projection. The add was terminated.

Action Use the update form of the function where the record is displayed. Make the changes desired in the updatable fields of the screen, and press the ENTER key to update the current record. No entry is made in the COMMAND field for an update.

E242 THIS IS NOT A GENERATED CONFIGURATION. NO OPTION.

Explanation The configuration named in a GCRD function was not the name of a configuration generated by the Unicenter CA-SpaceMan Configurator. The named configuration does not have an option 1, 2, or 3 in its index record.

Action Specify the name of an appropriate output configuration to GCRD and retry the function.

E246 NO SAMPLES SELECTED, THEREFORE CONFIGURATION NOT SAVED

Explanation No samples were selected by the SCOM function to be combined into the output configuration specified to CIOS. Therefore no output configuration was saved under the name specified.

Action Reinvoke the CIOS function and select some I/O samples when SCOM is internally invoked by CIOS.

E247 FILE 0 RECORD NOT FOUND. CREATE WITH UFIL FIRST.

Explanation	A request to display the Work portion of file record 0 was made. File record 0 does not currently exist in the projection, configuration or I/O sample for which it was requested.
Action	If the Work components of file record 0 are to be updated, create this record first using the UFIL function and then reissue UWRK for the work updates.

E248 WORK RECORD CAN ONLY BE MODIFIED OR DELETED FOR PROJECTION

Explanation	An attempt to update or delete the file 0 Work record was made for a configuration or an I/O sample. The Work portion of a file 0 record can only be changed in a projection. The Work portion can always be displayed with UWRK.
Action	None required.

E249 VOLUME RECORDS CAN ONLY BE MODIFIED OR DELETED FOR PROJECTION

Explanation	An attempt to update or delete a volume record was made for a configuration or an I/O sample. The volume records can only be changed in a projection. The volume records can always be displayed with UVOL function.
Action	None required.

E250 ENTER VOLUME NAME OF RECORD TO BE RETRIEVED

Explanation	The UVOL function requires the volume serial to be specified so that the appropriate volume data may be displayed for update or to select the volume record for deletion.
Action	Enter the desired volume serial in the NEXT VOLUME field near the bottom of the screen and press ENTER.

E251 UNUSED SPACE CAN ONLY BE DISPLAYED FOR A CURRENT DATABASE

- Explanation An attempt was made to display unused space in the Associator and Data Storage data sets of a configuration, projection, or I/O sample. If a configuration was specified, it was not a current configuration for a database which has a name of the form '**CURRENTxxxx**' where xxxxx is the database id. Unused space elements are only available for current configurations, and not for projections or I/O samples.
- Action Use SCDB to select a current database for which you wish to display the unused space.

E252 ENTER PROJECTION NAME OF RECORD TO BE RETRIEVED

- Explanation The UPRO function requires the projection name to be specified so that its index record data may be displayed for update or to select the projection for deletion.
- Action Enter the desired projection name in the NEXT NAME field near the bottom of the screen and press ENTER.

E253 NO DATA SET RECORD FOR THIS CONFIGURATION OR DATABASE

- Explanation The GCRD function was invoked to generate utility control cards for Configurator option 1 or 2. In attempting to generate control cards for the Unicenter CA-SpaceMan OS Allocator, it required the data set record to obtain the data set names to allocate. This record has not been created by the user for this configuration or current database.
- Action Use the UDSN function to create the data set record for either the output configuration being used to generate control cards or for the current database configuration whose DBID is specified in the configuration. Then execute the GCRD function again.

E254 THERE IS NO RECORD FOR "xxxxxx"

- Explanation Various functions produce this message with variable data in the xxxxxx field. The function did not find any data for the specified record type which could be a database id, file number, start date-time or volume, depending on the update function.
- Action Correct the entered value if it erroneous, or enter a new value for which data exists.

E255 DATABASE RECORD CAN ONLY BE MODIFIED OR DELETED FOR PROJECTION

Explanation	An attempt to update or delete a database record was made for a configuration or an I/O sample. The database record can always be displayed with UDBA.
Action	None required.

E256 ENTER START DATE-TIME OF RECORD TO BE RETRIEVED

Explanation	The UIOS function requires the start date-time to be specified for an I/O sample so that its index record data may be displayed for update or to select the I/O sample for deletion.
Action	Enter the desired start date-time in the NEXT DATE field near the bottom of the screen and press ENTER.

E257 FILE RECORDS CAN ONLY BE MODIFIED OR DELETED FOR PROJECTION

Explanation	An attempt to update or delete a file record was made for a configuration or an I/O sample. The file records can be changed only in a projection. The file records can always be displayed using UFIL.
Action	None required.

E258 ENTER FILE NUMBER OF RECORD TO BE RETRIEVED

Explanation	The UFIL function requires the file number to be specified so that a file record may be displayed for update or to select the file for deletion.
Action	Enter the desired file number in the NEXT FILE field near the bottom of the screen and press ENTER.

E259 ENTER DATABASE ID OF RECORD TO BE RETRIEVED

Explanation	The UDBA function requires the database id to be specified so that its index record data may be displayed for update or to add or delete the database record.
Action	Enter the desired database id in the COMMAND field near the bottom of the screen and press ENTER.

E260 ENTER CONFIGURATION NAME OF RECORD TO BE RETRIEVED

- Explanation The UCNF function requires the name of a configuration to be specified so that its index record data may be displayed for update or to select the configuration for deletion.
- Action Enter the desired configuration in the NEXT NAME field near the bottom of the screen and press ENTER.

E261 NO PROJECTION WITH GIVEN NAME

- Explanation A projection for the name specified is not available in the Unicenter CA-SpaceMan Repository file.
- Action Correct the projection name specified and retry the function or create a projection with the specified name using CPRO.

E262 NO CONFIGURATION WITH GIVEN NAME

- Explanation A configuration for the name specified is not available in the Unicenter CA-SpaceMan Repository file.
- Action Correct the configuration name specified and retry the function.

E263 NO I/O SAMPLE WITH GIVEN TIME STAMP

- Explanation An I/O sample for the start date-time specified is not available in the Unicenter CA-SpaceMan Repository file.
- Action Correct the start date-time specified and retry the function.

E264 MUST SPECIFY PROJECTION NAME

- Explanation The function requires that a projection name must be entered in the appropriate field as a parameter to the function.
- Action The CCNF function requires the input projection name to be entered in the 'Projection' field. Enter and retry the function. The name of the projection to be created must be entered on the Projection Name line for CPRO. Enter and retry.

E265 SELECT EITHER CONFIGURATION OR I/O SAMPLE

Explanation	The CCNF function requires either an input configuration name or an input I/O sample start date-time to be entered as a parameter for Configurator options 2 and 3. Neither has been specified in the appropriate fields.
Action	Select an input configuration or I/O sample and enter the name or start date-time in the appropriate field on the display screen.

E266 MUST SELECT EITHER CONFIGURATION OR I/O SAMPLE

Explanation	The CCNF function requires either an input configuration name or an input I/O sample start date-time to be entered as a parameter for Configurator options 2 and 3. Both have been specified.
Action	Select either the input configuration name or the I/O sample start date-time and blank out the other field. Then press ENTER key to retry the function.

E267 MUST SPECIFY AN OUTPUT CONFIGURATION NAME

Explanation	The CCNF function requires that an output configuration name must be entered in the appropriate field as a parameter to the function.
Action	Enter a 1 to 16 character name in the output configuration field and retry the function. The name entered must not already exist as a configuration name.

E268 DUPLICATE OUTPUT CONFIGURATION NAME

Explanation	The name given in the output configuration field as a parameter to the CCNF function is a duplicate of an existing configuration name.
Action	Enter a new output configuration name and retry the function or delete the existing configuration of the same name using the UCNF function and then execute the CCNF function again specifying the same output configuration name.

E269 DUPLICATE PROJECTION NAME

Explanation	The name given in the PROJECTION NAME field as a parameter to the CPRO function is a duplicate of an existing projection name.
Action	Enter a new projection name and retry the function or delete the existing projection of the same name using the UPRO function and then execute the CPRO function again specifying the same projection name.

E270 MUST ENTER A DATABASE ID BETWEEN 1 AND 65535

Explanation The database id entered in the DBID field was outside the range of allowable values. The allowable range is from one through 65535.

Action Reenter a DBID value within the range specified and retry the function.

E271 OPTION MUST BE 1 THRU 3

Explanation The Configurator option must be a value of from one (1) to three (3). Option 1 indicates a total reconfiguration request. Option 2 indicates a request to reallocate the database data sets only. Option 3 is a request to add or relocate files in the specified database.

Action Reenter the appropriate option value in the option field and retry the CPRO function.

E272 DUPLICATE CONFIGURATION NAME

Explanation The name given in the CONFIGURATION field as the new configuration name for the displayed index record for an update is a duplicate of an already existing configuration name.

Action Enter a new configuration name into the CONFIGURATION field and retry the function.

E273 MAXFILES MUST BE 5 THRU 32766

Explanation The MAXFILES value specified in an update or add operation to a database record in a projection was not in the range of five through 32766.

Action Reenter the appropriate MAXFILES value in the allowable range and retry the UDBA function.

E274 THE MINIMUM DATABASE SIZE MAY NOT BE DECREASED

Explanation The minimum database sizes for ASSO or DATA may not be decreased in the database record in a projection. These are the minimums required so that the database may be reloaded after a total reconfiguration.

Action None required.

E275 FILE MUST BE BETWEEN 0 AND 32766

Explanation The file number entered in the FILE field was outside the range of allowable values. The allowable range is from zero (0) through 32766.

Action Reenter a file value within the range specified and retry the function.

E276 ENTER NAME OF CURRENT DATABASE TO BE RETRIEVED

Explanation The UCNF function requires the name of a configuration to be specified so that its index record data may be displayed for update or to select the configuration for deletion.

Action Enter the desired configuration in the NEXT NAME field near the bottom of the screen and press ENTER.

E277 NO CURRENT DATABASE WITH GIVEN NAME

Explanation A configuration for the name specified is not available in the Unicenter CA-SpaceMan Repository file.

Action Correct the configuration name specified and retry the function.

E281 INVALID FILE NUMBER

Explanation The file number specified was invalid. The file number was not a number between zero (0) and 32766.

Action Enter the correct file number.

E282 INVALID DATA SET TYPE; MUST BE "A", "D", OR "W"

Explanation The data set type specified in the DATASET field is not the letter "A" for Associator, "D" for Data Storage, or "W" for Work. Components only exist for these three types of data sets in a configuration or I/O sample.

Action Enter the correct data set type in the DATASET field.

E283 INVALID SELECTION NUMBER

- Explanation The selection number specified in the SELECT field of the function screen was not one of the prefix numbers indicated to the left of the displayed configurations, projections or I/O samples. Select value has a range of one to seven, depending on the number of displayed entries.
- Action Enter the number displayed to the left of the desired entry name in the SELECT field. Enter the function desired in the NEXT FUNCTION field and depress the ENTER key.

E284 INVALID SELECT CODE; MUST BE S OR BLANK

- Explanation The selection code specified in the selection field for a file entry displayed by the SFIL function must be either the letter "S" or blank. The selection field is at the left hand margin of the screen. The value S indicates to select the file, and blank means not to select the file.
- Action Enter one of the two allowable responses in the file entry's selection field and press ENTER.

E285 ALTERNATE RABNS CAN ONLY BE DISPLAYED FOR A CURRENT DATABASE

- Explanation An attempt was made to display the list of Alternate RABNs defined in a configuration, projection, or I/O sample. If a configuration was specified, it was not a current configuration for a database which has a name of the form '**CURRENTxxxx**' where xxxxx is the database id. Alternate RABN elements are only available for current configurations, and not for projections or I/O samples.
- Action Use SCDB to select a current database for which you wish to display the alternate RABNs.

E298 NO DATA FOR SPECIFIED RECID

- Explanation There is no data for the RECID passed to the internally invoked Unicenter CA-SpaceMan utility SPBLDFIL. The RECID is the internally generated identification for configurations, projections or I/O samples.
- Action Contact Computer Associates technical support for help in resolving the problem or collecting information to assist in resolution.

E299 NO DATA FOR SPECIFIED FILE/DBID

Explanation	There is no data to display from this function for the specified file number or database id.
Action	Enter the file number or the database id, which ever is appropriate for the function, for which data exists for the function. To see which files or databases have data, use BFIL to browse files, SCNF for current configurations, or SCDB for current databases.

E9BK BACK COMMAND NOT RECOGNIZED.

Explanation	The BACK command was issued at an inappropriate time. BACK is only accepted by the Unicenter CA-SpaceMan Help Manager as a request to return to the function from which help was requested. Either you typed BACK to some other program or the Help Manager was invoked directly and had no previous function to which it could return.
Action	Enter the command code for the function you want to invoke in the NEXT FUNCTION field or enter MENU in the NEXT FUNCTION field to see your last menu.

E9MM THERE IS NO PREVIOUS MENU TO BE SHOWN

Explanation	The menu currently being displayed is the oldest one that the Menu Manager has stored on its stack.
Action	Since there are no previous menus that have been saved, choose a function or menu from those currently displayed, or enter HELP for a description of how the MENU-MANAGER manages the stack of menus. This should explain why the MENU-MANAGER cannot display any previous menus. The MENU-MANAGER maintains a stack of seven (7) menus. If this is not sufficient, it is possible to increase the number of menus maintained on the stack. If you feel this is necessary, see H9CP for the names and telephone numbers for the support personnel who can register your request.

E9NH NO HELP REQUEST WAS PASSED TO THE HELP MANAGER

Explanation	The Help Manager was invoked without a help request.
Action	Do not invoke the Help Manager directly. If the Help Manager was invoked by a Unicenter CA-SpaceMan program, please contact Computer Associates technical support for help in resolving the problem or collecting information to assist in resolution.

E9NP THERE IS NO MEMBER TO PAGE THROUGH

Explanation	A paging request has been made to the Help Manager when there was no text member to page through. Remember that just pressing the ENTER key is an implied page forward, +P command.
Action	Enter any valid request other than a paging request or enter ? for more information of the Help Manager.

E9PN INVALID PAGE REQUEST; PLEASE REENTER

Explanation	A paging request was entered which does not follow the rules for for valid page requests.
Action	Valid page requests are +P, -P and P1 through P50. See H910 for more information.

E900 PLEASE ENTER SOMETHING

Explanation	This message is displayed when data must be entered in at least one field on a screen and no data was entered.
Action	The cursor is positioned at one of the input fields. Make sure you have followed the instructions on the screen. If these do not give you enough information, try the help facility by entering HELP or ? in the NEXT FUNCTION field. See H9CP for a list of support personnel if you need more help.

E963 ATTEMPT TO ACCESS AN UNAUTHORIZED FUNCTION

Explanation	Access to the application function is denied.
Action	<ul style="list-style-type: none">■ Enter any other application function request available to you.■ See your supervisor if you think you should have access to the function you entered, after verifying that you entered the function correctly by entering it again at the bottom of this screen in the NEXT FUNCTION field.■ See H9CP for a list of support personnel if you need more help.

E972 ATTEMPT TO ACCESS AN UNAUTHORIZED COMMAND

Explanation	Access to the Natural command entered is denied.
Action	<ul style="list-style-type: none">■ Enter any other command available to you.■ See your supervisor if you think you should have access to the command you entered, after verifying that you entered the command correctly by entering it again at the bottom of this screen in the NEXT FUNCTION field.■ See H9CP for a list of support personnel if you need more help.

E980 DO NOT INVOKE ERROR RECOVERY DIRECTLY

Explanation	The error recovery program was invoked without any error data, most likely as the result of entering its function code directly as a function request.
Action	<ul style="list-style-type: none">■ Never invoke the error recovery function directly.■ From the error recovery screen, it is possible to enter any function request that can be entered from any other screen of the application. No harm has been done to anything.

E982 FUNCTION UNAVAILABLE TO THIS APPLICATION

Explanation	The function you tried to access is not in this application nor in the common application library. Perhaps you are trying to access a function you have in another application or the function request was entered incorrectly.
Action	Double check the spelling of the function request entered, log on to the application in which the function exists, or contact the support person assigned to help you with system problems.

E990 MISCELLANEOUS ERROR; CONTACT COMPUTER ASSOCIATES

Explanation	An error has been recognized for which there is no specific error clarification member. This is the catch-all error message for the error recovery program.
Action	<ul style="list-style-type: none">■ Use the BACK function to go back to the function from which the error message was displayed.■ Record all pertinent information such as the function id and the key(s) of any record(s) you were working with.■ Contact the support person assigned to help you with system problems.

Index

A

- Abnormal Termination
 - U0026, 2-26
- Accounting control records, 10-5
- ADAASSO environment variable, 2-5
- Adabas
 - data sets, 3-4, 6-24
 - nucleus, 3-5, 7-5
 - space management, 4-6
- Adabas ADAIOR module, 2-22
- ADACLOG environment variable, 2-5
- ADADATA environment variable, 2-5
- ADADBID environment variable, 2-5
- ADADBS, 6-17
 - MODFCB, 4-10, 6-9, 6-12
 - MODFCB function, 6-2
- ADADTYP environment variable, 2-5
- ADAIOR, 3-4, 3-9, 3-4, 3-9, 11-1
- ADAIOR PRINT, 12-11
- ADALOAD environment variable, 2-5
- ADALOD, 4-7
 - LOAD, 6-9
- ADALP environment variable, 2-5
- ADALWKP2 environment variable, 2-6
- ADAM, 6-8
 - files, 4-2
 - key, 6-7
- ADAMER utility, 6-7
- ADAORD, 6-12
- ADARUN
 - TMOD, 2-23
 - TRACE, 2-23
- ADARUN LP, 6-18
- ADASAV, 3-12
- ADASVC environment variable, 2-6
- ADAWORK environment variable, 2-6
- Add/reload files, 9-8
- ADRC parameter, 5-3, 5-6
- APF authorized library, 1-2
- Artificial device types, 4-8
- ASSO parameter, 5-3, 5-7
- Associator
 - blocksize, 3-6, 4-8
 - data set, 4-6
 - padding, 6-12

B

- Billing, 10-3
- Block splits, 6-13
- Blocksize, 4-8
 - Associator, 3-6, 4-8
 - Data Storage, 3-6, 4-8
 - Work, 3-6, 4-8
- Buffer efficiency, 6-4
- Buffer pool, 7-5
- BY, 11-10, 11-11

C

CA Common Services for z/OS and OS/390
 CA LMP, 1-6

CA-APAS/Insight, 3-5

Cache, 4-9
 controller, 3-8

Capacity planning, 4-4

CA-SpaceMan
 and Natural Interface, 2-13
 components, 3-2
 history reports, 10-5
 Installation Worksheet, 2-5
 installing, 2-1
 Main Menu, 8-3
 Natural components, 2-15
 Repository, 7-1
 Repository file, 2-13

CATALL, 2-16

Chargeback, 4-4, 7-2, 10-3

CICS, 2-29

CICSLOAD environment variable, 2-6

CMDLOG, 11-1, 11-6, 11-10, 11-14

CMDLOG Fields, 12-1

CMWKF01, 7-6, 7-7

CMWKF02, 7-7

CNFLOAD environment variable, 2-6

Command Log, 6-23
 data set, 4-6

COMMAND-CODE field, 12-2

COM-LETE, 2-29

COMPONENT field, 12-3

Component record type, 7-3

Compression, 4-9, 6-4, 6-7
 file, 4-10

Configurations, 7-2
 Menu, 8-19

Configurator, 3-2, 7-2
 Add/reload files, 9-8
 Gathering Data, 9-2, 9-6, 9-8
 Input, 9-2, 9-5, 9-8

 Output, 9-4, 9-7, 9-10
 Processing, 9-3, 9-6, 9-9
 Reallocation, 9-5
 Reload files, 9-8
 Sorting Data, 9-2, 9-3, 9-6, 9-9

Control card generator. *Also see GCRD. Also see GCRD*

Control Card Generator, 3-2

Control-break-field list, 11-11

COPY, 11-14

CPU, 4-5

Current Database, 7-2
 Menu, 8-4

CYLINDER field, 12-4

CYLINDER-INC, 6-23
 parameter, 5-3

D

DASD storage requirements, 1-3

Data Collector, 3-2, 3-5, 3-2, 3-5, 7-7

DATA parameter, 5-3, 5-8

Data set record type, 7-3

Data sets

 Associator, 4-6
 Command Log, 4-6
 Data Storage, 4-6
 Protection Log, 4-6
 Sort, 4-6
 Temp, 4-6
 Work, 4-6

Data Storage

 blocksize, 3-6, 4-8
 data set, 4-6

Data Storage Space Table, 6-2, 6-6

Database

 by Pack, 6-18, 6-22
 Configurator, 3-2
 history, 7-2
 layout, 7-2
 reconfiguration, 3-12
 record type, 7-3

DATABASE parameter, 5-3

DATASET field, 12-4
DBGIN, 2-26
DBGIOR5, 3-2, 7-3, 11-1
DBGIOR5 exit routine, 2-22
DDIALOG, 11-1
DDLOG DD statement, 6-23
DDPRINT data set, 12-11
Debugging, 4-4
DECLARE, 11-5
DESCRIPTOR field, 12-4
Descriptors, 4-2
DEVICE parameter, 5-3
DGIOSUM, 2-24, 7-4, 7-7
DGRLDNAT, 2-15
DISKUNIT environment variable, 2-6
distribution tape format, 1-3
DSREUSE option, 6-5
DSST, 6-2, 6-6
DURATION field, 12-5

E

Effective descriptors, 6-11
environment variables
 CA-SpaceMan, 2-5
Environment variables
 ADAASSO, 2-5
 ADACLOG, 2-5
 ADADATA, 2-5
 ADADBID, 2-5
 ADALOAD, 2-5
 ADALP, 2-5
 ADALWKP2, 2-6
 ADASVC, 2-6
 ADATYP, 2-5
 ADAWORK, 2-6
 CICSLOAD, 2-6
 CNFLOAD, 2-6
 DISKUNIT, 2-6
 FSYSDBID, 2-6

FSYSFNDR, 2-6
FSYSPSWD, 2-6
FUSRDBID, 2-6
FUSRFNDR, 2-7
FUSRPSWD, 2-7
HLQ, 2-7
JOB CARD1, 2-7
JOB CARD2, 2-7
MLQ, 2-7
NATBATCH, 2-7
NATLOAD, 2-7
PRODVOL, 2-7
SPMLIBID, 2-7
SPMPSWD, 2-8
TAPEUNIT, 2-8
TAPEVOL, 2-8
VVVV, 2-8

EXTENT field, 12-5
Extents, 4-7
EXTRACT, 3-7, 3-9, 3-7, 3-9, 11-6, 11-7
EXTSEC parameter, 5-3

F

FCB, 6-9, 6-18
FDT, 6-10
Field Description Table, 6-10
File
 compression, 4-10
 history, 7-2
 record type, 7-3
FILE
 field, 12-5
 parameter, 5-6
File Control Block, 6-9, 6-16, 6-18
FIND SORTED, 3-9
Forward Index Compression, 6-10
Free Space Table, 6-18
FST, 6-18
FSYSDBID environment variable, 2-6
FSYSFNDR environment variable, 2-6
FSYSPSWD environment variable, 2-6

FUNCTION field, 12-6

FUSER files, 10-8

FUSRDBID environment variable, 2-6

FUSRFNR environment variable, 2-7

FUSRPSWD environment variable, 2-7

G

GCB, 6-18

GCRD, 2-30, 3-2, 3-13, 3-2, 3-13, 9-5, 9-10. *Also see Control Card Generator. See Control Card Generator*

General Control Block, 6-18

GET NEXT, 3-8

GET SAME, 3-6

GLOBALS

- parameter, 5-3 to 5-4
- statement, 5-2
- syntax, 5-2

GUIDMAIN program, 2-16

H

Head movement, 4-7, 6-23

HEAD-TRAVEL field, 12-7

HISTOGRAM, 3-6

HISTRECS, 7-5

- file, 10-2

HLQ environment variable, 2-7

I

I/O bottleneck, 3-4

I/O by Cylinder, 6-23

I/O load balancing, 4-5, 6-24

I/O Samples, 7-2, 7-6

- Menu, 8-9

IF, 11-5

Index

- block splits, 6-12
- record type, 7-3

Index Compression, 6-10

Inefficient padding, 6-12

INMENU program, 2-16

INSIGHT-LINES, 11-6, 11-10

Installation Worksheet

- CA-SpaceMan, 2-5

INTERVAL, 11-10

IO-COMPLETED field, 12-7

IOLOG, 6-23, 7-4, 11-1, 11-6, 11-10, 11-14

IOLOG data fields

- COMMAND-CODE, 12-2
- COMPONENT, 12-3
- CYLINDER, 12-4
- DATASET, 12-4
- DESCRIPTOR, 12-4
- DURATION, 12-5
- EXTENT, 12-5
- FILE, 12-5
- FUNCTION, 12-6
- HEAD-TRAVEL, 12-7
- IO-COMPLETED, 12-7
- IORCMDSQ, 12-9
- IOR-CTYPE, 12-9
- IOR-DATA, 12-10
- IOR-FUNC, 12-10
- IOR-PROGRAM, 12-10
- IOR-RESP, 12-11
- IO-SEQ-NO, 12-8
- IO-TYPE, 12-8
- PRINT-LINE, 12-11
- RABN, 12-11
- R-W, 12-12
- STCK, 12-12
- VOLSER, 12-12

IOLOG Data Fields, 12-2

IORCMDSQ field, 12-9

IOR-CTYPE field, 12-9

IOR-DATA field, 12-10

IOR-FUNC field, 12-10

IOR-PROGRAM field, 12-10

IOR-RESP field, 12-11
IO-SEQ-NO field, 12-8
IO-TYPE field, 12-8
ISNREUSE, 6-16
ISNs per block, 6-18

J

JCLSPMAN member, 2-23
JCLSPMIO member, 2-25
JOB CARD1 environment variable, 2-7
JOB CARD2 environment variable, 2-7

L

Label, 11-11
LBP size, 3-6
LDSPMNAT, 2-15
libraries
 authorizing, 1-2
license management program, 1-6
List of current reusable ISNs, 6-17
LMP key, 2-4
loading Natural components, 2-13
LOGIO parameter, 5-3
LOGIO=YES, 6-23
Lost blocks, 6-14
LP, 4-6, 6-18
 parameter, 5-3
LWKP2 parameter, 5-4
LWP/LS size, 3-6
LZ, 4-6, 6-22
LZAREA, 6-22

M

Managing space, 4-5
MAP parameter, 5-4
MAP=YES, 6-18
MAX ISN, 6-16, 6-18
MAXDS, 4-7
MAXNI, 4-7, 6-15
MAXUI, 4-7, 6-15
MENU program, 2-16
MISIZE, 6-15
MLQ environment variable, 2-7
MODFCB, 4-10, 6-2
MPM mode, 3-5

N

NATBATCH environment variable, 2-7
NATLOAD environment variable, 2-7
NSISN, 4-7

O

OBTAIN SAME, 3-6
OS Allocator, 6-24
OS Space Allocator, 2-31, 3-2, 9-5, 9-11. *Also see SPMALC. Also see SPMALC*
Output
 Configurator, 9-4, 9-7, 9-10
OUTPUT-EXIT, 11-6, 11-10
OUTPUT-FILE, 11-6, 11-10, 11-14
OUTPUT-INSIGHT-LINES, 11-7, 11-12
OWNER, 11-12
Ownership, 7-2

P

Padding

- area, 4-2, 4-10, 6-2, 6-13
- percentage, 6-2, 6-9, 6-12

Parameters

- GLOBALS, 5-3
- REPORT, 5-6

Prime area, 6-8

PRINT TO, 11-6, 11-10

PRINT-INSIGHT-LINES, 11-7, 11-12

PRINT-LINE field, 12-11

Processing

- Configurator, 9-3, 9-6, 9-9

PRODVOL environment variable, 2-7

Projections, 7-2

- Menu, 8-13

Protection Log data set, 4-6

R

RABN field, 12-11

RBI, 6-4

READ LOGICAL, 3-6, 4-8

READ PHYSICAL, 3-6

Reallocation, 9-5

Record type, 7-3, 11-5, 11-6, 11-11

RECSIZE-INC, 6-4, 6-7

- parameter, 5-8

RECS-PER-BLOCK-INC, 6-4

- parameter, 5-8

REDEFINES, 11-5

Reload files, 9-8

REORASSO, 6-12

REORDER ASSOCIATOR, 6-14

Reorganize, 9-2

REPORT

- parameter, 5-6 to 5-8

statement, 5-5

syntax, 5-5

Report program control cards, 5-1

Repository file, 3-1

maintenance, 10-4

requirements

DASD storage, 1-3

Unicenter CA-SpaceMan, 1-2

RESULTIS, 11-5

RETAIN, 3-6

Retention control records, 10-4

REUSEISN, 6-17

RMF statistics, 7-3

RSI, 6-4, 6-7

RSP072, 6-18

R-W field, 12-12

S

S1, 3-9

S122 Abend, 1-4

Secondary extents, 6-9

SECURITY, 11-12

SMS controlled volumes, 2-31, 9-11

SORT data set, 4-6

Sorting, 9-3, 9-6, 9-9

Space management, 4-5

Space usage

forecasting, 10-6

history reports, 10-5

SPDELETE, 2-11

SPEDIT edit macro, 2-10

SPM101, 6-9

SPM102, 6-5

SPM103, 6-4

SPM104, 6-3, 6-6

SPM105, 6-7

SPM201, 6-15
SPM202, 6-10, 6-11
SPM203, 6-12
SPM301, 6-17
SPM302, 6-16, 6-17
SPM303, 6-17
SPM401, 6-18, 6-22
SPM402, 6-23
SPMALC, 2-31, 3-2, 9-11. *Also see OS Space Allocator*
SPMCHARG, 10-3
SPMCMP, 2-13
SPMCNUF, 2-14
SPMDFLTS, 7-7
SPMDFLTS member, 2-24
SPMDIRUP, 7-3, 7-6
SPMHISUP, 10-3
SPMLIBID, 2-15
SPMLIBID environment variable, 2-7
SPMNENSU, 2-23, 2-25, 7-4, 11-3, 11-5
SPMPSWD environment variable, 2-8
SPMREORG, 3-13
SPMREP, 2-12, 2-23, 9-1
 Reports, 6-1
SPMUPD, 2-23, 7-3, 7-5, 10-2, 10-3
SPREADME, 2-10
SPUNLOAD, 2-13
SPUNLOAD member, 2-11
Statements
 GLOBALS, 5-2
 REPORT, 5-5
STCK field, 12-12
Subtasking, 3-5
SUBTOT, 11-11
SUMMARIZE, 3-7, 6-22, 6-23, 11-10, 11-11, 11-12
Summary of ADAM Distribution, 6-7
Summary of Compressed Record Sizes, 6-3, 6-6

Summary of Descriptor Allocations, 6-10, 6-11
Summary of Extents, 6-9, 6-15, 6-17
Summary of Records per Block, 6-4
Summary of Unused Bytes, 6-1, 6-5, 6-12
Summary of Unused ISNs per Block, 6-16
Super-descriptors, 3-6
SVC parameter, 5-4
Syntax
 GLOBALS statement, 5-2
 REPORT statement, 5-5

T

TAPEUNIT environment variable, 2-8
TAPEVOL environment variable, 2-8
TEMP data set, 4-6
Time, 4-5
TITLE, 11-6, 11-10
TOP ISN, 6-16, 6-18
TSO, 2-29

U

UBI, 6-2, 6-6, 6-13
UII, 6-16
Unicenter CA-SpaceMan requirements, 1-2
Unique, 3-6, 6-10
UNLOAD/RELOAD, 6-14
Unused bytes, 6-2, 6-6, 6-13
Unused ISNs, 6-16, 6-17
UNUSED-BYTE-INC, 6-2, 6-6, 6-13
 parameter, 5-7, 5-8
UNUSED-ISN-INC, 6-16
 parameter, 5-6
UQ, 6-10
Utility control cards, 7-3

V

V6FLOAD member, 2-30

V6REORFIL member, 2-30

V6UNRLD member, 2-30

VALOF, 11-5

VOLSER field, 12-12

Volume

record type, 7-3

serial table, 6-18

VVVV environment variable, 2-8

W

WHERE, 11-6, 11-10, 11-14

Work

blocksize, 3-6, 4-8

data set, 4-6

Part 1, 4-6, 6-18

Part 2, 4-6, 6-22

Part 3, 3-8, 4-7, 6-22

Part 4, 4-7

WTO, 12-11

WTOR, 12-11

X

XDIRUPD, 7-6, 7-7

XDIRUPD JCL, 2-18

XGCRD1 member, 2-30

XGCRD2 member, 2-30

XGCRD3 member, 2-30

XHISUPD JCL, 2-19

XSPMALC member, 2-31. *Also see SPMALC. Also see OS Space Allocator*

XSPMNENS JCL, 2-25

XSPMREP, 2-12

XSPMUPD JCL, 2-18

Z

ZERO-ENTRY-PRINT, 6-4, 6-7, 6-13, 6-16
parameter, 5-6, 5-7, 5-8

ZP, 6-4, 6-7, 6-13, 6-16



Computer Associates™