

CA-IDMS[®]/DB Reorg

User Guide

15.0



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.

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.

First Edition, December 2000

© 2000 Computer Associates International, Inc.
One Computer Associates Plaza, Islandia, NY 11749
All rights reserved.

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

Contents

About This Guide	vii
Chapter 1. General Information	1-1
1.1 What is CA-IDMS/DB Reorg?	1-3
1.2 Why Reorganize a Database?	1-4
1.3 Select Any Area or Entire Database	1-5
1.3.1 Updates Set Linkage Automatically	1-5
1.3.2 Leaves Relationships and Records Unaltered	1-5
1.3.3 Recognizes SR8 Index Areas	1-5
1.4 Improve System and CA-IDMS/DB Performance	1-7
1.5 How CA-IDMS/DB Reorg Performs Reorganization	1-8
1.5.1 Reorganization Controlled by Parameter	1-8
1.5.2 Reorganization Controlled by Subschema and DMCL	1-9
1.5.3 Reorganization Controlled by Internal Logic	1-10
1.6 Operational Flexibility During Reorganization	1-11
1.6.1 STOPAFTER and RESTART Parameters	1-11
1.6.2 Console Communication Facility	1-11
1.7 CA-IDMS/DB Reorg's Output	1-12
1.7.1 Audit Report Designed for Operations Flexibility	1-12
1.7.2 CA-IDMS/DB Reorg Interface Capabilities	1-12
Chapter 2. Parameters and Commands	2-1
2.1 Overview	2-3
2.2 Parameters and Their Functions	2-4
2.2.1 PROCESS Statement	2-4
2.2.2 COPY Statement	2-4
2.2.3 NOSWEEP Statement	2-4
2.3 PROCESS Statement	2-6
2.3.1 PROCESS Parameter	2-7
2.3.2 OLDSUB Parameter	2-7
2.3.3 NEWSUB Parameter	2-7
2.3.4 OLDDMCL Parameter	2-7
2.3.5 NEWDMCL Parameter	2-8
2.3.6 OLDDBN Parameter	2-8
2.3.7 NEWDBN Parameter	2-8
2.3.8 RPT Statement	2-8
2.3.8.1 RPT Syntax and Options	2-9
2.3.9 RESTART Parameter	2-10
2.3.9.1 RESTART Syntax and Options	2-11
2.3.10 STOPAFTER Parameter	2-12
2.3.10.1 STOPAFTER Syntax and Options	2-12
2.3.11 Use RESTART with STOPAFTER	2-13
2.3.12 DSPLYINT Parameter	2-14
2.3.12.1 DSPLYINT Syntax and Options	2-14
2.3.12.2 Route Code Syntax and Parameters	2-15
2.3.13 TSUSERS Parameter	2-17
2.3.14 SORTFLAG Parameter	2-17

2.3.14.1 SORTFLAG Syntax and Options	2-17
2.3.15 SORTSIZE Parameter	2-17
2.3.16 BLOCKNUM Parameter	2-17
2.4 COPY Statement	2-19
2.4.1 COPY Parameter	2-19
2.4.2 AREA Parameter	2-19
2.4.3 PRIORITY Parameter	2-20
2.4.3.1 Specify Record Type Priority by Class or Sublist	2-20
2.4.3.2 PRIORITY Syntax and Options	2-21
2.4.4 PAGERESERVE Parameter	2-22
2.4.5 TARGETROOT Parameter	2-22
2.5 NOSWEEP Statement	2-23
2.5.1 When to Use NOSWEEP	2-23
2.5.2 NOSWEEP Statement Syntax	2-23
2.6 Console Communications Facility	2-25
2.6.1 Functions of Console Communication Commands	2-25
2.6.1.1 DISPLAY STATUS Command	2-25
2.6.1.2 SET INTERVAL Command	2-25
2.6.1.3 STOP Command	2-26
2.6.1.4 CANCEL Command	2-26
Chapter 3. Concepts	3-1
3.1 Overview	3-3
3.2 Understanding CA-IDMS/DB Reorg Processing	3-4
3.2.1 Reorganization--Not Restructuring	3-4
3.2.2 How Is CA-IDMS/DB Reorg Different From Other Utilities?	3-4
3.3 CA-IDMS/DB Reorg's Inputs	3-5
3.4 CA-IDMS/DB Reorg's Record Allocation Techniques	3-7
3.4.1 Minimize Record Overflow	3-7
3.4.2 Reduce Fragmented Records	3-7
3.4.3 Allocating DIRECT Records	3-7
3.4.4 Allocating CALC Records	3-8
3.4.5 Allocating VIA Records	3-9
3.4.6 Sorted VIA Records	3-10
3.4.7 Allocation and the PRIORITY Parameter	3-11
3.4.8 Allocation and Page Reserve	3-12
3.5 CA-IDMS/DB Reorg Reorganization Capabilities	3-14
3.5.1 Reorganization Capabilities Controlled by Parameters	3-14
3.5.2 Direct an Efficient Reorganization Through Parameters	3-16
3.5.3 Reorganization Capabilities Controlled by Subschema and DMCL Definition	3-17
3.5.4 Automatic Reorganization Controlled by Internal Logic	3-19
3.6 Reorganization Steps	3-21
3.6.1 Functions of the Reorganization Steps	3-23
Chapter 4. System Output	4-1
4.1 Overview	4-3
4.2 CA-IDMS/DB Reorg Output	4-4
4.2.1 Audit Report	4-4
4.2.2 Status Display	4-4
4.2.3 Interface File	4-4

4.3	Audit Report	4-5
4.3.1	Audit Subreport--Area Recap for Old Subschema	4-6
4.3.2	Audit Subreport--Set Linkage Recap for Area	4-8
4.3.3	Restart Information Messages	4-10
4.4	Status Display	4-12
4.4.1	Status Display Fields	4-12
4.5	DBKEYS Interface File	4-13
Chapter 5.	Operations	5-1
5.1	Overview	5-3
5.2	Database Access	5-4
5.3	Operational Flow and Considerations	5-5
5.3.1	Integrated Indexes	5-5
5.3.2	File Space Not Needed	5-5
5.3.3	Allocating Space for Work Files and Sort-Work Areas	5-6
5.3.4	Work Files	5-6
5.3.4.1	ANALYSIS Step	5-6
5.3.4.2	OPTALLOC Step	5-8
5.3.4.3	PRIALLOC Step	5-10
5.3.4.4	SECALLOC Step	5-12
5.3.4.5	MATCH Step	5-17
5.3.4.6	IIXSDBK Step	5-19
5.3.4.7	UPDLINK Step	5-20
5.3.4.8	STORE Step	5-22
5.3.5	Sort-Work Areas	5-22
5.3.6	Summary of Procedures for Operating CA-IDMS/DB Reorg	5-23
5.4	CA-IDMS/DB Reorg Processing--OS/390 Environment	5-26
5.4.1	Process 1--Defining the New Areas	5-26
5.4.2	Process 2--Gathering Information	5-27
5.4.3	Process 3--Allocating Work Files	5-28
5.4.4	Process 4--Reorganizing the Database	5-30
5.5	CA-IDMS/DB Reorg Processing--VSE/ESA Environment	5-34
5.5.1	VSE/ESA File Assignments	5-34
5.5.2	VSE/ESA File Processing Alternate Method	5-34
5.5.3	Process 1--Defining the New Areas	5-35
5.5.4	Process 2--Information Gathering	5-35
5.5.5	Process 3--Allocating Work Files	5-37
5.5.6	Process 4--Reorganizing the Database	5-39
5.6	CA-IDMS/DB Reorg Processing--VM/ESA Environment	5-43
5.6.1	Process 1--Defining the New Areas	5-43
5.6.2	Process 2--Information Gathering	5-43
5.6.3	Process 3--Allocating Work Files	5-47
5.6.4	Process 4--Reorganizing the Database	5-49
Chapter 6.	Messages	6-1
6.1	Condition Codes	6-4
6.2	Audit Report	6-5
6.3	VSE/ESA File Processing Alternate Method	6-6
6.4	Message Text	6-7

Index X-1

About This Guide

What this Manual is About

This guide provides the information needed to run CA-IDMS/DB Reorg. In addition, the many features that CA-IDMS/DB Reorg offers are documented to assist you.

Organization

Chapter	Description
1	Presents a summary of CA-IDMS/DB Reorg features and functions.
2	Provides a guide to CA-IDMS/DB Reorg's parameter statements and console communication commands.
3	Provides detailed background information on how CA-IDMS/DB Reorg performs a reorganization through a step-by-step process. Each step and its function is discussed and illustrated.
4	Describes the output created by CA-IDMS/DB Reorg. Pages from a sample Audit Report and a sample Status Display are provided with detailed explanations of the content from both types of output.
5	Describes operational procedures. It provides information about accessing the database, important operational considerations, and file allocation techniques.
6	Provides a list of all informative, error, and warning messages generated by CA-IDMS/DB Reorg, along with reasons for occurrence and suggested actions to be taken.
Index	Provides an alphabetical list of CA-IDMS/DB Reorg concepts with their locations in the user guide.

CA-IDMS/DB Reorg Publications

In addition to this guide, Computer Associates supplies an installation guide to use as a reference tool that gives complete information about the installation of the products.

Understanding Syntax Diagrams

Look at the list of notation conventions to see how syntax is represented in this manual. The example following the list shows how the conventions are used.

Table 0-1. Syntax Notation and Parameter Syntax Rules

Example	Function
PROCESS	Keywords appear in UPPERCASE. The minimum required portion of each keyword is UNDERSCORED. If a portion of or an entire keyword is not underscored, you can omit that portion or that keyword.
AREA= <i>area-name</i>	Variables appear in lowercase italics. You must substitute an appropriate value for each variable.
[,TARGETROOT]	Brackets indicate optional clauses.
←	Points to the default in a list of choices.
[,DSPLYINT={ <i>interval</i> REQUEST STEP }]	Braces enclose two or more options. You must select one of the options.
[,PRIORITY=({DIRECT CALC <i>record-name</i> ...})]	An ellipsis indicates that one or more options can be selected and entered in any order.
Order of Parameter Statements	The PROCESS statement must be entered first. The other three parameter statements can be entered in any order
Continuing a Parameter Statement	To continue a parameter statement onto the next record, key in a trailing comma. Do not split a keyword phrase between two records.
Entering Blanks In Parameter Statements	You can enter blanks (character spaces) to separate keywords and improve readability in a parameter statement without affecting processing. However, do not embed blanks within a keyword or value field.
Entering Parameter Statements	On an 80-character input record, you must enter all parameter syntax between positions 1 and 72, inclusive.
Comments	Enter an asterisk (*) in column 1 to indicate a comment.
Series of Variables	Parenthesis are required around a series of variables. Variables must be separated by commas.

Sample syntax diagram

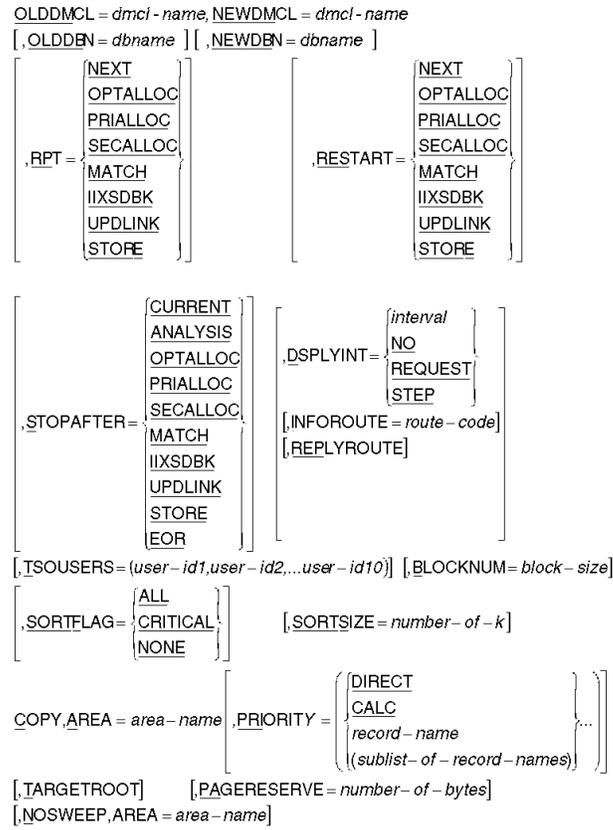


Figure 0-1. Sample Syntax Diagram

Chapter 1. General Information

- 1.1 What is CA-IDMS/DB Reorg? 1-3
- 1.2 Why Reorganize a Database? 1-4
- 1.3 Select Any Area or Entire Database 1-5
 - 1.3.1 Updates Set Linkage Automatically 1-5
 - 1.3.2 Leaves Relationships and Records Unaltered 1-5
 - 1.3.3 Recognizes SR8 Index Areas 1-5
- 1.4 Improve System and CA-IDMS/DB Performance 1-7
- 1.5 How CA-IDMS/DB Reorg Performs Reorganization 1-8
 - 1.5.1 Reorganization Controlled by Parameter 1-8
 - 1.5.2 Reorganization Controlled by Subschema and DMCL 1-9
 - 1.5.3 Reorganization Controlled by Internal Logic 1-10
- 1.6 Operational Flexibility During Reorganization 1-11
 - 1.6.1 STOPAFTER and RESTART Parameters 1-11
 - 1.6.2 Console Communication Facility 1-11
- 1.7 CA-IDMS/DB Reorg's Output 1-12
 - 1.7.1 Audit Report Designed for Operations Flexibility 1-12
 - 1.7.2 CA-IDMS/DB Reorg Interface Capabilities 1-12

1.1 What is CA-IDMS/DB Reorg?

CA-IDMS/DB Reorg is a comprehensive, time-saving tool for database administrators and systems programmers who want to reorganize a CA-IDMS/DB database. CA-IDMS/DB Reorg gives you the freedom to solve physical organization problems anywhere in the database without requiring you to unload and load the entire database. Besides expanding page ranges for a selected area, CA-IDMS/DB Reorg provides a variety of reorganization possibilities--including the ability to prioritize records for target page storage.

1.2 Why Reorganize a Database?

Your CA-IDMS/DB database is most useful when information can be accessed quickly and additional data can be stored efficiently. As more records are added to and requested from your database, response time often increases and CA-IDMS/DB operates inefficiently.

CA-IDMS/DB Reorg is a comprehensive and flexible reorganization tool for solving problems in any CA-IDMS/DB database. In addition, CA-IDMS/DB Reorg eliminates the use of multiple utilities and special programs, and the need to unload and load the entire database when only a single area needs to be reorganized.

1.3 Select Any Area or Entire Database

CA-IDMS/DB Reorg was designed to simplify the process of solving physical organization problems in an entire database or any area of the database. You can select one or more areas, to alter page ranges and perform other reorganization tasks without tying up the entire database. See Figure 1-1 on page 1-6.

1.3.1 Updates Set Linkage Automatically

CA-IDMS/DB Reorg lets you reorganize an area independent of its set linkage with other areas. When reorganizing a single area that has set linkage to another area in the database, CA-IDMS/DB Reorg automatically updates the set linkage in the other areas.

1.3.2 Leaves Relationships and Records Unaltered

CA-IDMS/DB Reorg will not alter the logical relationships between records or change the size or content of any record.

1.3.3 Recognizes SR8 Index Areas

Further, CA-IDMS/DB Reorg recognizes and integrates SR8 index areas during a reorganization. CA-IDMS/DB Reorg automatically updates the index pointers when reorganizing member records.

1.3 Select Any Area or Entire Database

When reorganizing only Area 4, CA-IDMS/DB REORG will expand page ranges in Area 4 and update set linkage with Area 3 and the Index Area.

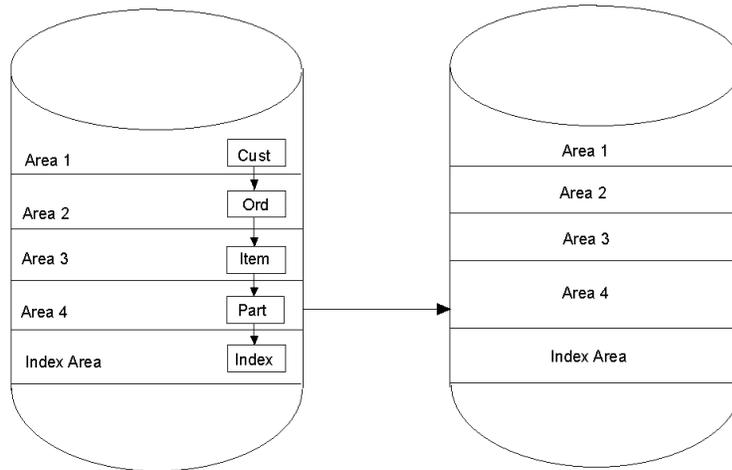


Figure 1-1. Reorganizing a Single Area in a Database

1.4 Improve System and CA-IDMS/DB Performance

CA-IDMS/DB Reorg provides many features available for the first time in a single utility. In addition to selecting specific areas of your database for reorganization, CA-IDMS/DB Reorg improves performance by allowing you to:

- Rank records for target page storage
- Control the placement of record fragments
- Rebuild integrated index structure
- Optimize the CALC set
- Recluster VIA sets
- Eliminate logically deleted records
- Eliminate relocated records.

Many of these performance-tuning features are related to CA-IDMS/DB Reorg's record allocation techniques. The techniques ensure that records are stored in the most efficient manner possible by being placed on, or as near as possible to, their target pages. This method reduces record fragmentation and keeps record overflow to a minimum. You can also allocate records according to the priorities of your processing environment. (All CA-IDMS/DB Reorg concepts are explained in detail in Chapter 3, "Concepts" on page 3-1.)

After you have run CA-IDMS/DB Reorg against your database, you will be able to achieve better system performance. Programs will run more efficiently because I/Os will be significantly reduced. CA-IDMS/DB Reorg's independence from other utilities and from special purpose programs makes it the most efficient, time-saving, and comprehensive tool for reorganizing your CA-IDMS/DB database.

1.5 How CA-IDMS/DB Reorg Performs Reorganization

CA-IDMS/DB Reorg performs a physical reorganization of all or part of the database, leaving logical relationships between records, and the size and content of each record, unchanged. The reorganization is controlled by user-specified parameters, the definition of the subschemas, and the utility's own internal logic (see Figure 1-2 on page 1-9). Because CA-IDMS/DB Reorg reorganizes a database by copying the old area to the new area, you must identify both the old subschema and the new subschema when you execute CA-IDMS/DB REORG. Any changes made to the database through the definition of the new subschema will be reflected in the reorganized database.

CA-IDMS/DB Reorg exploits CA-IDMS/DB's I/O features:

- XA database buffers and control blocks
- ESA database support
- Dynamic database file allocation
- Unrestricted SEGMENT name usage as database names

For more information on these CA-IDMS database features, see *CA-IDMS Database Administration* and *CA-IDMS System Operations*.

In addition, the SYSIDMS PREFETCH parameter for read-ahead processing by CA-IDMS/DB engine, replaces the read-ahead processing previously provided by FASTSCAN and EXCP I/O level processing formerly specified in the GSDTPARM installation defaults.

1.5.1 Reorganization Controlled by Parameter

CA-IDMS/DB Reorg's parameters allow you to control internal physical reorganization functions for sets and records and to specify where the reorganization is to be performed. By specifying parameters you can also interrupt CA-IDMS/DB Reorg, stop and restart processing, and control its run time. These capabilities give you the power to perform a reorganization that is compatible with your processing environment and tailored to the way your applications access the database.

Using CA-IDMS/DB Reorg's parameters, you can:

- Identify the area to be reorganized regardless of set linkage with other areas
- Specify the priority in which record types are placed on their target pages
- Identify the type of clustering for VIA sets
- Establish a page reserve for each area
- Monitor the reorganization process
- Direct the operational efficiency of CA-IDMS/DB Reorg in your environment.

Chapter 2, "Parameters and Commands" on page 2-1 and Chapter 3, "Concepts" on page 3-1 include details about parameter-controlled reorganization capabilities.

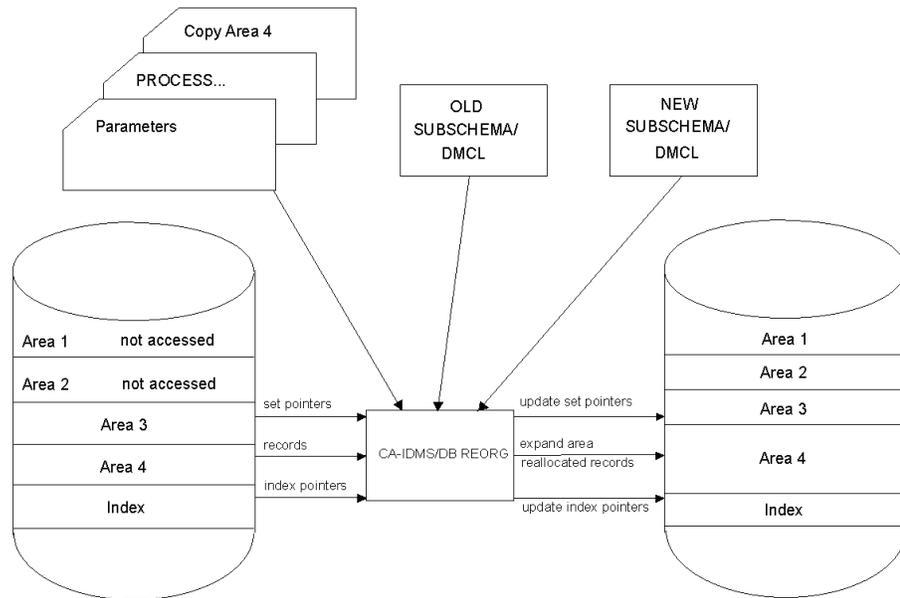


Figure 1-2. CA-IDMS/DB Reorg Process

1.5.2 Reorganization Controlled by Subschema and DMCL

CA-IDMS/DB Reorg is also controlled by the subschema and DMCL definitions of the old and the new database. Many reorganization functions at the area, set, index, and record levels depend on how you have defined the new subschema and DMCL. By comparing the new subschema and DMCL definition to the old subschema and DMCL definition, DB/REORG will:

- Update integrated indexes when member records are reorganized
- Retarget CALC records
- Retarget VIA records
- Expand or reduce an area's page range
- Expand or reduce an area's page size
- Expand or reduce a record's page range
- Change a record's location mode
- Move a record to another area

Chapter 3, “Concepts” on page 3-1 includes details about these subschema-controlled reorganization capabilities.

Note: All records in the old subschema and all records in the new subschema must be assigned to the same page group.

1.5.3 Reorganization Controlled by Internal Logic

Many CA-IDMS/DB Reorg reorganization functions are directed by internal processing logic. They occur automatically, whenever you invoke CA-IDMS/DB Reorg and identify an area to be reorganized; you do not need to supply parameters or changes to the definition of the subschema. CA-IDMS/DB Reorg's internal processing logic automatically provides these reorganization functions:

- Optimize CALC sets
- Optimize VIA sets
- Minimize fragmented records
- Eliminate logically deleted records without executing additional utilities
- Retarget relocated records to their home pages
- Maintain order of all CALC records
- Store records in true physical sequential order.

Chapter 3, “Concepts” on page 3-1 contains details about these reorganization functions.

1.6 Operational Flexibility During Reorganization

CA-IDMS/DB Reorg performs a reorganization on a step-by-step basis (steps are discussed in detail in Chapter 3, “Concepts” on page 3-1). To gain operational flexibility during a reorganization you can control processing for each step with STOPAFTER/RESTART parameter options of the PROCESS statement. You can also monitor the reorganization through CA-IDMS/DB Reorg's Console Communication Facility.

1.6.1 STOPAFTER and RESTART Parameters

The STOPAFTER parameter option in the PROCESS statement allows you to interrupt CA-IDMS/DB Reorg after any individual step. Later, you can direct CA-IDMS/DB Reorg to pick up where it left off and complete the reorganization by using the RESTART option. See 2.3.11, “Use RESTART with STOPAFTER” on page 2-13 for a detailed explanation of these options.

The STOPAFTER/RESTART feature allows you to interrupt the reorganization process to:

- Recover from a system failure
- Perform backup procedures
- Perform operations functions
- Conserve and manage disk space.

1.6.2 Console Communication Facility

CA-IDMS/DB Reorg provides a Console Communication Facility that allows you to monitor the reorganization from the operator console and TSO terminals. The facility is established through a CA-IDMS/DB Reorg parameter and controlled by issuing dynamic commands at the operator console. The parameter gives you the ability to request that CA-IDMS/DB Reorg status be displayed at specified intervals, at the end of each step, or on demand. You can also turn off the communication facility.

The Communication Commands allow you to view the status, to stop or cancel the reorganization process, and to change the frequency of the interval at which the status is displayed. The Console Communication Facility and its commands are covered in detail in 2.6, “Console Communications Facility” on page 2-25.

1.7 CA-IDMS/DB Reorg's Output

CA-IDMS/DB Reorg's primary output is the reorganized database. It is structured from the parameter options you select and the definition of the new subschema and DMCL. In addition to the reorganized database, CA-IDMS/DB Reorg automatically produces a comprehensive Audit Report and may create an interface file for CA-IDMS Utilities or DB-EZ Reorg.

1.7.1 Audit Report Designed for Operations Flexibility

The CA-IDMS/DB Reorg Audit Report is composed of up to four parts, depending on the parameter options you select. The two standard portions of the Audit Report include:

- A complete listing of the parameters used during the current run
- A log of all messages generated by CA-IDMS/DB Reorg, which includes a log of start and stop times for all reorganization steps executed.

In addition to the two standard portions, the Audit Report can contain:

- Audit/Analysis Reports
- Restart Information Messages.

Audit/Analysis Reports are produced only when the first step (analysis) of CA-IDMS/DB Reorg is run. These reports show how CA-IDMS/DB Reorg will execute against your database, analyzing the reorganization that will occur for the database. This analysis is based on the parameter options you have specified and the area, set, and record definitions in the old and new subschemas.

Restart Information Messages appear on the Audit Report if you halted CA-IDMS/DB Reorg before all reorganization steps executed successfully. These messages contain the information necessary to restart CA-IDMS/DB Reorg and information about the steps that have not executed or steps that terminated unsuccessfully.

1.7.2 CA-IDMS/DB Reorg Interface Capabilities

CA-IDMS/DB Reorg can create an interface file for use with DB-EZ Reorg or CA-IDMS Utilities. The interface file contains database keys and when used with DB-EZ Reorg's inflight reorganization product, resolves changed database record locations.

Chapter 2. Parameters and Commands

2.1 Overview	2-3
2.2 Parameters and Their Functions	2-4
2.2.1 PROCESS Statement	2-4
2.2.2 COPY Statement	2-4
2.2.3 NOSWEEP Statement	2-4
2.3 PROCESS Statement	2-6
2.3.1 PROCESS Parameter	2-7
2.3.2 OLDSUB Parameter	2-7
2.3.3 NEWSUB Parameter	2-7
2.3.4 OLDDMCL Parameter	2-7
2.3.5 NEWDMCL Parameter	2-8
2.3.6 OLDDBN Parameter	2-8
2.3.7 NEWDBN Parameter	2-8
2.3.8 RPT Statement	2-8
2.3.8.1 RPT Syntax and Options	2-9
2.3.9 RESTART Parameter	2-10
2.3.9.1 RESTART Syntax and Options	2-11
2.3.10 STOPAFTER Parameter	2-12
2.3.10.1 STOPAFTER Syntax and Options	2-12
2.3.11 Use RESTART with STOPAFTER	2-13
2.3.12 DSPLYINT Parameter	2-14
2.3.12.1 DSPLYINT Syntax and Options	2-14
2.3.12.2 Route Code Syntax and Parameters	2-15
2.3.13 TSOUSERS Parameter	2-17
2.3.14 SORTFLAG Parameter	2-17
2.3.14.1 SORTFLAG Syntax and Options	2-17
2.3.15 SORTSIZE Parameter	2-17
2.3.16 BLOCKNUM Parameter	2-17
2.4 COPY Statement	2-19
2.4.1 COPY Parameter	2-19
2.4.2 AREA Parameter	2-19
2.4.3 PRIORITY Parameter	2-20
2.4.3.1 Specify Record Type Priority by Class or Sublist	2-20
2.4.3.2 PRIORITY Syntax and Options	2-21
2.4.4 PAGERESERVE Parameter	2-22
2.4.5 TARGETROOT Parameter	2-22
2.5 NOSWEEP Statement	2-23
2.5.1 When to Use NOSWEEP	2-23
2.5.2 NOSWEEP Statement Syntax	2-23
2.6 Console Communications Facility	2-25
2.6.1 Functions of Console Communication Commands	2-25
2.6.1.1 DISPLAY STATUS Command	2-25
2.6.1.2 SET INTERVAL Command	2-25
2.6.1.3 STOP Command	2-26
2.6.1.4 CANCEL Command	2-26

2.1 Overview

This chapter is a guide to CA-IDMS/DB Reorg's parameter statements and Console Communication Commands. CA-IDMS/DB Reorg parameters initiate processing, identify the portion of the database to be reorganized, and indicate the number of reorganization steps to be performed. Parameters also control production of the Restart Information Messages, allow you to rank records for optimum storage locations, and establish page reserves. To use these parameters effectively you must understand how your database is physically organized. While these major parameters control CA-IDMS/DB Reorg processing, much of the reorganization process depends on how you have defined the new database in the new subschema and DMCL.

2.2 Parameters and Their Functions

Three parameter statements control CA-IDMS/DB Reorg processing and report production. The PROCESS and COPY parameter statements are mandatory; NOSWEEP statement is optional. Figure 2-1 on page 2-5 lists all CA-IDMS/DB Reorg parameters.

2.2.1 PROCESS Statement

PROCESS initiates CA-IDMS/DB Reorg processing, controls selection of the old and new subschemas and DMCLs allow you to execute the utility as one or more operating system job steps. PROCESS also allows you to establish a run-time Status Display at the operator console (and at specified TSO terminals) and to request Restart Information Messages.

2.2.2 COPY Statement

COPY identifies an area which is to be reorganized, permits ranking of records for storage allocations within the area, controls record fragmentation of CALC or sorted VIA records, and allows you to specify how many bytes are to be reserved for future additions of VIA records.

2.2.3 NOSWEEP Statement

NOSWEEP prevents an area sweep from being performed against an area that has set linkage with areas being reorganized. This helps decrease run time when very few set linkages exist between the NOSWEEP area and other areas being reorganized.

OLDDMCL = *dmcl-name*, NEWDMCL = *dmcl-name*
 [, OLDDBN = *dbname*] [, NEWDBN = *dbname*]

<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; vertical-align: middle;">.RPT =</td> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 5px;"> NEXT OPTALLOC PRIALLOC SECALLOC MATCH IIXSDBK UPDLINK STORE </td> </tr> </table>	.RPT =	NEXT OPTALLOC PRIALLOC SECALLOC MATCH IIXSDBK UPDLINK STORE	<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; vertical-align: middle;">.RESTART =</td> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 5px;"> NEXT OPTALLOC PRIALLOC SECALLOC MATCH IIXSDBK UPDLINK STORE </td> </tr> </table>	.RESTART =	NEXT OPTALLOC PRIALLOC SECALLOC MATCH IIXSDBK UPDLINK STORE
.RPT =	NEXT OPTALLOC PRIALLOC SECALLOC MATCH IIXSDBK UPDLINK STORE				
.RESTART =	NEXT OPTALLOC PRIALLOC SECALLOC MATCH IIXSDBK UPDLINK STORE				

<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; vertical-align: middle;">.STOPAFTER =</td> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 5px;"> CURRENT ANALYSIS OPTALLOC PRIALLOC SECALLOC MATCH IIXSDBK UPDLINK STORE EOR </td> </tr> </table>	.STOPAFTER =	CURRENT ANALYSIS OPTALLOC PRIALLOC SECALLOC MATCH IIXSDBK UPDLINK STORE EOR	<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; vertical-align: middle;">.DSPLYINT =</td> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 5px;"> <i>interval</i> NO REQUEST STEP </td> </tr> <tr> <td></td> <td style="padding: 5px;"> [, INFOROUTE = <i>route-code</i>] [, REPLYROUTE] </td> </tr> </table>	.DSPLYINT =	<i>interval</i> NO REQUEST STEP		[, INFOROUTE = <i>route-code</i>] [, REPLYROUTE]
.STOPAFTER =	CURRENT ANALYSIS OPTALLOC PRIALLOC SECALLOC MATCH IIXSDBK UPDLINK STORE EOR						
.DSPLYINT =	<i>interval</i> NO REQUEST STEP						
	[, INFOROUTE = <i>route-code</i>] [, REPLYROUTE]						

[, ISOUSERS = (*user-id1, user-id2, ... user-id10*)] [, BLOCKNUM = *block-size*]

<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; vertical-align: middle;">.SORTFLAG =</td> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 5px;"> ALL CRITICAL NONE </td> </tr> </table>	.SORTFLAG =	ALL CRITICAL NONE	[, SORTSIZE = <i>number-of-k</i>]
.SORTFLAG =	ALL CRITICAL NONE		

COPY_AREA = <i>area-name</i>	<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; vertical-align: middle;">.PRIORITY =</td> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 5px;"> DIRECT CALC <i>record-name</i> (<i>sublist-of-record-names</i>) </td> </tr> </table>	.PRIORITY =	DIRECT CALC <i>record-name</i> (<i>sublist-of-record-names</i>)
.PRIORITY =	DIRECT CALC <i>record-name</i> (<i>sublist-of-record-names</i>)		

[, TARGETROOT] [, PAGERESERVE = *number-of-bytes*]
 [, NOSWEEP, AREA = *area-name*]

Figure 2-1. CA-IDMS/DB Reorg Parameter Summary

2.3 PROCESS Statement

The PROCESS statement is required; it must be the first statement entered. The PROCESS statement:

- Initiates execution of CA-IDMS/DB Reorg
- Identifies both the old subschema, DMCL, and database name that describes the database before reorganization and the new subschema, DMCL, and database name that describes the database after reorganization
- Allows you to interrupt and restart processing; this makes it possible to execute the utility as one or more operating system job steps
- Establishes the Console Communication Facility at the operator console (and at specified TSO terminals)
- Allows you to route messages to selected functional areas if you are operating in a OS/390 environment with multiple console support
- Allows a request for Restart Information Messages
- Specifies the number of buffers to be used for database access.

```

PROCESS, OLDSUB = subschema - name, NEWSUB = subschema - name,
OLDDMCL = dmcl - name, NEWDMCL = dmcl - name
[ , OLDDBN = dbname ] [ , NEWDBN = dbname ]

[ , RPT = {
  NEXT
  OPTALLOC
  PRIALLOC
  SECALLOC
  MATCH
  IIXSDBK
  UPDLINK
  STORE
} ]

[ , RESTART = {
  NEXT
  OPTALLOC
  PRIALLOC
  SECALLOC
  MATCH
  IIXSDBK
  UPDLINK
  STORE
} ]

[ , STOPAFTER = {
  CURRENT
  ANALYSIS
  OPTALLOC
  PRIALLOC
  SECALLOC
  MATCH
  IIXSDBK
  UPDLINK
  STORE
  EOR
} ]

[ , DSPLYINT = {
  interval
  NO
  REQUEST
  STEP
} ]

[ , INFOROUTE = route - code ]
[ , REPLYROUTE ]

[ , TSUSERS = (user - id1, user - id2, ... user - id10) ] [ , BLOCKNUM = block - size ]

[ , SORTFLAG = {
  ALL
  CRITICAL
  NONE
} ] [ , SORTSIZE = number - of - k ]

```

Figure 2-2. PROCESS Statement Syntax

2.3.1 PROCESS Parameter

PROCESS

Initiates execution of CA-IDMS/DB Reorg and indicates that processing options follow.

2.3.2 OLDSUB Parameter

,OLDSUB=*subschema-name*

This is a required parameter identifying the subschema describing the database to be reorganized during the current run of CA-IDMS/DB Reorg. The subschema identified contains descriptions of all area, records, and sets that will be affected by the reorganization. All records must be assigned to the same page group.

2.3.3 NEWSUB Parameter

,NEWSUB=*subschema-name*

This is a required parameter identifying the subschema describing the database after reorganization. This subschema contains the new definitions for all areas, records, and sets affected by the reorganization.

Rules:

- The subschemas must exist in the STEPLIB/JOBLIB/Core Image Library. CA-IDMS/DB Reorg accesses and loads the subschemas from the load/core image library.
- The database described by a subschema named in the PROCESS statement cannot contain native VSAM (Virtual Storage Accessing Method) records.
- All records in the new subschema must be assigned to the same page group.

2.3.4 OLDDMCL Parameter

,OLDDMCL=*dmcl-name*

OLDDMCL is a required parameter that identifies the DMCL that is related to the old subschema and describes the database to be reorganized during the current run of CA-IDMS/DB Reorg.

2.3.5 NEWDMCL Parameter

NEWDMCL=*dmcl-name*

NEWDMCL is a required parameter that identifies the DMCL that is related to the new subschema and describes the database after reorganization.

Rules:

- The DMCLs must exist in the STEPLIB/JOBLIB/Core Image Library. CA-IDMS/DB Reorg accesses and loads the DMCLs from the load/core image library.

2.3.6 OLDDBN Parameter

[**OLDDBN**=*database-name*]

OLDDBN is an optional parameter specifying the segment name in the DMCL for the database before reorganization. If you do not specify the OLDDBN, the dbname in the SYSIDMS statement is used.

Rules:

- Segment names specified in the DMCL must also be defined in the DBNAME table.

2.3.7 NEWDBN Parameter

[**NEWDBN**=*database-name*]

NEWDBN is an optional parameter specifying the segment name in the DMCL for the database after reorganization. If you do not specify the NEWDBN, the dbname in the SYSIDMS statement is used.

Rules:

- Segment names specified in the DMCL must also be defined in the DBNAME table.

2.3.8 RPT Statement

The RPT parameter is optional. It is used to request a report that contains Restart Information Messages. These messages contain the information necessary to restart CA-IDMS/DB Reorg at a given step. RPT instructs CA-IDMS/DB Reorg to produce only restart information without executing any reorganization steps.

When to request restart information--Restart messages are needed if you have experienced a system failure or the operator has canceled CA-IDMS/DB Reorg mid-run without using the CA-IDMS/DB Reorg CANCEL command from the console. You may also want to request restart messages to help you reexecute a step if intermediate

work files were accidentally damaged or scratched in a previous run. The information from these messages will help you restart CA-IDMS/DB Reorg correctly.

More information on Restart Information Messages appears in 4.3.3, “Restart Information Messages” on page 4-10. Figure 4-5 on page 4-11 contains sample Restart Information Messages.

2.3.8.1 RPT Syntax and Options

```
[,RPT= {NEXT
        OPTALLOC
        PRIALLOC
        SECALLOC
        MATCH
        IIXSDBK
        UPDLINK
        STORE}]
```

This option indicates that a restart report option follows. The step name you enter after RPT= indicates that you want Restart Information Messages containing information necessary to restart the utility at that step. Restart messages for a particular step can only be produced if CA-IDMS/DB Reorg has completed processing all steps which precede the step requested (the steps are covered in Chapter 3, “Concepts” on page 3-1).

NEXT

Requests messages about restarting at the next logical reorganization step based upon the last step to have completed successfully.

OPTALLOC

Requests messages about restarting at the Optimum Allocation Pointer Extract (OPTALLOC) Step.

PRIALLOC

Requests messages about restarting at the Primary Allocation (PRIALLOC) Step.

SECALLOC

Requests messages about restarting at the Secondary Allocation (SECALLOC) Step.

MATCH

Requests messages about restarting at the Match (MATCH) Step.

IIXSDBK

Requests that messages about restarting at the IIX sorted dbkey (IIXSDBK) Step.

UPDLINK

Requests messages about restarting at the Update Set Linkage (UPDLINK) Step.

STORE

Requests messages about restarting at the Store (STORE) Step.

Default: None. If you enter the RPT parameter you must choose one of the options.

Rule:

- All steps prior to the step requested must have been previously executed and ended successfully.

Note: When a PROCESS statement contains the RPT parameter all other parameters are ignored. To resume processing you must initiate another run of CA-IDMS/DB Reorg without the RPT parameter.

2.3.9 RESTART Parameter

RESTART is an optional parameter that allows you to return to a reorganization and resume processing at the step you specify. Restart is required if you have used the STOPAFTER parameter to halt processing.

With the RESTART parameter you can:

Recover from a system failure--In the event of a system failure, you can restart CA-IDMS/DB Reorg at the reorganization step that was executing when the failure occurred by specifying RESTART=NEXT (or a particular step name) instead of having to rerun the entire utility. CA-IDMS/DB Reorg will continue processing reorganization steps in their logical sequence (see Chapter 3, “Concepts” on page 3-1 for information about the reorganization steps).

Re-create files--When intermediate files are accidentally damaged, scratched, or lost during a system failure, you can re-create those files by reexecuting steps that created the files originally.

Because each reorganization step must be executed in its logical sequence, you can restart CA-IDMS/DB Reorg in one of these ways:

- Enter the keyword NEXT
- Supply the name of the next logical step in the reorganization sequence (such as MATCH)
- Supply the name of a step (in the reorganization sequence) that has already executed successfully.

Note: If the UPDLINK step has begun or completed execution, the database has been modified and you cannot restart at or before the OPTALLOC step until you restore the areas that were updated.

2.3.9.1 RESTART Syntax and Options

```
[,RESTART= {NEXT  
OPTALLOC  
PRIALLOC  
SECALLOC  
MATCH  
IIXSDBK  
UPDLINK  
STORE}]
```

This option indicates that a RESTART option follows. You select the reorganization step at which CA-IDMS/DB Reorg will resume execution (steps are detailed in Chapter 3, “Concepts” on page 3-1).

NEXT

Resumes execution at the next logical reorganization step in the sequence. CA-IDMS/DB Reorg automatically determines which reorganization step is next and resumes execution. Reorganization continues until all reorganization steps have executed successfully or until the step specified in the STOPAFTER parameter has completed.

OPTALLOC

Resumes with the Optimum Allocation Pointer Extract (OPTALLOC) step.

PRIALLOC

Resumes with the Primary Allocation (PRIALLOC) step.

SECALLOC

Resumes with the Secondary Allocation (SECALLOC) step.

MATCH

Resumes with the Match (MATCH) step.

IIXSDBK

Resumes with the IIX sorted dbkey (IIXSDBK) step.

UPDLINK

Resumes with the Update Set Linkage (UPDLINK) step.

STORE

Resumes with the Store (STORE) step.

Default: None. If you enter the RESTART parameter you must choose one of the options.

Rules:

- The RESTART parameter can only be used after the first reorganization step (the Analysis step) has executed successfully.

- To successfully restart CA-IDMS/DB Reorg, all steps preceding the step at which you are restarting must have executed successfully.
- The RESTART parameter must be used to resume processing after the STOPAFTER parameter was used to halt processing.

2.3.10 STOPAFTER Parameter

The STOPAFTER parameter is optional. It allows you to interrupt the reorganization process. By selecting a step name, you identify the last step to be completed before processing stops.

2.3.10.1 STOPAFTER Syntax and Options

```
[,STOPAFTER= {CURRENT  
ANALYSIS  
OPTALLOC  
PRIALLOC  
SECALLOC  
MATCH  
IIXSDBK  
UPDLINK  
STORE  
EOR←}]
```

This option indicates that a STOPAFTER option follows. You select the reorganization step to be completed before processing stops. Chapter 3, “Concepts” on page 3-1 explains the function of each step.

Note: Once execution begins, an alternative to STOPAFTER is available. You can enter a Console Communication Command at the operator console to request a halt in execution after the current step is completed. You must have specified the DSPLYINT parameter in the current run. See 2.3.12, “DSPLYINT Parameter” on page 2-14 and 2.6, “Console Communications Facility” on page 2-25.

CURRENT

Stop execution after the current logical step in the reorganization sequence.

ANALYSIS

Stop execution after the Analysis (ANALYSIS) step has been completed.

OPTALLOC

Stop execution after the Optimum Allocation Pointer Extract (OPTALLOC) step has been completed.

PRIALLOC

Stop execution after the Primary Allocation (PRIALLOC) step has been completed.

SECALLOC

Stop execution after the Secondary Allocation (SECALLOC) step has been completed.

MATCH

Stop execution after the Match (MATCH) step has been completed.

IIXDBK

Stop execution after the IIX sorted db key (IIXSDBK) step.

UPDLINK

Stop execution after the Update Set Linkage (UPDLINK) step has been completed.

STORE

Stop execution after the Store (STORE) step has been completed.

EOR

Proceed to the end of the reorganization. When EOR (end of reorganization) is requested, CA-IDMS/DB Reorg will execute all remaining reorganization steps in sequence.

Default: EOR

Rule:

- The step name specified for STOPAFTER cannot precede (in execution sequence) the step name specified for RESTART.

2.3.11 Use RESTART with STOPAFTER

You can also use the RESTART parameter in conjunction with the STOPAFTER parameter to perform a reorganization in increments. The planned pause between reorganization steps gives you time to back up files, manage disk space, or to perform any other necessary operations functions.

For example, you can stop processing after the UPDLINK step has completed by using the STOPAFTER parameter; then use RESTART to resume execution at the next step, STORE. Figure 2-3 on page 2-14 illustrates one possible sequence of operations using STOPAFTER and RESTART. The use of STOPAFTER and RESTART is optional. However, if STOPAFTER is used to halt processing, RESTART is required to resume processing.

Perform backup procedures--For users who need to conform with operations standards and want to reorganize a large area or a large database, the stop/restart feature allows you to stop the reorganization process to back up work files between reorganization steps.

Interrupt processing to perform operations functions-- The stop/restart feature was also designed to accommodate scheduled data processing operations functions that require the entire machine. CA-IDMS/DB Reorg allows you to interrupt its processing to re-IPL, to back up disks, or to perform other operations functions.

Manage disk space effectively--With CA-IDMS/DB Reorg's convenient stop/restart feature, you can interrupt processing and run CA-IDMS/DB Reorg in two or more parts. This allows you to allocate space for files as needed, rather than allocating all data sets at the beginning of the run. In this way, you can execute CA-IDMS/DB Reorg in multiple steps to allow the new database to use disk space that was used by the old database.

```
3) PROCESS, OLDSUB= subschemaname,
   NEWSUB= subschemaname,
   OLDDMCL= dmclname,
   NEWDMCL= dmclname,
   RESTART=NEXT
```

Figure 2-3. Examples of STOPAFTER and RESTART

2.3.12 DSPLYINT Parameter

The DSPLYINT (display interval) parameter establishes the Console Communication Facility and indicates the times at which processing information will be displayed at the operator console and specified TSO terminals.

The Console Communication Facility is controlled by dynamic commands entered from the console. Unless you decide to turn the facility off (by specifying DSPLYINT=NO), you can enter dynamic commands at the operator console to change the status interval previously selected by parameter. The Console Communication Facility and its commands are described at the end of this section. Be sure that your system console operator is aware of valid reply choices.

You can send the status display to a selected functional area by using the INFORROUTE parameter or the REPLYROUTE parameter.

2.3.12.1 DSPLYINT Syntax and Options

```
[,DSPLYINT= {interval
             NO
             REQUEST
             STEP←}]
```

interval

Specifies the time interval (in minutes) at which the status is displayed at the operator console (and specified TSO terminals). Status information is also displayed when each step starts and ends. The minimum interval is one minute; the maximum interval is 120 minutes. The timing of a display interval begins each time a step begins. You can change this interval at run time by entering the SET INTERVAL command at the operator console.

You can enter a DISPLAY STATUS command at the operator console without affecting the specified display interval.

NO

Specifies that the status display is to be turned off for the current run of CA-IDMS/DB Reorg.

Rule:

- If you specify DSPLYINT=NO, you cannot enter dynamic commands at the operator console.

Note: For VSE/ESA users only--Enter DSPLYINT=NO if STXIT OC support is not generated in the supervisor.

REQUEST

Specifies that you want reorganization processing information on demand and allows you to request the information by using the DISPLAY STATUS command at the operator console.

DSPLYINT=REQUEST can be changed during execution of CA-IDMS/DB Reorg by entering a SET INTERVAL command at the operator console. See 2.6, "Console Communications Facility" on page 2-25.

STEP

Specifies that you do not want a periodic display and that status information is to be displayed automatically after the completion of each individual reorganization step.

DSPLYINT=STEP can be changed during execution of CA-IDMS/DB Reorg by entering the SET INTERVAL command at the operator console. See 2.6, "Console Communications Facility" on page 2-25.

If you specify DSPLYINT=STEP, you can enter a DISPLAY STATUS command at the operator console without affecting the display at the end of each step.

Default: STEP.

2.3.12.2 Route Code Syntax and Parameters

For an OS/390 environment with multiple console support, status displays and status messages requiring replies can be routed to selected functional areas by using the optional INFOROUTE and REPLYROUTE parameters.

[,INFOROUTE=*route-code*]

[,REPLYROUTE=*route-code*]

INFOROUTE

Indicates the route code for WTO messages.

REPLYROUTE

Indicates the route code for WTOR messages.

route-code

Identifies the functional area that is to receive the status display at the intervals specified in the DSPLYINT parameter. Valid route codes for the various areas are listed in Table 2-1.

Defaults: The default value for INFOROUTE is 2. The default value for REPLYROUTE is 1.

Rules:

- INFOROUTE and REPLYROUTE function only under OS/390.
- Do not enter INFOROUTE or REPLYROUTE if DSPLYINT=NO.
- Specify only one route code for INFOROUTE and/or one route code for REPLYROUTE.
- Code number 11 is reserved for internal use only. All messages will automatically be routed for programmer information in addition to any route specified by using INFOROUTE or REPLYROUTE.

Table 2-1. Route Codes for Status Display Locations

Code	Functional Area
1	Master Console Action
2	Master Console Information
3	Tape Pool
4	Direct Access Pool
5	Tape Library
6	Disk Library
7	Unit Record Pool
8	Teleprocessing Control
9	System Security
10	System Error/Maintenance
12	Emulator Information
13	User Routing Code
14	User Routing Code
15	User Routing Code
16	(reserved for future use)

2.3.13 TSOUSERS Parameter

[,TSOUSERS=(*user-id1,user-id2,...,user-id10*)]

The TSOUSERS (TSO users) parameter controls the display of status information at TSO terminals. All status displays appearing at the operator console may also be directed to selected TSO user IDs.

Rules:

- TSO users must be logged on to receive the status displays.
- You can specify up to 10 TSO identifiers.
- TSOUSERS functions only under OS/390.

2.3.14 SORTFLAG Parameter

The SORTFLAG parameter controls the level of sort message reporting during the internal sorting done by CA-IDMS/DB Reorg.

2.3.14.1 SORTFLAG Syntax and Options

[,SORTFLAG= {ALL
CRITICAL←
STEP}]

Default: The default is CRITICAL

2.3.15 SORTSIZE Parameter

[,SORTSIZE = *number-of-k*]

The SORTSIZE Parameter controls the numbers of bytes to be reserved for the internal sort process.

CA-IDMS/DB Reorg converts the number-of-kilobytes specified into the number-of-bytes by multiplying by 1024.

Default: The default value is 200K.

2.3.16 BLOCKNUM Parameter

[,BLOCKNUM=*block-size*]

The BLOCKNUM Parameter controls blocking factor for most of the workfiles used by CA-IDMS/DB Reorg. You cannot specify a blocking factor for the CNTRL1, CNTRL2 and PAGUTIL files.

Rule:

- The block-size value must be from 1 through 32.
- For work files stored on tape, use 32.
- For work files stored on disk, use a number that represents half track blocking.

2.4 COPY Statement

The COPY statement is required; at least one COPY statement must be entered along with the PROCESS statement (unless RPT or RESTART is specified on the PROCESS statement).

Use the COPY statement to:

- Identify an area to be reorganized.
- Establish the priority for storing records on their target pages.
- Control fragmentation of CALC records and sorted VIA records, if fragmentation is necessary.
- Reserve space for future insertions of VIA records.

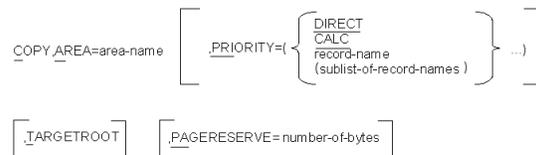


Figure 2-4. COPY Statement Syntax

2.4.1 COPY Parameter

COPY

Initiates copy statement processing and indicates that an area name parameter and other options will follow.

2.4.2 AREA Parameter

AREA=area-name

Area is a required parameter that identifies an area to be reorganized by CA-IDMS/DB Reorg. To reorganize more than one area, enter a separate COPY statement for each area that is to be reorganized.

Default: None. If you want to reorganize an area, you must supply an area name.

Rules:

- The area named must be one that exists within the old subschema and the new subschema.
- Supply at least one COPY statement for each run of CA-IDMS/DB Reorg unless RPT or RESTART is specified on the PROCESS statement.
- If either RPT or RESTART is specified on the PROCESS statement, the COPY statement will be ignored (COPY will be treated as a comment).
- A COPY statement can only be specified once for a given area.

2.4.3 PRIORITY Parameter

The optional PRIORITY parameter allows you to specify the order in which records will be assigned to their target pages.

To establish a priority for records, list the record names in the order they are to receive priority (highest to lowest). CA-IDMS/DB Reorg will store the highest priority records on their target pages before storing any other records. The second priority listed will take precedence over the third listed, and so on.

2.4.3.1 Specify Record Type Priority by Class or Sublist

You can specify priority for a record type by class (CALC or DIRECT) and you can rank records by record type (you can mix these priority methods).

When you list CALC or DIRECT in the PRIORITY parameter, all CALC or DIRECT records not specifically named have equal priority. However, individual record types belonging to the CALC or DIRECT class can be specified and assigned a different priority. For example, if CALC records have highest priority, you can still place an individual CALC record in a lower priority position.

For example, when you specify:

PRIORITY=(REC1,REC2,CALC,(REC3,REC4,REC5),REC6)

- REC1 receives first priority
- REC2 receives second priority
- Records stored CALC (except for records 1-6) receive third priority
- REC3-REC5 receive fourth priority
- REC6 receives fifth priority.

Because DIRECT was not specified, all DIRECT records have sixth priority and VIA records (which cannot be specified as a class) have seventh priority. See Chapter 3, “Concepts” on page 3-1 for further discussion of ranking records.

You can also assign a group of record types the same priority. In the previous example, REC3, REC4, and REC5 are assigned equal priority. You can use this method of ranking to specify equal rank for all members of a multi-member set.

2.4.3.2 PRIORITY Syntax and Options

```
[,PRIORITY= ({DIRECT
              CALC
              record-name
              (sublist-of-record names})
              ...)
```

PRIORITY=

Indicates that priority options will be listed in the order they are to receive priority.

DIRECT

Depending on the priority position of DIRECT in the sequence, DIRECT records will be located on their target page before any other records with a lower priority are stored.

CALC

Depending on the priority position of CALC in the sequence, CALC records will be located on their target page before any other records with a lower priority are stored.

record-name

The records of this name will be located on their target pages before other records with lower priority.

(sublist-of-record-names)

If you enter a list of record types, each record type will receive storage priority of equal value. A sublist must be enclosed in parentheses and each record name must be separated by a comma.

Note: If both an owner and member record of the same set are specified in a sublist, and both record types have a location mode of VIA, then all the owner records are clustered together followed by their member records.

Default: Priority is DIRECT, CALC, and VIA records after any user specified priorities.

Rules:

- List PRIORITY options in order from highest to lowest.
- Use DIRECT, CALC, or a specific record name only once in the PRIORITY statement.
- You can specify up to 500 individual record names for the PRIORITY parameter.
- Record-name must be the name of a record within OLDSUB and NEWSUB.

2.4.4 PAGERESERVE Parameter

[,PAGERESERVE=*number-of-bytes*]

PAGERESERVE is an optional parameter that pertains only to VIA records and records stored physically sequential. CA-IDMS/DB Reorg will reserve part of each page for future insertions of records within the area specified in the COPY statement. PAGERESERVE specifies the number of bytes to be left free in each page. CA-IDMS/DB Reorg ignores the page reserve value coded into the subschema from the DMCL.

Default: The default value is zero (no reserve).

2.4.5 TARGETROOT Parameter

[,TARGETROOT]

The optional TARGETROOT parameter pertains only to variable length records stored CALC or as members of a sorted VIA set. It allows you to control the degree of fragmentation of these types of records if fragmentation is necessary.

Indicates that for CALC and sorted VIA records, CA-IDMS/DB Reorg will store the record's root on the target page and fragment the rest of the record when the entire record will not fit on the target page. The root will be at least as large as the minimum ROOT defined in the new subschema and will be as large as will fit on the target page. This applies only to variable-length records.

TARGETROOT keeps the record's key on the target page, which makes it possible for CA-IDMS to search the CALC or sorted set without having to access other database pages. This reduces the number of I/Os when the CALC or sorted VIA set is being searched.

2.5 NOSWEEP Statement

NOSWEEP is an optional parameter statement that prevents an area sweep from being performed against the non-copy area you identify.

The NOSWEEP parameter decreases CA-IDMS/DB Reorg execution time when it is used for non-copy areas that contain few set occurrences with cross-area linkage. When NOSWEEP is used, CA-IDMS/DB Reorg will resolve pointers between areas by creating extracts during set walking, or by using NEXT and PRIOR pointers to establish set linkage with the area being reorganized.

2.5.1 When to Use NOSWEEP

To determine the number of record occurrences with cross-area set linkage, run CA-IDMS/DB Analyzer and review the Record Reports. If the number of record occurrences with set linkage between the copy area and the non-copy area is less than the total number of pages in the non-copy area, use the NOSWEEP statement.

During the first reorganization step (the Analysis Step), CA-IDMS/DB Reorg determines whether it can create pointers for records in a copy area that have set linkage with records in a non-copy area, without sweeping every area. If pointers can be created, CA-IDMS/DB Reorg will automatically set the NOSWEEP parameter.

2.5.2 NOSWEEP Statement Syntax

[NOSWEEP,AREA=*area-name*]

where:

NOSWEEP

Initiates NOSWEEP statement processing and indicates that an area name parameter will follow.

,AREA=*area-name*

Identifies the non-copy area for which an area sweep is to be prevented.

Default: If you do not enter the name of an area, CA-IDMS/DB Reorg usually sweeps all areas that have cross-area set linkage with the area being reorganized.

Rules:

- Area-name must be name of an area within OLDSUB and NEWSUB.
- The area-name specified must not be an area-name specified in a COPY statement.
- The area-name specified must have set linkage with an area being copied.
- A NOSWEEP statement can only be specified once for a given area.

- The area-name specified must not contain records that are linked (as members) to an SR8 set in a COPY area.

2.6 Console Communications Facility

The CA-IDMS/DB Reorg Console Communication Facility allows you to monitor the status of the reorganization process at the operator console and at specified TSO terminals. First, you must establish the Facility using the DSPLYINT parameter. You can enter dynamic console commands to control the facility from the operator console unless you selected DSPLYINT=NO.

Because CA-IDMS/DB Reorg runs with an outstanding operator reply, your system console operator should be aware of the valid reply choices. The reply can be ignored until you or the operator wants to communicate with CA-IDMS/DB Reorg.

2.6.1 Functions of Console Communication Commands

Four commands allow you to view status on demand and to stop CA-IDMS/DB Reorg after successful completion of the current step, to cancel CA-IDMS/DB Reorg immediately, or to change the interval at which status is currently being displayed (as established in the DSPLYINT parameter). The four commands are:

DISPLAY STATUS

SET INTERVAL {*interval*/OFF/REQUEST/STEP}

STOP

CANCEL [DUMP]

2.6.1.1 DISPLAY STATUS Command

DISPLAY STATUS

This command produces a status display on demand. The information displayed includes the name of the step being processed and its current status. This command has no effect on the value you entered with the DSPLYINT parameter.

2.6.1.2 SET INTERVAL Command

SET INTERVAL {*interval*/OFF/REQUEST/STEP}

With this command, you can establish, reset, or turn off the periodic display. SET INTERVAL can also be issued to change the option you selected for the DSPLYINT parameter.

interval

Requests an interval in minutes, where the interval is an integer between 1 and 120, inclusive. The timing of a display interval begins each time a step begins. If a periodic

display interval was already active (selected by using the DSPLYINT parameter), it will be canceled and reset to the new interval value. SET INTERVAL 0=SET INTERVAL OFF.

OFF

Cancels the current periodic display interval (if active) or the display at the completion of each step (if active). You can then request status or set a new interval.

REQUEST

Specifies that instead of a periodic display, you want status information to be displayed on demand. When REQUEST is used, a status display is produced only in response to the DISPLAY STATUS operator command. REQUEST cancels the current periodic display interval, if one was active. REQUEST also cancels status displays at step initiation and termination.

STEP

Cancels the current periodic display (if active) and tells CA-IDMS/DB Reorg to automatically display step initiation and termination statistics. If a SET INTERVAL *interval* command is issued after a SET INTERVAL STEP command, status information will still be displayed when each step starts and ends.

2.6.1.3 STOP Command

STOP

This command stops the reorganization process after CA-IDMS/DB Reorg has completed the current reorganization step. To resume reorganization, initiate a new run of CA-IDMS/DB Reorg by entering a PROCESS statement which includes a RESTART parameter. STOP is useful when you want to interrupt processing to IPL the system or to perform other operations tasks.

2.6.1.4 CANCEL Command

CANCEL [DUMP]

This command halts CA-IDMS/DB Reorg processing immediately, canceling the current reorganization step. Restart Information Messages that contain information necessary for restarting the utility will appear on the Audit Report. The DUMP part of the command is optional. It directs CA-IDMS/DB Reorg to produce a dump.

Chapter 3. Concepts

3.1 Overview	3-3
3.2 Understanding CA-IDMS/DB Reorg Processing	3-4
3.2.1 Reorganization--Not Restructuring	3-4
3.2.2 How Is CA-IDMS/DB Reorg Different From Other Utilities?	3-4
3.3 CA-IDMS/DB Reorg's Inputs	3-5
3.4 CA-IDMS/DB Reorg's Record Allocation Techniques	3-7
3.4.1 Minimize Record Overflow	3-7
3.4.2 Reduce Fragmented Records	3-7
3.4.3 Allocating DIRECT Records	3-7
3.4.4 Allocating CALC Records	3-8
3.4.5 Allocating VIA Records	3-9
3.4.6 Sorted VIA Records	3-10
3.4.7 Allocation and the PRIORITY Parameter	3-11
3.4.8 Allocation and Page Reserve	3-12
3.5 CA-IDMS/DB Reorg Reorganization Capabilities	3-14
3.5.1 Reorganization Capabilities Controlled by Parameters	3-14
3.5.2 Direct an Efficient Reorganization Through Parameters	3-16
3.5.3 Reorganization Capabilities Controlled by Subschema and DMCL Definition	3-17
3.5.4 Automatic Reorganization Controlled by Internal Logic	3-19
3.6 Reorganization Steps	3-21
3.6.1 Functions of the Reorganization Steps	3-23

3.1 Overview

This chapter provides detailed background information on how CA-IDMS/DB Reorg performs a reorganization through a step-by-step process. Each step and its function is discussed and illustrated. In addition, this chapter discusses CA-IDMS/DB Reorg's inputs and record allocation techniques that improve system performance.

3.2 Understanding CA-IDMS/DB Reorg Processing

To use CA-IDMS/DB Reorg effectively, you must understand how CA-IDMS/DB Reorg performs database reorganization--the inputs required, record allocation techniques integral to the system, and the step-by-step nature of the reorganization process.

3.2.1 Reorganization--Not Restructuring

CA-IDMS/DB Reorg was designed to reorganize a database and will not restructure it. CA-IDMS/DB Reorg will not alter relationships between records nor the size or content of records. CA-IDMS/DB Reorg does not decompress or compress records. Therefore the control length of variable or fixed compressed records cannot change. This means that CA-IDMS/DB Reorg processing increases application program efficiency without requiring program changes or recompilation.

3.2.2 How Is CA-IDMS/DB Reorg Different From Other Utilities?

While utilities available in the past were not designed as reorganization utilities, they are often used as major building blocks in a user-directed, multiple-utility approach to database reorganization.

CA-IDMS/DB Reorg was designed specifically as a reorganization utility. It eliminates the need for a time-consuming complete unload and reload of the database. If you want to reorganize an area that has set linkage with another area in the database, CA-IDMS/DB Reorg will update the set linkage without requiring a reorganization of the linked area. Also, logically deleted records (LDELS) do not have to be removed before reorganization. All of these tasks can be accomplished with one utility when you use CA-IDMS/DB Reorg.

In addition, many other true reorganization features were built into CA-IDMS/DB Reorg to optimize changing page ranges and page sizes, record placement, and reduction of I/Os and record fragmentation.

This section contains detailed information about CA-IDMS/DB Reorg's:

- Inputs
- Record allocation techniques
- Reorganization capabilities
- Reorganization steps.

3.3 CA-IDMS/DB Reorg's Inputs

Database reorganization is a complex process. However, many of CA-IDMS/DB Reorg's performance-tuning functions are performed automatically each time you run CA-IDMS/DB Reorg. You need to supply only the inputs to CA-IDMS/DB Reorg illustrated in Figure 3-1 on page 3-6

- **Parameters**--You supply input parameters to direct CA-IDMS/DB Reorg, to indicate which subschemas and DMCLs are to be used for reorganization, and to indicate which areas are to be copied. (Chapter 2, "Parameters and Commands" on page 2-1 contains complete details.).
- **Old and new subschemas and DMCLs**--You define the old and new subschemas and old and new DMCLs to show what the database looks like before reorganization and what its characteristics will be after reorganization occurs.
- **Old database**--You supply the old database--one that needs to be reorganized.
- **Commands**--You can enter status and control commands from the operator console while CA-IDMS/DB Reorg is executing. See 2.3.12, "DSPLYINT Parameter" on page 2-14.

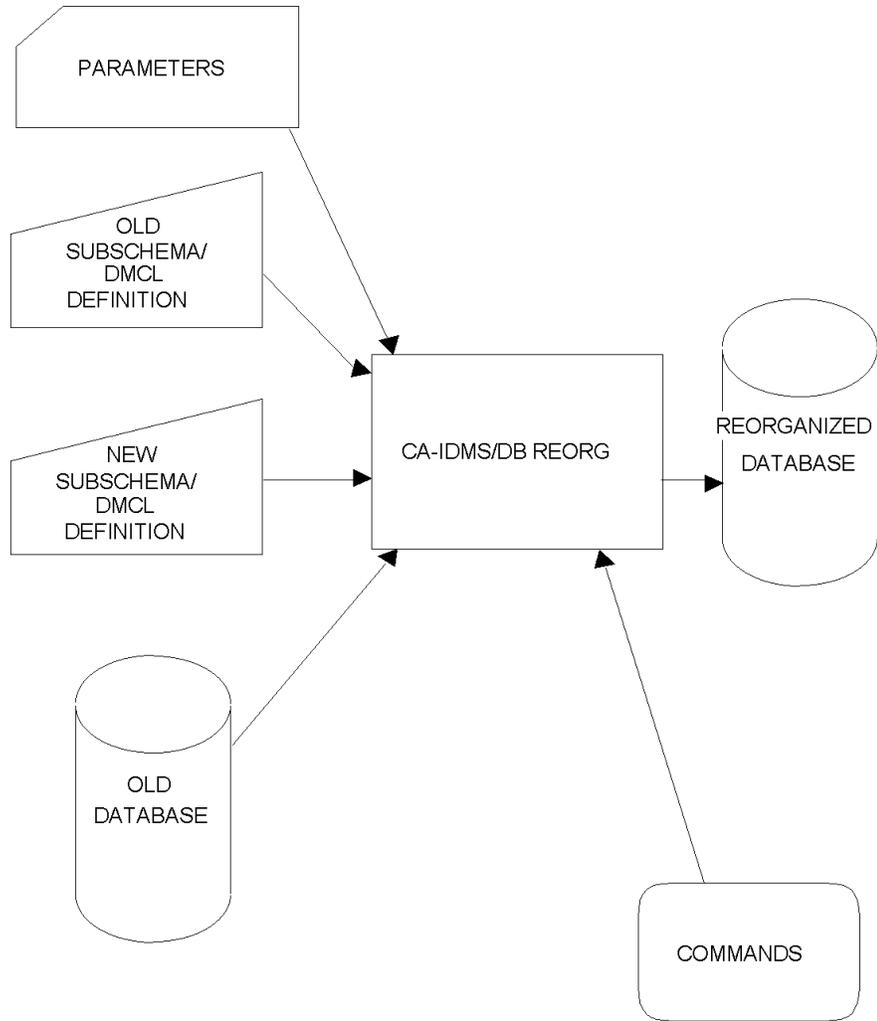


Figure 3-1. Inputs to CA-IDMS/DB Reorg

3.4 CA-IDMS/DB Reorg's Record Allocation Techniques

The record allocation techniques used by CA-IDMS/DB Reorg were specifically designed to improve performance. Record fragmentation is reduced and record overflow is kept to a minimum. As a result, CA-IDMS/DB performs fewer I/Os and response time improves significantly.

3.4.1 Minimize Record Overflow

CA-IDMS/DB Reorg will overflow a record only when its target page is full. Records that will not fit on their target pages are not stored on another target page until CA-IDMS/DB Reorg has finished allocating all other records to their assigned target pages. This method of postponing record overflow prevents a record that does not fit on its target page from displacing other records allocated to the following target pages. Once all records have been allocated to their target pages, CA-IDMS/DB Reorg then allocates the records that were held back by searching for a page with enough free space that is as close (sequentially) as possible following the record's target page.

3.4.2 Reduce Fragmented Records

CA-IDMS/DB Reorg fragments a variable length record only when the record is too large to be stored on the target page and there is no other page where the entire record will fit. The fragmentation is kept to a minimum by keeping record fragments as large as possible and storing them on as few pages as possible.

Record fragments are not allocated until all target page allocations have been made. This ensures that a fragment from one record will not cause a second record to fragment or overflow from its target page.

When an entire record will not fit on the target page, CA-IDMS/DB Reorg, under your direction, will create a minimum of one fragment for CALC and sorted VIA record types even though there is a page somewhere in the record's page range with enough free space to accommodate the complete record. The TARGETROOT parameter provides control in these situations. See 2.4.5, "TARGETROOT Parameter" on page 2-22.

3.4.3 Allocating DIRECT Records

There are two methods of allocating DIRECT records. The method used depends on the record's page range in the old and the new subschema.

When the page range is identical in both the old and the new database, a DIRECT record automatically targets to the page it was stored on in the old database. If the target page is full, CA-IDMS/DB Reorg will allocate the direct record to the next available page where the entire record will fit. If there is no page in the DIRECT record's page range and the record is variable length, CA-IDMS/DB Reorg will fragment the record.

When the new page range is different from the page range in the old database, a DIRECT record is allocated to the same relative position within that page range. In this way, the record is stored in the same relative location in the new database.

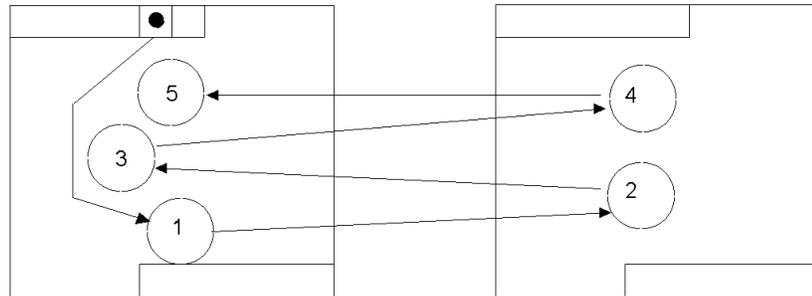
3.4.4 Allocating CALC Records

The target page for a CALC record is determined by the IDMSCALC routine. A CALC record will be stored on the target page unless that page is full. If the target page is full and the TARGETROOT parameter was not specified, CA-IDMS/DB Reorg will find the next available page in the record's page range that can accommodate the entire record.

If CALC records must overflow to another page, CA-IDMS/DB Reorg will overflow the CALC records with symbolic keys that are higher than those of records stored on the home page. This reduces the average number of I/Os required to retrieve CALC records. See Figure 3-2 on page 3-9.

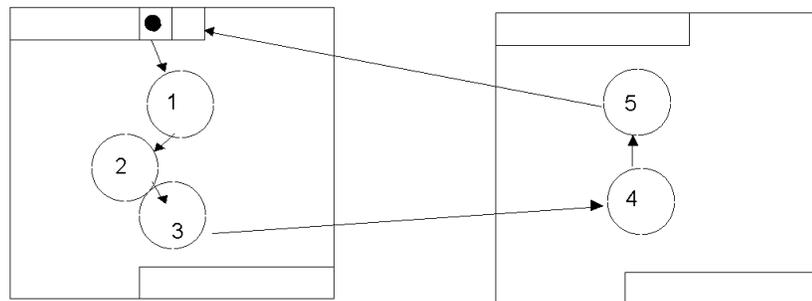
When a variable length CALC record does not fit on the target page, but the target page can accommodate the root portion and TARGETROOT parameter was specified, CA-IDMS/DB Reorg will store the record in two parts. The root is stored on the target page and the rest of the record (the record fragment) is stored on the first available page which will fit the entire fragment.

In the database, duplicate CALC keys will maintain the same order as in the old database.



BEFORE CA-IDMS/DB REORG

Before CA-IDMS/DB REORG, three I/O operations are required to retrieve the CALC record with the symbolic key 3. (Maximum number of records per page = 3.)



AFTER CA-IDMS/DB REORG

After the database is reorganized using CA-IDMS/DB REORG, only one I/O is required to retrieve the CALC record with the symbolic key of 3.

Figure 3-2. Optimizing the CALC Set

3.4.5 Allocating VIA Records

VIA records that are stored on the owner page are allocated when the owner record is allocated. If a record is stored at a displacement from its owner page or if it is stored in a different page range than its owner, the VIA record will be allocated after all primary allocations have been made.

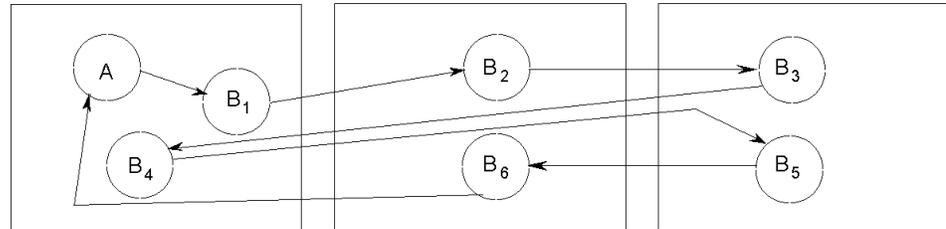
VIA records are also allocated in logical set order (the physical and logical order will be the same). This can reduce the number of I/Os required to walk the set. See Figure 3-3 on page 3-11.

In an active online environment where buffer use is low, the number of I/Os to walk a VIA set will be the same as the number of page changes encountered when walking the set. The simple example in Figure 3-3 on page 3-11 illustrates that prior to reorganization, the set was spread across 3 pages and it took 6 page changes (6 I/Os) to walk the set. After CA-IDMS/DB Reorg, the set is still spread across three pages, but it only takes three page changes (3 I/Os) to walk the set. The net result is a 50 percent savings in I/O.

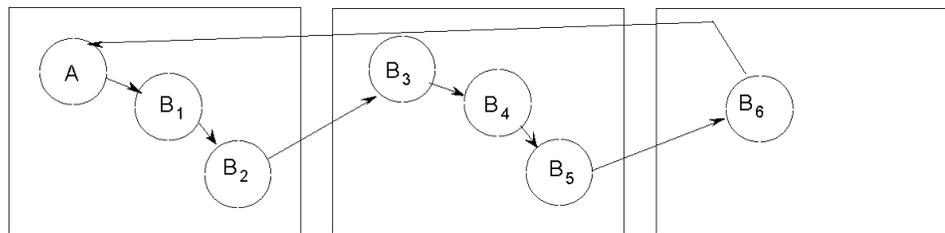
3.4.6 Sorted VIA Records

CA-IDMS/DB Reorg physically places sorted set members in key sequence, keeping physical and logical clustering of set members the same. This reduces the number of I/Os required to walk the set and to retrieve a sorted member when a FIND/OBTAIN USING command is specified (see Figure 3-3 on page 3-11). CA-IDMS/DB Reorg will overflow the records with symbolic keys that are higher than those of records stored on the home page. This reduces the average number of I/Os required to search for a record occurrence.

A variable length, sorted VIA record that does not fit on its owner's page will have its root portion stored there if the TARGETROOT parameter was specified. This helps reduce the average number of I/Os required to retrieve a sorted VIA member when using the FIND/OBTAIN USING command because CA-IDMS/DB only looks at the root. The TARGETROOT parameter is most useful when the sorted VIA records are large with small roots. See 2.4.5, "TARGETROOT Parameter" on page 2-22.

**BEFORE CA-IDMS/DB REORG**

When the logical and physical VIA set order are not the same, excessive I/O operations are performed. In this example, six I/O operations are required for CA-IDMS/DB to walk the VIA set. (Maximum number of records per page = 3.)

**AFTER CA-IDMS/DB REORG**

After reorganization with CA-IDMS/DB REORG, the logical and physical VIA set order are identical. Only three I/Os are needed to walk the VIA set.

Figure 3-3. Optimizing VIA Records

3.4.7 Allocation and the PRIORITY Parameter

The PRIORITY parameter takes effect when more records target to a page than that page can hold. When this happens, the priorities you assigned to record types are used to determine which records (or group of record types) will be stored on the target page and which ones should overflow if the page is full.

The sublist feature of the PRIORITY parameter allows you to specify that two or more record types have the same priority, and it allows you to specify that VIA members

and their non-VIA owner are to be stored in a cluster. However, if the owner location mode is VIA, then all the owner records are clustered together followed by their member records. In the example at the bottom of Figure 3-4 on page 3-13, the type B records are clustered near their type A owners.

When you rank record types using the `PRIORITY` parameter, CA-IDMS/DB Reorg will accommodate your priority requests by altering its allocation techniques as needed. CA-IDMS/DB Reorg's default priority order is `DIRECT`, `CALC`, and `VIA`.

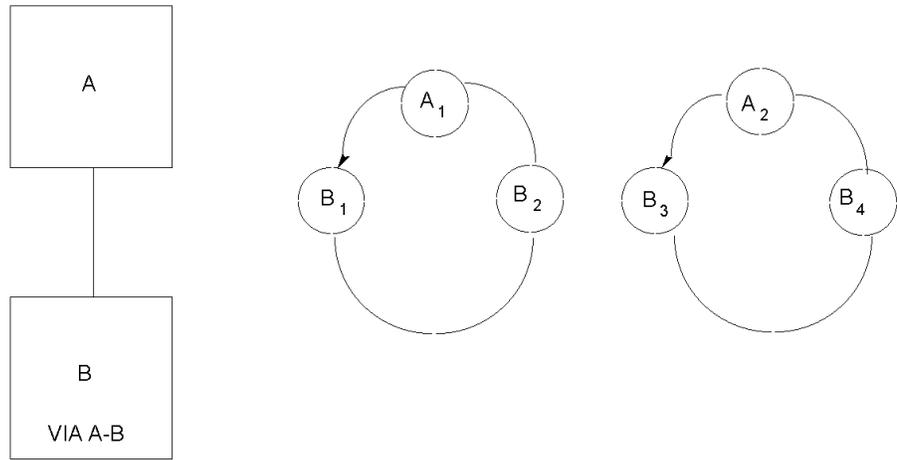
You can rank individual records or groups of record types. For example, if you specify a sublist of record types as first priority, and `DIRECT` records as second priority, you are overriding CA-IDMS/DB Reorg's automatic record allocation techniques. By using the `PRIORITY` parameter you can tailor the physical structure of the database to meet your needs, based on your processing requirements.

3.4.8 Allocation and Page Reserve

The page reserve value specified by parameter affects only allocation of `VIA` and physical sequential records. (For more information on the `PAGERESERVE` parameter, see 2.4.4, "PAGERESERVE Parameter" on page 2-22.)

The page reserve is ignored for `CALC` records because CA-IDMS/DB Reorg will not overflow a `CALC` record unless absolutely necessary. This conserves I/Os significantly because every `CALC` record that overflows requires at least two I/Os of system overhead.

This parameter-defined page reserve overrides the value coded into the `DMCL`. This means you do not have to define a separate `DMCL` with a page reserve to run CA-IDMS/DB Reorg. In addition, it is not necessary to recompile to resume normal database operations after CA-IDMS/DB Reorg has executed.



EXAMPLE 1: Each record type is given a different priority. PRIORITY = (A,B)



EXAMPLE 2: Each record type is given an equal priority. PRIORITY = ((A,B))



Figure 3-4. User-Defined Clustering

3.5 CA-IDMS/DB Reorg Reorganization Capabilities

CA-IDMS/DB Reorg's reorganization capabilities are controlled by:

- Parameters
- Subschema definition
- Internal logic

The following subsections explain various CA-IDMS/DB Reorg capabilities in detail.

3.5.1 Reorganization Capabilities Controlled by Parameters

Most of CA-IDMS/DB Reorg's parameters allow you to control internal physical reorganization functions for sets and records and to specify where the reorganization is to be performed. By using these parameters you can direct CA-IDMS/DB Reorg to:

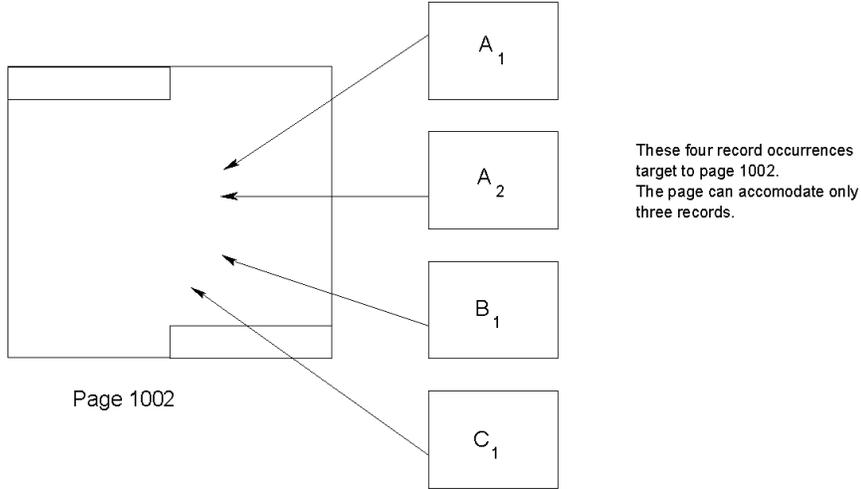
Resolve record/page conflicts. The PRIORITY parameter allows you to specify which record types (or group of record types) are to be stored on the target page and which ones should overflow if the page is full. The priorities you assign take effect only when records overflow. The priorities you assigned to record types are used to determine which records receive priority storage locations and which ones should overflow if a page isn't large enough to store all records targeting to it. (See Figure 3-5 on page 3-15)

Define clustering. The PRIORITY parameter also allows you to define how VIA clusters are stored. Figure 3-5 on page 3-15 shows how you can direct CA-IDMS/DB Reorg to cluster VIA records in the order in which they will be retrieved.

Specify page reserve. CA-IDMS/DB Reorg does not force you to create a special DMCL just to specify a page reserve for reorganization. CA-IDMS/DB Reorg uses a parameter-defined value instead.

Rebuild integrated indexes. CA-IDMS/DB Reorg will rebuild all indices in an index area. This allows you to reduce or eliminate a number of intermediate-level index records that were created due to the order in which the member records were stored.

Reduce the number of I/Os required to search a sorted set. For CALC and sorted VIA record types, you can specify by the TARGETROOT parameter that CA-IDMS/DB Reorg is to store the root portion of the record, which contains the record's symbolic key, on the target page if the entire record will not fit. This reduces the number of I/Os required to search for a CALC or sorted VIA member. The remainder of the record is stored on as few pages as possible.



Given the priority statement for record types A, B, and C.

PRIORITY=(A,C,B)

CA-IDMS/DB REORG allocates the records this way:

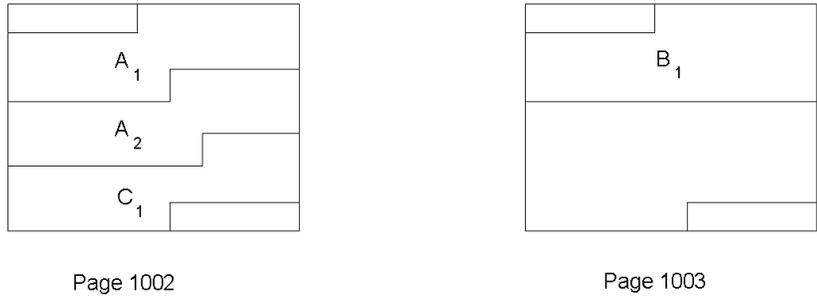


Figure 3-5. Resolving Record/Page Conflicts

3.5.2 Direct an Efficient Reorganization Through Parameters

In addition to the parameters that control internal physical reorganization functions for sets and records, some of CA-IDMS/DB Reorg's parameters provide operational efficiency features. By using these parameters, you can direct a more efficient reorganization--one that is compatible with your processing environment and its requirements. The parameters give you the capabilities to:

Interrupt the reorganization process. During a reorganization you may find it necessary to recover from a system failure, perform backup procedures, perform operations functions, or conserve and manage disk space. With the STOPAFTER parameter you can interrupt CA-IDMS/DB Reorg after any individual step. Later, you can direct CA-IDMS/DB Reorg to resume execution where it left off and complete the reorganization by using the RESTART parameter.

Monitor the process through console communication. The DSPLYINT parameter lets you establish a console communication facility through which you can display CA-IDMS/DB Reorg's status at specified time intervals, at the end of each step, or on demand. In addition to establishing console communication, you can also monitor the reorganization from TSO terminals by using the TSOUSERS parameter.

Once the communication facility is established through the DSPLYINT parameter, you can issue dynamic commands from the operator console to view the status, to stop or cancel the reorganization process, and to change the interval at which the status is displayed.

Sweep or not sweep non-copy areas. When the area you are reorganizing has set linkage with records in another area, CA-IDMS/DB Reorg must access those records to update their pointers. CA-IDMS/DB Reorg will sweep the non-copy area and create extracts for any records with set linkage unless directed otherwise.

When the non-copy area contains few record occurrences with set linkage to records in a copy area, random access of these records (set walking) is the most efficient way to create the necessary pointer extracts. The NOSWEEP parameter directs CA-IDMS/DB Reorg to walk only the sets which are necessary to create pointer extracts.

If CA-IDMS/DB Reorg can generate pointer extracts for a record in a non-copy area by using the NEXT and PRIOR pointers in the copy area, the NOSWEEP parameter will be turned on automatically, and no sets will be walked.

To determine whether area sweeping or set walking is more efficient it is necessary to know the number of record (or active set) occurrences in an area. You can use the CA-IDMS/DB Analyzer Set Report to give you this information. If you are not sure which mode is faster, use the area sweep default.

Chapter 2, "Parameters and Commands" on page 2-1 includes details for each parameter and command.

3.5.3 Reorganization Capabilities Controlled by Subschema and DMCL Definition

Many reorganization capabilities depend on how you have defined the new subschema and DMCL. During reorganization, CA-IDMS/DB Reorg compares the area, set, index, and record definitions in the new subschema with those in the old subschema. After comparing the old subschema with the new subschema, CA-IDMS/DB Reorg has the capabilities to:

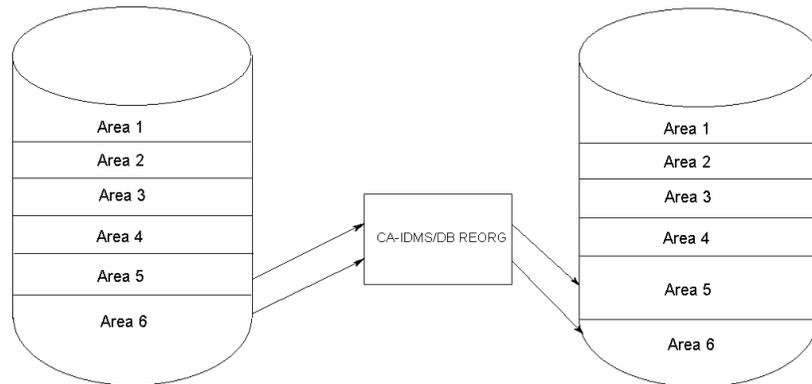
Expand/reduce the page range of one or more areas of the database. This special feature makes the database adaptable to changes in area size triggered by database growth. When an area needs to be enlarged, you can easily add pages to an area by defining a larger area in the definition of the DMCL (see Figure 3-6 on page 3-18).

Expand/reduce an area's page size. CA-IDMS/DB Reorg bases the SMP interval on the page size in the new DMCL. This feature eliminates the need for the area's SMP page interval to reflect the SMP interval with which the area was initialized.

Change record's page range. CA-IDMS/DB Reorg automatically changes a record's page range when you reorganize an area.

Update cross-area set linkage. You can reorganize a single area—even if it has set linkage with other areas in the database. CA-IDMS/DB Reorg determines from the subschema which sets point outside of the area being reorganized and updates the set linkage in these other areas. With CA-IDMS/DB Reorg, you do not have to reorganize areas that have set linkage with the area being reorganized. (When using utilities where reorganization of linked areas is required, often a “snowball effect” occurs and the entire database ends up being reorganized.)

Integrate integrated indexes. CA-IDMS/DB Reorg updates the index table entries for all indexes whose member records are reorganized. Just as reorganization of areas containing members or owners of sets whose set linkage crosses area boundaries does not require reorganization of each linked area, CA-IDMS/DB Reorg does not require reorganization of the index area just because you reorganized the member record's area. CA-IDMS/DB Reorg allows you to copy only two areas of the database and it optionally changes the page ranges of both areas.



CA-IDMS/DB REORG allows you to copy only two areas of the database and it optionally changes the page ranges of both

Figure 3-6. Changing the Page Range of Two Areas

Eliminate logically deleted records. When CA-IDMS logically deletes records, CA-IDMS removes data from a record, but the prefix remains intact. CA-IDMS/DB Reorg automatically deletes logically deleted records (LDELs) in areas that are being copied. This means that you do not have to use another utility to eliminate logically deleted records prior to using CA-IDMS/DB Reorg.

Expand/reduce the number of index table entries. CA-IDMS/DB Reorg allows the SR8 record type to change its record size (a change in record size is otherwise considered to be a restructure feature and is not allowed). Changing the size of an SR8 record by changing the INDEX BLOCK CONTAINS value in the subschema gives you control over the number of levels required to build an index. This change to the subschema does not require recompilation of any programs.

Move a record type to another area. For performance, backup, or security purposes, you can direct CA-IDMS/DB Reorg to move a record type to a different area. This is done by changing the record's WITHIN clause in the new subschema's schema and reorganizing both the record's old area and new area.

Change a record's location mode. By analyzing statistics from the CA-IDMS/Journal Analyzer and CA-IDMS/DB Analyzer, you may determine that VIA records would be in better locations if they were clustered around a different owner. CA-IDMS/DB Reorg also allows you to change a record's current location mode to or from DIRECT,

CALC, or VIA. However, because CA-IDMS/DB Reorg does not compress or decompress records, the control length of variable or fixed compressed records must not change.

Optimize VIA records. CA-IDMS/DB Reorg reclusters VIA set occurrences so that VIA members are stored physically in set order. This reduces the number of I/Os required to walk the set (see Figure 3-3 on page 3-11).

3.5.4 Automatic Reorganization Controlled by Internal Logic

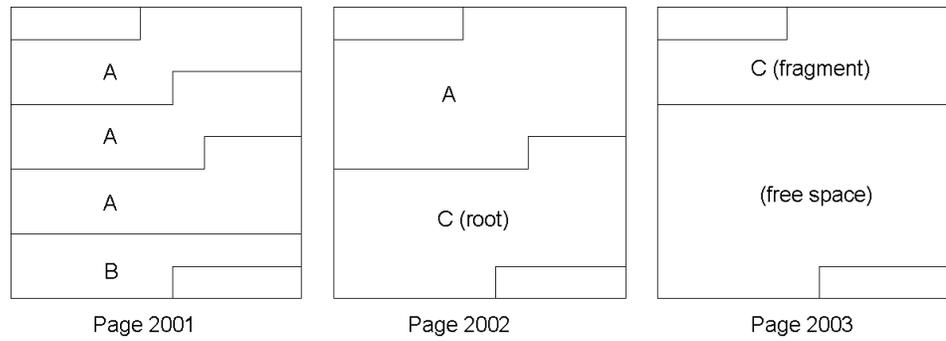
Many of CA-IDMS/DB Reorg's capabilities are controlled automatically by its own internal processing logic. For these functions you do not have to enter a parameter or define subschemas. Through its own internal processing logic CA-IDMS/DB Reorg has the capabilities to:

Minimize fragmented records. CA-IDMS/DB Reorg does not fragment records unless there is not a page available with enough free space to accommodate the entire record (see Figure 3-7 on page 3-20).

Retarget relocated records to their home page. CA-IDMS/DB Reorg automatically retargets relocated records in the area(s) being reorganized.

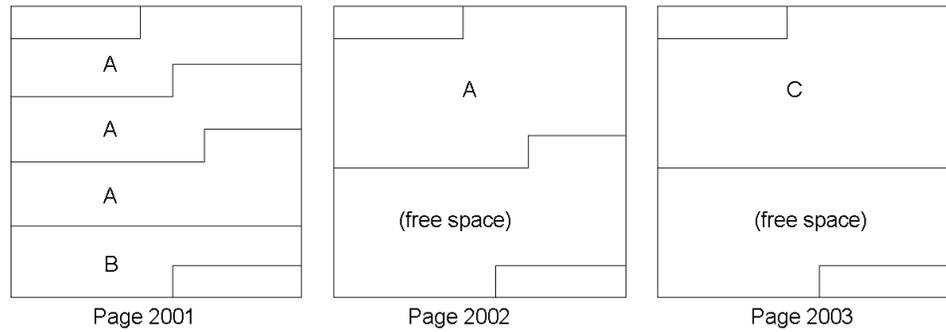
Optimize CALC and sorted VIA sets. CA-IDMS/DB Reorg allocates CALC and sorted VIA record types in symbolic key sequence. This minimizes the number of page changes required to walk the set; therefore, the number of I/Os performed when searching for a record by its symbolic key can be reduced (see Figure 3-2 on page 3-9 and Figure 3-3 on page 3-11).

Rebuild the index structure. When you COPY an area containing an integrated index structure, the index structure is rebuilt automatically. As a result, negative effects of index splits and spawns are eliminated and you can process more efficiently. In addition, the index pointer in the member record is maintained regardless of the record in a non-copy area. Furthermore, the inherent inefficiency of index orphans are eliminated as well.



BEFORE CA-IDMS/DB REORG

When storing VIA member record C, CA-IDMS will attempt to place record C on page 2001 with its owner record B. However, because page 2001 is already full, CA-IDMS will fragment record C across the successive pages until the entire record is stored. Two I/Os are required to access record C.



AFTER CA-IDMS/DB REORG

When allocating VIA member record C, CA-IDMS/DB REORG determines that there is not enough room on the owner page and will not attempt to place the member record on that page. CA-IDMS/DB REORG will search for a page where the entire record will fit. Only one I/O is required to access record C.

Figure 3-7. Minimizing Fragmented Records

3.6 Reorganization Steps

CA-IDMS/DB Reorg performs reorganization incrementally by executing a series of steps (see Figure 3-8 on page 3-22). This incremental approach to reorganization makes possible several operational benefits.

Step-by-step reorganization permits you to stop and restart the utility after completion of any step. By using the restart feature (see 1.6.1, “STOPAFTER and RESTART Parameters” on page 1-11), you can interrupt processing and restart it later.

The interruption feature gives you the ability to use the reorganization steps as milestones when monitoring the reorganization process from the operator console or TSO terminal and to predict how long reorganization will take. See Chapter 4, “System Output” on page 4-1 for complete information on the status display available at the operator console.

CA-IDMS/DB Reorg must execute each reorganization step sequentially (see Figure 3-8 on page 3-22). While you cannot eliminate a reorganization step, an understanding of the function of each step can help you to:

- Allocate work files and estimate their sizes efficiently
- Stop and restart the utility
- Back up files
- Overlay the old database with the new database to conserve disk space
- Perform other operational functions, such as the initial program load (IPL) of the system.

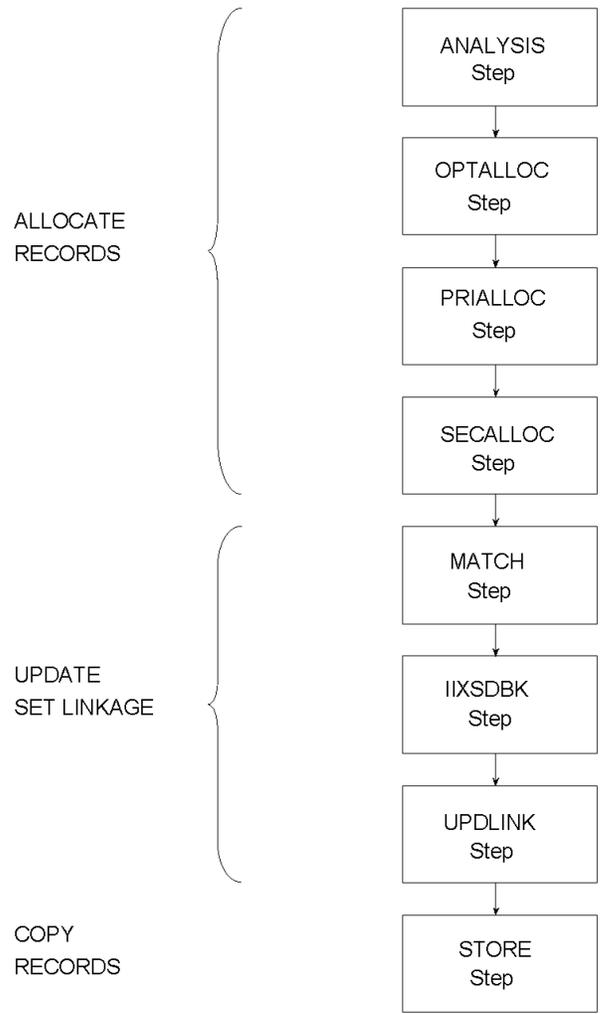


Figure 3-8. Reorganization Steps and Their Logical Sequence

3.6.1 Functions of the Reorganization Steps

The table below shows how the reorganization steps can be grouped together by function. The table gives you a clear picture of which steps perform the allocation and update functions and which steps access the old database and the new database, or both.

ANALYSIS (Analysis) Step--CA-IDMS/DB Reorg compares the old subschema and DMCL to the new subschema and DMCL (all are named in the PROCESS statement) and determines how to process records that have set linkage with the COPY areas.

OPTALLOC (Optimum Allocation/Pointer Extract) Step--Determines the optimum target page allocation for every record in the areas to be reorganized. Processing in OPTALLOC allocates records to pages as if a page could hold every record that targeted to that page. CA-IDMS/DB Reorg then creates pointer extracts for set linkages with records in the areas selected by using the COPY statement.

If a record is indexed and the index is maintained in db-key sequence, the extract is flagged for later special processing by the MATCH step and the IIXSDBK step.

PRIALLOC (Primary Allocation) Step--Allocates a record's real db-key (based on the user-specified priority in the PRIORITY parameter) for records stored DIRECT and CALC. For VIA records, CA-IDMS/DB Reorg allocates the db-key for records that target to the owner page. DIRECT, CALC and VIA records that will not fit on the target page are written to the UNALLOC (unallocated) file.

SECALLOC (Secondary Allocation) Step--Allocates real db-keys for VIA records that do not target to the owner page, and for records that were written to the UNALLOC file. For variable length records that cannot fit as a whole on any page in that record's page range, the SECALLOC Step assigns db-keys to record fragments.

If an integrated index (SR8) resides in an area being copied, the program will:

- Reconstruct the integrated index
- Allocate real db-keys for the new SR8 records
- Create the appropriate pointer extracts for set linkage with records in the copy area.

MATCH (Match) Step--Matches the db-keys from the previous two allocation steps (PRIALLOC and SECALLOC) with pointer extracts created in the OPTALLOC Step. The db-keys in the pointer extracts are then updated with the new db-keys from the allocation steps.

This step also creates a separate file for integrated index sets sorted by db-key. The file contains two records for each lower level SR8 record and one record for each upper level SR8 record.

The first lower level records identifies the new db-key. The second lower level record contains a sequence number that identifies the entry's relative position among the duplicates that have the same db-key key. If there are any upper level records that point to lower level duplicates, a record is created that contains the same sequence number as the lower level record to which it points.

Table 3-1 (Page 1 of 2). Reorganization Steps and Their Functions

Step/Function	Allocates Records to Area Specified in COPY Statement	Updates Set Linkage	Accesses Old Database	Access New Database
OPTALLOC Step	Computes optimum target page.	Extracts pointers which will change. If copying area with IIX set, writes individual bottom level entries to SECOPTA.	Reads records being copied and those with set linkage.	
PRIALLOC Step	Allocates DIRECT, CALC, and VIAs stored on owner's page.			
SECALLOC Step	Allocates CALC, DIRECT, VIA overflows, fragments, VIAs not stored on owner's page and SR8 records. Rebuilds IIX tables to create new SR8 structures.	Extracts pointers for all SR8 set linkage.		

Table 3-1 (Page 2 of 2). Reorganization Steps and Their Functions

Step/Function	Allocates Records to Area Specified in COPY Statement	Updates Set Linkage	Accesses Old Database	Access New Database
MATCH Step	Replaces old db-key in pointer extracts with new db-keys from allocations.	Replaces old db-key in pointer extracts with new db-keys from allocations.		
IIXSDBK Step	Reorders IIX set sorted by db-key into proper sequence of db-key.	Reorders IIX set sorted by db-key into proper sequence of db-key.		
UPDLINK Step		Updates pointer prefix or IXDET table entry with new db-key.	Modifies records with set linkage in non-copy areas. Extracts all records residing in copy areas. Creates the DBREC file and . optionally creates the DBKEYS file.	
STORE Step				Populates areas being copied.

IIXSDBK--Reorders duplicate SR8 entries that may have been put out of their proper sequence (indexed record db-key) when CA-IDMS/DB Reorg assigned new db-keys to the indexed records (during the PRIALLOC or SECALLOC steps).

The IIXSDBK step is executed only if you have COPYed an area containing an integrated index set sorted by db-key. Otherwise CA-IDMS/DB Reorg automatically bypasses the step.

UPDLINK (Update Set Linkage) Step--Updates the record's set linkage, using the new db-keys placed in pointer extracts by the MATCH step. Records in a copy area are then written to the DBREC file. Records that are not in a copy area, but whose pointers are being updated, are written back to the old database. If specified, an interface file for DB-EZ Reorg's inflight utility program is created.

STORE (Store) Step--Records that reside in a copy area are stored in the new database.

Chapter 4. System Output

4.1 Overview	4-3
4.2 CA-IDMS/DB Reorg Output	4-4
4.2.1 Audit Report	4-4
4.2.2 Status Display	4-4
4.2.3 Interface File	4-4
4.3 Audit Report	4-5
4.3.1 Audit Subreport--Area Recap for Old Subschema	4-6
4.3.2 Audit Subreport--Set Linkage Recap for Area	4-8
4.3.3 Restart Information Messages	4-10
4.4 Status Display	4-12
4.4.1 Status Display Fields	4-12
4.5 DBKEYS Interface File	4-13

4.1 Overview

This chapter is a guide to the output of CA-IDMS/DB Reorg: the Audit Report, the Status Display, and the interface file. Pages from a sample Audit Report and a sample Status Display are provided with detailed explanations of the content from both types of output.

4.2 CA-IDMS/DB Reorg Output

CA-IDMS/DB Reorg produces three types of output: a printed Audit Report, a Status Display which can be viewed at the operator console and specified TSO terminals, and an interface file for use with DB-EZ Reorg or the CA-IDMS/DB Utilities. per.

4.2.1 Audit Report

The Audit Report is produced automatically every time you run CA-IDMS/DB Reorg. This report traces all CA-IDMS/DB Reorg steps and activities by providing three types of messages:

- Parameter Messages
- Processing Messages
- Restart Information Messages.

When CA-IDMS/DB Reorg performs the Analysis step, the Audit Report also contains two subreports that list how records are accessed in the old database:

- Area Recap for Old Subschema
- Set Linkage Recap for Area.

A sample Audit Report that includes the two subreports and the three types of messages generated by CA-IDMS/DB Reorg is illustrated and explained on the following pages.

4.2.2 Status Display

The Status Display is produced by CA-IDMS/DB Reorg at the operator console at specified TSO terminals, or at selected functional areas designated by route code parameters.

From this display you can monitor the reorganization and estimate CA-IDMS/DB Reorg run time. You can view information about the current reorganization step, such as step name, area name, page ranges, and number of records processed.

The content of the Status Display is thoroughly explained at the end of this chapter and illustrated in Sample Status Display - OPTALLOC Step on page 4-12 and Sample Status Display - PRIALLOC Step on page 4-12.

4.2.3 Interface File

CA-IDMS/DB Reorg permits to interface with DB-EZ Reorg via the DBKEYS file. DB-EZ Reorg utilizes this file for its inflight utility program. This file is described in detail in 4.5, “DBKEYS Interface File” on page 4-13

4.3 Audit Report

The Audit Report is produced automatically every time you run CA-IDMS/DB Reorg. The information the report contains depends on the parameters you specified and on occurrences during processing. Figure 4-2 on page 4-6 shows two pages from an Audit Report with the following information:

Report Title--The title of the report.

UR--The tape from which CA-IDMS/DB Reorg was installed.

RELEASE--The version number of CA-IDMS/DB Reorg that has been installed.

Parameter Messages list all input parameters and convey information about parameter processing. The messages appear in a two-column format. The left column lists the message ID and the text of the message. The right column displays the parameter as you entered it.

If the parameter was entered incorrectly, CA-IDMS/DB Reorg underlines the parameter in error with asterisks (****). The portion of the parameter that was not checked is underscored with a series of Xs (XXXX).

Processing Messages list the times at which reorganization steps begin and end, step statistics, errors detected in the database, and other internal auditing information. Processing messages can be found on most pages of the Audit Report.

Figure 4-2 on page 4-6 shows processing messages generated by the routine that checks parameter input and the OPTALLOC step.

All messages on the Audit Report have an eight-character message ID. The message ID begins with a seven-character identifier and ends with a severity code of I, W, or E, signifying an informative message, a warning message, or an error message. See Chapter 6, "Messages" on page 6-1 for an explanation of the three types of severity codes and information about specific messages.

4.3 Audit Report

ID	RELEASE Rnn.nn	CA-IDMS/DB REORG AUDIT REPORT	DATE mm/dd/yy	TIME hh:mm:ss	PAGE nnnn
AUDT001I	STEP: PARM, STARTED		DATE: 01/31/85	TIME: 13:33:43	
			V---+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-V		
NKWP091I	INPUT PARAMETER STATEMENT		PROCESS, OLD=DBR18ALL,NEW=DBR18ALL,ST=UPDLINK		
PARM043I	NO PARAMETER ERRORS DETECTED				
NKWP091I	INPUT PARAMETER STATEMENT		COPY, AREA=LOC-REGION		
PARM043I	NO PARAMETER ERRORS DETECTED				
NKWP091I	INPUT PARAMETER STATEMENT		COPY, AREA=DEPT-REGION		
PARM043I	NO PARAMETER ERRORS DETECTED				
NKWP091I	INPUT PARAMETER STATEMENT		COPY, AREA=STUDENT-REGION		
PARM043I	NO PARAMETER ERRORS DETECTED				
NKWP091I	INPUT PARAMETER STATEMENT		COPY, AREA=CLASS-REGION		
PARM043I	NO PARAMETER ERRORS DETECTED				
NKWP091I	INPUT PARAMETER STATEMENT		COPY, AREA=MISC2-REGION		
NKWP094I	END OF PARAMETER INPUT				
AUDT002I	STEP: PARM, ENDED, CC= 0000		DATE: mm/dd/yy	TIME: hh:mm:ss	

Figure 4-1. Audit Report - Processing Parameters Step

ID	RELEASE Rnn.nn	CA-IDMS/DB REORG AUDIT REPORT	DATE mm/dd/yy	TIME hh:mm:ss	PAGE nnnn
AUDT001I	STEP: OPTALLOC, STARTED		DATE: mm/dd/yy	TIME: hh:mm:ss	
STAT001I	IDMS STATISTICS FOLLOW.				
	CALLS TO IDMS.....1,160				
	DBMS PAGE REQUESTS.....262				
	PAGES READ.....192				
	PAGES WRITTEN.....6				
STAT002I	OPTALLOC STATISTICS FOLLOW.				
OPTX001I	OBTAINS FOR AREA SWEEP.....245				
OPTX002I	OBTAINS FOR SET WALK.....142				
OPTX007I	EXTRACT RECORDS WRITTEN.....817				
OPTX008I	EXTRACT RECORDS WRITTEN.....39				
OPTX009I	SECOPTA RECORDS WRITTEN.....34				
OPTX013I	NUMBER OF IIX SORTED DBKEYS AT BOTTOM LEVEL0				
OPTX014I	NUMBER OF IIXSRKY RECORDS THAT WILL BE CREATED.....0				
OPTX015I	NUMBER OF IIXSDBK POINTING TO UPPER LEVELS0				
AUDT002I	STEP: OPTALLOC, ENDED, CC= 0000		DATE: mm/dd/yy	TIME: hh:mm:ss	

Figure 4-2. Audit Report - OPTALLOC Step

4.3.1 Audit Subreport--Area Recap for Old Subschema

The Area Recap for Old Subschema is produced automatically whenever the ANALYSIS step of CA-IDMS/DB Reorg is executed. This subreport lists the names of all areas in the old database, the number of pages in each area, and the method CA-IDMS/DB Reorg uses to access the records in each area.

Figure 4-3 on page 4-7 shows an Area Recap for Old Subschema with the following information:

Subreport Heading--The title of the subreport.

AREA--A list of the names of all the areas in the old subschema.

NBR-PAGES--A list of the number of pages in each area of the old database. These numbers can be used to estimate execution time, and can calculate the size (number of record) of the PAGUTIL file.

PROCESSING-TYPE--The type of processing CA-IDMS/DB Reorg will perform, according to your parameter specifications, against each area in the old subschema. For each area listed one of five processing types may appear:

- **SWEEP**--Indicates that an area sweep will be performed.
- **NOT ACCESSED**--Indicates that CA-IDMS/DB Reorg will not access the area.
- **NOSWEEP - SET WALKING**--Indicates that CA-IDMS/DB Reorg will walk sets in the area.
- **NOSWEEP - NEXT AND PRIOR**--Indicates that CA-IDMS/DB Reorg will not sweep (or walk sets) to create pointer extracts for records that have cross-area linkage with a copy area. Instead, pointer extracts are created by using next and prior pointers from the records in the copy area.
- **COPY**--Indicates that the area will be copied to the new database.

ID	RELEASE Rnn.nn	CA-IDMS/DB REORG AUDIT REPORT	DATE mm/dd/yy	TIME hh:mm:ss	PAGE nnnn
AUDT001I STEP: ANALYSIS, STARTED			DATE: mm/dd/yy	TIME: hh:mm:ss	
AREA RECAP FOR OLD SUBSCHEMA					
		-----AREA-----	NBR-PAGES	PROCESSING-TYPE	
		BOGUS-1-REGION	30	NOT ACCESSED	
		BOGUS-2-REGION	30	NOT ACCESSED	
		BOGUS-3-REGION	30	NOT ACCESSED	
		CLASS-REGION	30	COPY	
		DEPT-REGION	15	COPY	
		LOC-REGION	5	COPY	
		MISC1-REGION	20	SWEEP	
		MISC2-REGION	20	COPY	
		STUDENT-REGION	80	COPY	

Figure 4-3. Audit Subreport--Area Recap for Old Subschema

4.3.2 Audit Subreport--Set Linkage Recap for Area

The Set Linkage Recap for Area is produced for each area in the old subschema during the Analysis Step. This subreport lists whether or not records have set linkage to records in the area being reorganized. This subreport also lists the method CA-IDMS/DB Reorg uses, such as AREA SWEEP or SET WALKING, to access the old database to create pointer extracts, and displays the number of extracts created for each record. These numbers can be used to calculate sizes of the intermediate files. See Chapter 5, "Operations" on page 5-1 for instructions to calculate file sizes.

Figure 4-4 on page 4-9 illustrates a Set Linkage Recap for Area with the following information:

Subreport Heading--The title of the subreport.

area-name--The name of the area for which set linkage is reported.

record-type--The name of a record in the area. Each record is separated by a line of dashes.

nn EXTRACTS/RECORD (AVG)--The number of extracts created for each occurrence of the record type. The number of extracts is needed to compute the size of the intermediate work file. If the text **NO SET LINKAGE/NO EXTRACTS** appears in this field, the record does not have set linkage with any records in the COPY area and there will be no information in the rest of the fields for that record.

location mode--The location mode of the record in the old database and in the new database.

page range--The page range for the record in the old database and in the new database.

OWN/MEM--Indicates whether the record is an owner or a member of the set in the SET NAME field.

SET-NAME--The name of the set.

SET-POINTERS--Indicates types of pointers in the set: NEXT, PRIOR, or OWNER.

POINTER-EXTRACTS-CREATED-BY--Indicates one of three methods CA-IDMS/DB Reorg will use to create extracts: AREA SWEEP, SET WALKING, or USING NEXT AND PRIOR.

NEW VIA-OWNER NAME--The name of the VIA owner record type, if the record is stored VIA. If this field is blank, the record type is not stored VIA the set to which it belongs (set name shown in the SET NAME field).

LOC-MODE--The location mode of the record type listed under NEW VIA-OWNER.

HI-VIA-OWNER--CA-IDMS/DB Reorg walks VIA sets to determine how to allocate them in logical set order. Because a VIA set owner can be stored VIA another set, CA-IDMS/DB Reorg will not walk the set until it accesses the high VIA owner. The high VIA owner either is stored in a noncopy area or has a location mode which is not VIA.

ID	RELEASE Rnn.nn	CA-IDMS/DB REORG AUDIT REPORT	DATE mm/dd/yy	TIME hh:mm:ss	PAGE nnnn
SET LINKAGE RECAP FOR AREA CLASS-REGION					

08106	CLASS.....	RECORD	14 EXTRACTS / RECORD (AVG)	OLD LOCATION MODE OF VIA NEW LOCATION MODE OF VIA	PAGE RANGE OF 89,001 89,030 PAGE RANGE OF 89,001 89,030
OWN/MEM	---SET-NAME----	--SET-POINTERS--	POINTER-EXTRACTS-CREATED-BY	NEW VIA-OWNER NAME	LOC-MODE
OWNER	CLASS-SCHEDULE	NEXT,PRIOR	AREA SWEEP		
MEMBER	TEACHER-CLASS	NEXT,PRIOR,OWNER	AREA SWEEP		
MEMBER	PERIOD-CLASS	NEXT,PRIOR,OWNER	AREA SWEEP		
MEMBER	ROOM-CLASS	NEXT,PRIOR,OWNER	AREA SWEEP		
MEMBER	SUBJECT-CLASS	NEXT,PRIOR,OWNER	AREA SWEEP	SUBJECT.....	CALC HI-VIA-OWNER

08107	SCHEDULE...	RECORD	7 EXTRACTS / RECORD (AVG)	OLD LOCATION MODE OF VIA NEW LOCATION MODE OF VIA	PAGE RANGE OF 89,001 89,030 PAGE RANGE OF 89,001 89,030
OWN/MEM	---SET-NAME----	--SET-POINTERS--	POINTER-EXTRACTS-CREATED-BY	NEW VIA-OWNER NAME	LOC-MODE
OWNER	SCHEDULE-BOGUS	NEXT,PRIOR	AREA SWEEP		
MEMBER	CLASS-SCHEDULE	NEXT,PRIOR,OWNER	AREA SWEEP	CLASS.....	VIA
MEMBER	STUDENT-SCHEDULE	NEXT,PRIOR	AREA SWEEP		

08115	CON-TROL....	RECORD	NO SET LINKAGE / NO EXTRACTS		

08116	BOGUS-1.....	RECORD	5 EXTRACTS / RECORD (AVG)	OLD LOCATION MODE OF VIA NEW LOCATION MODE OF VIA	PAGE RANGE OF 89,001 89,030 PAGE RANGE OF 89,001 89,030
OWN/MEM	----SET-NAME----	--SET-POINTERS--	POINTER-EXTRACTS-CREATED-BY	NEW VIA-OWNER NAME	LOC-MODE
OWNER	BOGUS-ARTS	NEXT,PRIOR	AREA SWEEP		
MEMBER	SCHEDULE-BOGUS	NEXT,PRIOR	AREA SWEEP	SCHEDULE.....	VIA
MEMBER	ACTIVITY-BOGUS	NEXT	AREA SWEEP		
NLYZ006I MAXIMUM RECORD SIZE FOR NEW SUBSCHEMA.....492 NLYZ007I MAXIMUM CALC KEY SIZE FOR OLD SUBSCHEMA.....20 NLYZ008I MAXIMUM IX-DETAIL SIZE FOR OLD SUBSCHEMA.....108 NLYZ009I MAXIMUM IX-DETAIL SIZE FOR NEW SUBSCHEMA.....108 NLYZ012I MAXIMUM SYMBOLIC KEY SIZE FOR INDEXED SET.....0 NLYZ013I MAXIMUM RECORD SIZE FOR AN SR8 RECORD.....0 NLYZ010I new/old SUBSCHEMA subschema-name WAS COMPILED compile-date compile-time DMCL dmc1-name, DBNAME dbname					
NLYZ005I END OF ANALYSIS REPORTS					
AUDT002I STEP: ANALYSIS, ENDED, CC=0000 DATE: mm/dd/yy TIME: hh:mm:ss					

Figure 4-4. Audit Subreport--Set Linkage Recap for Area

4.3.3 Restart Information Messages

Restart Information Messages supply information that is helpful in restarting CA-IDMS/DB Reorg. The messages appear automatically on the Audit Report when CA-IDMS/DB Reorg is interrupted by the STOPAFTER and RESTART parameters or the STOP or CANCEL Console Communication Commands, and one or more steps are needed to complete the reorganization. You can also request Restart Information Messages in a report-only run by using the RPT parameter in the PROCESS statement.

If CA-IDMS/DB Reorg processing terminates abnormally because of a system failure or system error, you will not receive Restart Information Messages on the Audit Report. Figure 4-5 on page 4-11 shows a page from the sample Audit Report which contains Restart Information Messages.

Message ID--Restart Information Message ID. Messages are described in Chapter 6, "Messages" on page 6-1.

RESTART DATA...

- Names of the old and new subschemas.
- The name, date, and execution times of the routine that checks parameter input.
- All reorganization step names and the dates and times the steps executed. If the date and time do not appear, the reorganization step has not executed.

RESTART WITH.....

- Name of the reorganization step with which CA-IDMS/DB Reorg resumes execution when the NEXT option is selected for the RESTART parameter.

DATASET DISPOSITION FOR RESTART.....

This message contains work file information which is helpful in restarting CA-IDMS/DB Reorg:

- Filenames (OS/390 DDNAMEs).
- File status--Disposition of the files as though each step were a separate OS/390 step.
 - **OLD/NEW**--Indicates the current status of each dataset. NEW indicates that the file will be created during the step named in the report. OLD indicates that the file existed before the reorganization step executed.
 - **KEEP**--The file is saved because the reorganization step executed normally.
 - **DELETE**--The file is not saved because the step is not executed correctly and results in a system error.
- The type of processing for each file and the step in which it is performed. This information can be used to determine when disk space must be allocated and when it can be freed.
 - **CREATED BY**--The name of the step in which the file is created.

- **READ BY**--The name of the step in which the file is read.
- **UPDATED BY**--The name of the step in which the file is modified.

ID	RELEASE Rnn.nn	CA-IDMS/DB REORG AUDIT REPORT	DATE mm/dd/yy	TIME hh:mm:ss	PAGE nnnn
RESTART STATUS INFORMATION					
PRST001I	RESTART DATA.....	OLD SUBSCHEMA DBR18ALL NEW SUBSCHEMA DBR18ALL			
hh:mm:ss		PARM.....	STARTED mm/dd/yy AT hh:mm:ss	ENDED mm/dd/yy AT	
		ANALYSIS.....	STARTED mm/dd/yy AT hh:mm:ss	ENDED mm/dd/yy AT hh:mm:ss	
		OPTALLOC.....	STARTED mm/dd/yy AT hh:mm:ss	ENDED mm/dd/yy AT hh:mm:ss	
		PRIALLOC.....	STARTED mm/dd/yy AT hh:mm:ss	ENDED mm/dd/yy AT hh:mm:ss	
		SECALLOC.....	STARTED mm/dd/yy AT hh:mm:ss	ENDED mm/dd/yy AT hh:mm:ss	
		MATCH.....	STARTED mm/dd/yy AT hh:mm:ss	ENDED mm/dd/yy AT hh:mm:ss	
		IIXSDBK.....	STARTED mm/dd/yy AT hh:mm:ss	ENDED mm/dd/yy AT hh:mm:ss	
		UPDLINK.....	STARTED mm/dd/yy AT hh:mm:ss	ENDED mm/dd/yy AT hh:mm:ss	
		STORE.....			
PRST002I	RESTART WITH	STORE			
PRST003I	DATASET DISPOSITION FOR RESTART.....	CNTRL1	OLD KEEP	CREATED BY ANALYSIS STEP UPDATED BY PARM STEP UPDATED BY ANALYSIS STEP	
		CNTRL2	OLD KEEP	CREATED BY ANALYSIS STEP UPDATED BY PARM STEP	
		EXTRACT	OLD KEEP	CREATED BY OPTALLOC STEP READ BY MATCH STEP	
		PRIOPTA	OLD KEEP	CREATED BY OPTALLOC STEP READ BY PRIALLOC STEP	
		SECOPTA	OLD KEEP	CREATED BY OPTALLOC STEP READ BY SECALLOC STEP	
		PAGUTIL	OLD KEEP	CREATED BY PRIALLOC STEP READ BY SECALLOC STEP	
		UNALLOC	OLD KEEP	UPDATED BY SECALLOC STEP CREATED BY PRIALLOC STEP	
		PRIREAL	OLD KEEP	READ BY SECALLOC STEP CREATED BY PRIALLOC STEP	
		SECREAL	OLD KEEP	READ BY MATCH STEP CREATED BY SECALLOC STEP	
		ALLOCX	OLD KEEP	READ BY MATCH STEP CREATED BY MATCH STEP	
		IIXSRKY	OLD KEEP	READ BY UPDLINK STEP CREATED BY IIXSDBK STEP	
		IIXEXOL	OLD KEEP	READ BY IIXSDBK STEP CREATED BY MATCH STEP	
		IIXEXTR	OLD KEEP	READ BY IIXSDBK STEP CREATED BY IIXSDBK STEP	
		DBREC	OLD KEEP	READ BY UPDLINK STEP CREATED BY UPDLINK STEP	
				READ BY STORE STEP	

Figure 4-5. Restart Information Messages

4.4 Status Display

The optional Status Display lets you monitor CA-IDMS/DB Reorg's progress during the reorganization from the operator console or TSO terminals specified in the PROCESS statement. In an OS/390 environment with multiple console support, you can route the display to selected functional areas by using the route code parameters.

The Status Display shows which reorganization step is currently executing, gives processing status, and lists other data that will help you gauge CA-IDMS/DB Reorg run time. Sample Status Display - OPTALLOC Step on page 4-12 and Sample Status Display - PRIALLOC Step on page 4-12 provides sample Status Displays.

Status information is displayed at the completion of each step if you specify the STEP option (default value) of the DSPLYINT parameter. You can also request the Status Display shown at regular time intervals--from one minute to two hours.

To change the time intervals and turn off the display once CA-IDMS/DB Reorg is executing, you can enter dynamic commands from the operator console. Chapter 2, "Parameters and Commands" on page 2-1 provides a more complete explanation of the Status Display parameters and commands.

4.4.1 Status Display Fields

The Status Display is dynamic. The messages displayed depend on the status of the reorganization when a status update is requested from the operator console. Most messages are auditing messages that are self-explanatory and pertain to execution of the current reorganization step. All messages are described in detail in Chapter 6, "Messages" on page 6-1.

```
AUDT003I CURRENT STEP: OPTALLOC, IN SUBSTEP NON-ACTIVE, DUR 00.00.05
AUDT004I AREA: DEPT-REGION, PAGE RANGE+ 91,000 - 91,050
AUDT005I CURRENT PAGE 91,025 ( 50%)
```

Sample Status Display - OPTALLOC Step.

```
AUDT003I CURRENT STEP: PRIALLOC, IN SUBSTEP NON-ACTIVE, DUR 00.00.04
AUDT006I 2,000 RECORDS IN INPUT FILE
AUDT007I 1,000 RECORDS READ ( 50%)
```

Sample Status Display - PRIALLOC Step

4.5 DBKEYS Interface File

The DBKEYS file is created when the DDName DBKEYS is specified in the the JCL. The DBKEYS file can be used with CA-IDMS/DB Utilities or with DB-EZ Reorg. The file is described in Table 4-1.

Table 4-1. DBKEYS Interface File.

Field Name	Element Description	Explanation
OLDDBKEY	PIC S9(8) COMP SYNC	Old dbkey
NEWDBKEY	PIC S9(8) COMP SYNC	New dbkey
SR3DBKEY	PIC S9(8) COMP SYNC	SR3 dbkey. See note.
RECORDID	PIC S9(4) COMP	Record id
SR3RECID	PIC S9(4) COMP	Original record id. See note.

Note: True, only if the record was originally a SR2.

The file's logical record length (LRECL) is 16, the blocksize is specified via the BLOCKNUM parameter on the PROCESS statement.

Chapter 5. Operations

5.1 Overview	5-3
5.2 Database Access	5-4
5.3 Operational Flow and Considerations	5-5
5.3.1 Integrated Indexes	5-5
5.3.2 File Space Not Needed	5-5
5.3.3 Allocating Space for Work Files and Sort-Work Areas	5-6
5.3.4 Work Files	5-6
5.3.4.1 ANALYSIS Step	5-6
5.3.4.2 OPTALLOC Step	5-8
5.3.4.3 PRIALLOC Step	5-10
5.3.4.4 SECALLOC Step	5-12
5.3.4.5 MATCH Step	5-17
5.3.4.6 IIXSDBK Step	5-19
5.3.4.7 UPDLINK Step	5-20
5.3.4.8 STORE Step	5-22
5.3.5 Sort-Work Areas	5-22
5.3.6 Summary of Procedures for Operating CA-IDMS/DB Reorg	5-23
5.4 CA-IDMS/DB Reorg Processing--OS/390 Environment	5-26
5.4.1 Process 1--Defining the New Areas	5-26
5.4.2 Process 2--Gathering Information	5-27
5.4.3 Process 3--Allocating Work Files	5-28
5.4.4 Process 4--Reorganizing the Database	5-30
5.5 CA-IDMS/DB Reorg Processing--VSE/ESA Environment	5-34
5.5.1 VSE/ESA File Assignments	5-34
5.5.2 VSE/ESA File Processing Alternate Method	5-34
5.5.3 Process 1--Defining the New Areas	5-35
5.5.4 Process 2--Information Gathering	5-35
5.5.5 Process 3--Allocating Work Files	5-37
5.5.6 Process 4--Reorganizing the Database	5-39
5.6 CA-IDMS/DB Reorg Processing--VM/ESA Environment	5-43
5.6.1 Process 1--Defining the New Areas	5-43
5.6.2 Process 2--Information Gathering	5-43
5.6.3 Process 3--Allocating Work Files	5-47
5.6.4 Process 4--Reorganizing the Database	5-49

5.1 Overview

This chapter discusses CA-IDMS/DB Reorg operations. It provides information about accessing the database, important operational considerations, and file allocation techniques.

5.2 Database Access

CA-IDMS/DB Reorg utilizes CA-IDMS/DB I/O features such as, XA database buffers and control blocks; ESA database support; and dynamic database file allocation. In addition, the SYSIDMS PREFETCH parameter for read-ahead processing is available.

All CA-IDMS conventions regarding locking the database against concurrent access are used. While CA-IDMS/DB Reorg is executing you must ensure that no CV or local CA-IDMS update jobs are accessing the database. CA-IDMS/DB Reorg will not attempt to run against an area locked for update.

5.3 Operational Flow and Considerations

There are some important operational factors to consider before and during the execution of CA-IDMS/DB Reorg.

- Each step of CA-IDMS/DB Reorg must be executed in sequence. You can repeat steps, but you cannot skip a step.
- While the database is being reorganized, you must ensure that no other jobs are run against any area that is being reorganized.
- The new database for all areas specified in COPY statements must be formatted by running IDMSBCF prior to the STORE Step.
- In a VSE/ESA environment, you must assign the symbolic units for work files to a device type, even if you have a file manager, because of CA-IDMS/DB Reorg's own device-independent support.
- Execution time is directly related to the size of the area or areas being reorganized, the number of records in those areas, and the number of set linkages for each of those records. As the database reorganization executes, you can monitor the progress of the utility from the operator console.
- CA-IDMS/DB Reorg runs with an outstanding operator reply unless DSPLYINT=NO is specified. Be sure that your system console operator is aware of valid reply choices. See 2.3.12, “DSPLYINT Parameter” on page 2-14.
- CA-IDMS/DB Reorg automatically bypasses the IIXSDBK step, if it is not needed. IIXSDBK is executed only if you are updating an area containing integrated index sets sorted by db-key.

5.3.1 Integrated Indexes

- Integrated indexes must be defined in both the old and the new subschemas.
- If you are copying areas containing SR8 (integrated index) records, you must allocate space for SECALX2 and SECIIX2.
- If you are updating areas containing integrated index sets sorted by db-key, after the OPTALLOC step you must allocate space for three files (IIXEXOL, IIXEXTR, and IIXSRKY).
- The amount of database space needed for SR8 records defined in the new subschema can be determined from the calculations in Figure 5-5 on page 5-16

5.3.2 File Space Not Needed

- If you are not updating an area containing integrated index sets sorted by db-key, you do not need to define the files IIXEXOL, IIXEXTR, and IIXSRKY, in the JCL.
- If you are not copying an area containing SR8 (integrated index) records, you do not need to define the files SECALX2 and SECIIX2 in the JCL.

5.3.3 Allocating Space for Work Files and Sort-Work Areas

CA-IDMS/DB Reorg uses work files and sort-work areas to perform a reorganization. This work space can be assigned to tape or disk. If you decide to assign the space to disk, you must first estimate the size of your work files and sort-work areas. Figure 5-1 on page 5-7 to Table 5-9 on page 5-23 contain information to help you allocate space to the files.

5.3.4 Work Files

CA-IDMS/DB Reorg uses work files that vary in size. These files are sequential with the exception of the PAGUTIL file which must be RRDS VSAM or relative record BDAM. If VSAM, the PAGUTIL file must be NUMBERED.

Each work file requires a certain amount of space when you assign it to disk. Calculate the space for every file, except the EXTRACT file, by using the number of records (from the CA-IDMS/DB Analyzer reports or from your own sources) and the information given for estimating work file space in Figure 5-1 on page 5-7 to Table 5-9 on page 5-23. To determine the space allocation for the EXTRACT file, use the number of extracts per record listed on the CA-IDMS/DB Reorg Audit Report (4.3.2, “Audit Subreport--Set Linkage Recap for Area” on page 4-8) in addition to the number of records listed on the CA-IDMS/DB Analyzer Set Report.

When you use the information in Figure 5-1 on page 5-7 to Table 5-9 on page 5-23 to estimate the space required for the files, you may find that some files require more disk space than you have available; these files may be assigned to tape (except PAGUTIL).

File attributes such as RECORD LENGTH and RECORD FORMAT are defined internally and should not be coded in the JCL statements. The default BLOCKSIZE may be overridden, but it must conform to the information in Figure 5-1 on page 5-7 to Table 5-9 on page 5-23.

5.3.4.1 ANALYSIS Step

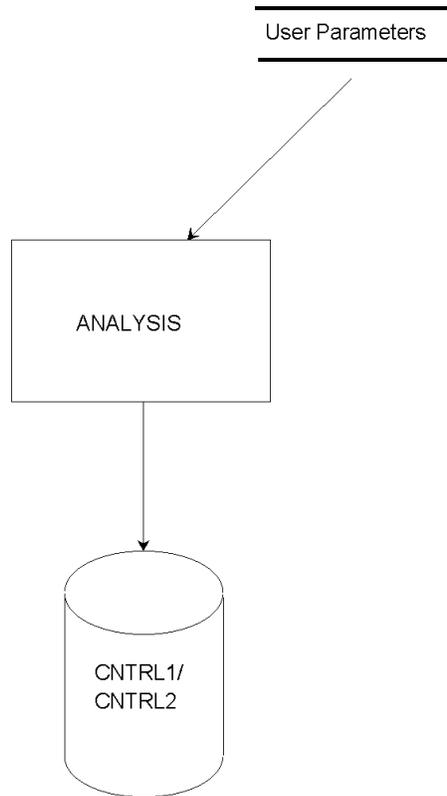


Figure 5-1. Work File Usage For Analysis Step

Table 5-1. Estimating Space for Work Files (CNTRL1 CNTRL2)

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
CNTRL1 CNTRL2	VB	6148	6152	The number of records varies depending on the contents of the old and new subschemas. However, the average file is not expected to exceed three cylinders on a 3350.

5.3.4.2 OPTALLOC Step

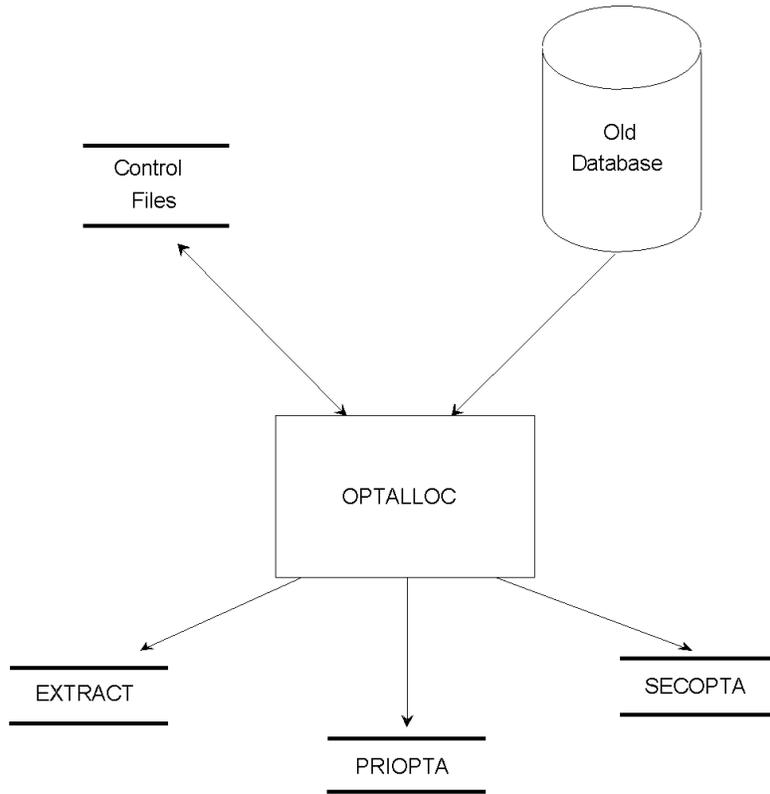


Figure 5-2. Work File Usage For OPTALLOC Step

Table 5-2. Estimating Space for Work Files (EXTRACT, PRIOPTA, SECOPTA)

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
EXTRACT	FB	22	6138	<p># of EXTRACT records =</p> <p>(# of EXTRACT records/record type (see note)) * (# of occurrences of that record type).</p> <p>The CA-IDMS/DB Analyzer Set Report lists the number of member record occurrences (for member records) and the number of set occurrences (for owner records). Extract records are not created for index pointers.</p>
PRIOPTA	FB	23 + length of longest CALC key in copy area	Multiple of LRECL closest to 6K.	<p># of PRIOPTA records = (CALCs + DIRECTs + VIAs) whose page range is the same as their owner's page range (see note).</p> <p>The length of the longest CALC key in any copy area must be added to the fixed length portion to determine the total length of a PRIOPTA record. (See CA-IDMS/DB Reorg message NLYZ007I in ANALYSIS Step).</p>
<p>Note: Refer to 4.3.2, "Audit Subreport--Set Linkage Recap for Area" on page 4-8 for additional information.</p>				
SECOPTA	FB	The greater of 34 or (34 + length of longest integrated index symbolic key) (See message NLYZ012I in ANALYSIS Step).	Multiple of LRECL closest to 6K.	<p># of SECOPTA records =</p> <p>(# of VIAs not in the same page range as the owner) + (# of all BL SR8 records)</p>

5.3.4.3 PRIALLOC Step

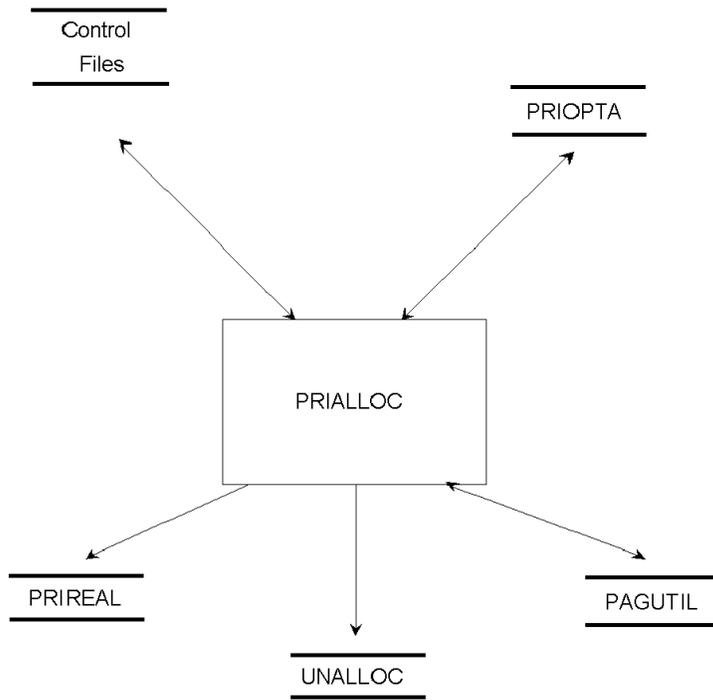


Figure 5-3. Work File Usage For PRIALLOC Step

Table 5-3. Estimating Space for Work Files (PRIREAL, UNALLOC, PAGUTIL)

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
PRIREAL	FB	22	6138	# of PRIREAL = # of PRIOPTA records that fit on its target page.
UNALLOC	FB	34 or 34+length of longest integrated index symbolic key.	Multiple of LRECL closest to 6K.	# of UNALLOC records = (# of PRIOPTA records) - (# of PRIREAL records). This is the number of overflow records.
PAGUTIL	F	8168	8168	# of PAGUTIL records = (3 + (pages-in-copy-area-01/1020) + (pages-in-copy-area-02/1020) ... + (pages-in-copy-area-nn/1020))

Note: 'pages-in-copy-area' refers to the new subschema.

PAGUTIL (Example)

Since PAGUTIL records never overlap area boundaries, there is one PAGUTIL per 1020 pages in an area. If the area contains less than 1020 pages, one PAGUTIL record is allocated for that area. If the area boundaries are not divisible by 1020, an additional PAGUTIL record is allocated for the remainder of pages in that area.

Example:

AREA-1 has 200 pages
AREA-2 has 2000 pages

$$\text{PAGUTIL} = 3 + (200/1020) + (2000/1020)$$

$$= 3 + 1 + 2$$

$$= 6 \text{ records}$$

Note: This file will not function properly if created with any other record size, block size, or record format. Disregard message "IEC161I 072-053..." which appears as an informational message in OS/390 when this is a VSAM file.

5.3.4.4 SECALLOC Step

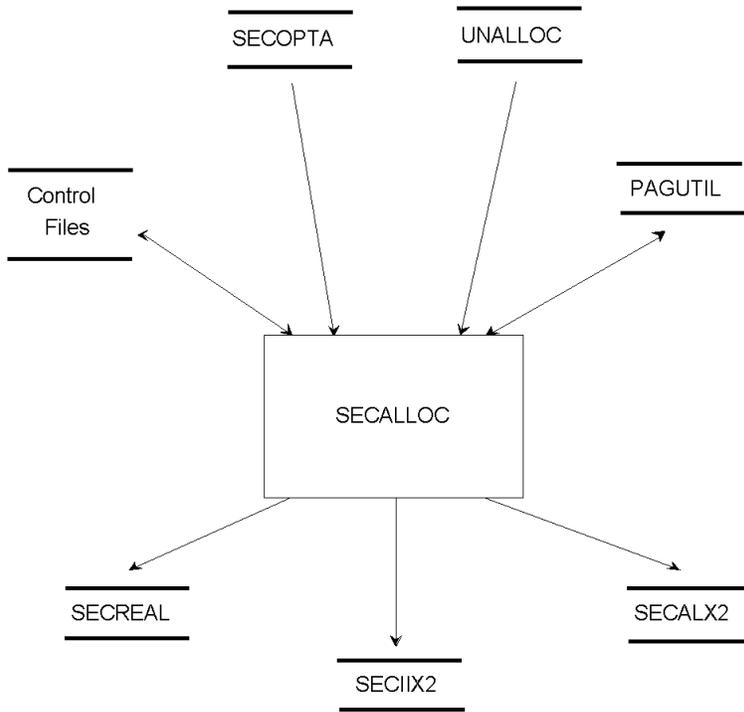


Figure 5-4. Work File Usage for SECALLOC Step

Table 5-4. Estimating Space for Work Files (SECREAL)

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
SECREAL	FB	22	6138	# of SECREAL records = (# of SECOPTA records) + (# of UNALLOC records) + (# of SR8 records created) + (# of entries in the BL SR8) + (# of entries for UL SR8's - sets sorted by DBKEY only)

Note: This calculation must be done for each indexed set in the area.

BL - Bottom Level

UL - Upper Level

Recap:

Number of records for each indexed set in the area =
 number of SECOPTA records
 +
 number of UNALLOC records
 +
 number of of SR8 records created
 +
 number of entries in bottom level SR8 records
 +
 (if sorted db-key) number of entries in upper level SR8 records.

The space needed because of SR8 (integrated index) records will never exceed one SECREAL record for each of the SR8s and bottom level entries in the existing database, plus (if sorted by db-key) the number of entries in the upper levels. These figures can be obtained from the CA-IDMS/DB Analyzer report, SR8 Index Statistics Section:

Upper Level--SR8 Records
 Bottom Level--SR8 Records
 Bottom Level-- Used Table Entries
 and if sorted db-key,
 Upper Level--Used Table Entries

You can use that total of records or do a more detailed calculation as shown in Figure 5-5 on page 5-16.

Table 5-5. Estimating Space for Work Files (SECALX2, SECIIX2)

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
SECIIX2	VB	12+length of largest possible SR8 record.	Either LRECL+4 or 6K, whichever is larger.	<p>The space required to store the new SR8 structure will not be greater than the space needed to store the existing SR8 structure. (Because of compression, a more exact estimate is not possible.)</p> <p>The amount of space is listed as "Total Bytes to Store Index" on the CA-IDMS/DB Analyzer SR8 Index Report, SR8 Index Statistics Section.</p>
SECALX2	VB	18	6138	<p>The space required will be no greater than the space for all existing upper and bottom level SR8 records. A more detailed calculation follows:</p> <p># of SECALX2 records = $(3 * \text{the \# of BL SR8 records}) +$ $(3 * \text{UL SR8 record}) +$ $(1 * \text{\# of UL SR8 entries}).$</p> <p>where:</p> <p>UL SR8 entries = $(\text{\# of SR8 records} - 1)$</p> <p>See Figure 5-5 on page 5-16 for more information.</p>

Note: These files are required only if you are copying an area with indexed sets. Space must be allocated for each indexed set in the area.

BL - Bottom Level SR8
UL - Upper Level SR8

For SECREAL and SECALX2 (optional detailed calculation): To calculate the number of bottom level SR8 records to be created (BL), see the CA-IDMS/DB Analyzer SR8 Index Report, SR8 Index Statistics Section. Divide the number of Used Table Entries (Bottom Level Statistics) in the existing database by the Maximum Number of Table Entries in the new subschema, and round up to the next whole number.

$$\text{BL} = \frac{\text{Bottom Level Statistics--Used Table Entries (existing DB)}}{\text{Maximum Number of Table Entries (new subschema)}}$$

The maximum number of table entries for the new subschema can be obtained from the Block Contains clause in the schema control statements for the new subschema.

If BL is greater than 1, find the number of upper level SR8 records to be created for this set by dividing the number of bottom level SR8 records (BL) by the maximum number of table entries and round up to the next whole number. If that result is greater than 1, divide the result by the maximum number of table entries and round up to the next whole number. Continue dividing by the maximum number of table entries and rounding up, until the result is equal to 1. The sum of the whole numbers obtained from these calculations is the number of upper level SR8 records required for this set.

5.3 Operational Flow and Considerations

For example:

From CA-IDMS/DB Analyzer SR8 Index Report, SR8 Index Statistics Section

Bottom Level Statistics--Number of Used Table Entries = 2100

Maximum Number of Table Entries (new subschema) = 10

Calculate:

$$\text{Bottom level SR8 records } \frac{2100}{10} = 210$$

$$\text{Upper level SR8 records } \frac{210}{10} = 21$$

$$\frac{21}{10} = 3 \text{ (rounded up)}$$

$$\text{(Top level SR8 record) } \frac{3}{10} = 1 \text{ (rounded up)}$$

SECREAL

number of SECOPTA records
+ number UNALLOC records
+ (210 + 21 + 3 + 1) total SR8 records
+ 2100 SR8 bottom level entries

SECALX2

3 * 210 bottom level SR8 records
+ 3 * (21 + 3 + 1) upper level SR8 records
+ 1 * (210 + 21 + 3) upper level SR8 entries

Figure 5-5. Calculating the Number of SR8 Records and Entries

5.3.4.5 MATCH Step

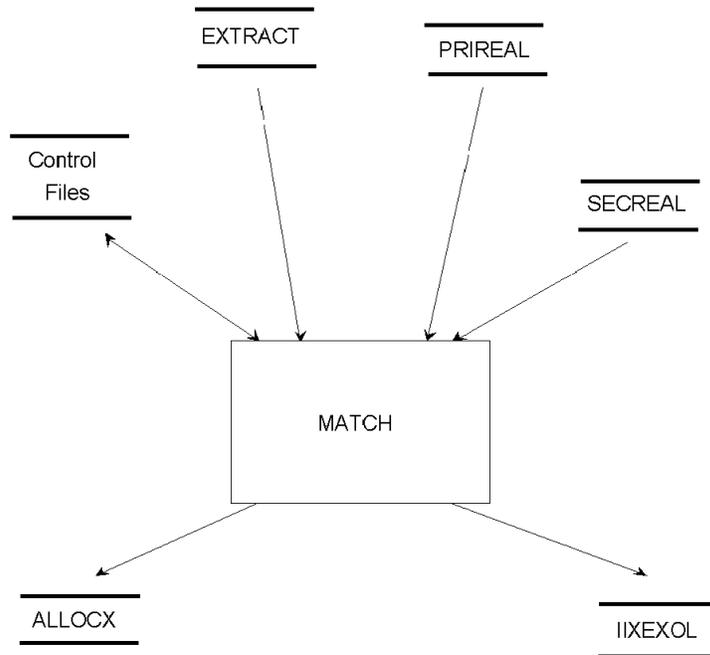


Figure 5-6. Work File Usage For Match Step

Table 5-6. Estimating Space for Work Files (ALLOCX, IIXEXOL)

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
ALLOCX	FB	18	6138	# of ALLOCX records = (# of EXTRACT records) + (# of PRIREAL records) + (# of SECREAL records).
IIXEXOL	FB	20	6140	See note. # of IIXEXOL records = (3) + (2 * BL SR8 records) + (1 * UL SR8 records)

Note: This file is required only if the area being copied contains an indexed set sorted by DB-Key.

BL - Bottom Level SR8

UL - Upper Level SR8

5.3.4.6 IIXSDBK Step

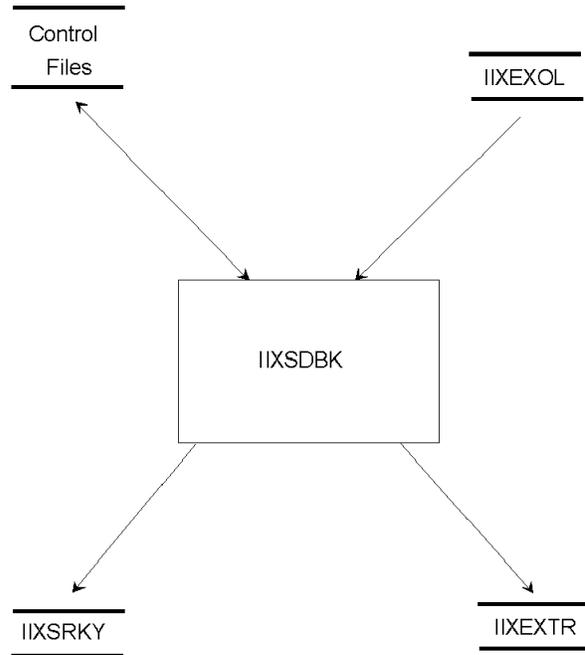


Figure 5-7. Work File Usage for IIXSDBK Step

Table 5-7. Estimating Space for Work Files (IIXSRKY IIXEXTR)

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
IIXSRKY	F	20	6140	The number of records is listed in message OPTX014I.
IIXEXTR	FB	18	6138	# of IIXEXTR records = (3 + (2 * BL SR8 record) + (1 * UL SR8 records)

Note: These files are needed only if the area being updated contain index sets sorted by db-key.

BL - Bottom Level SR8

UL - Upper Level SR8

5.3.4.7 UPDLINK Step

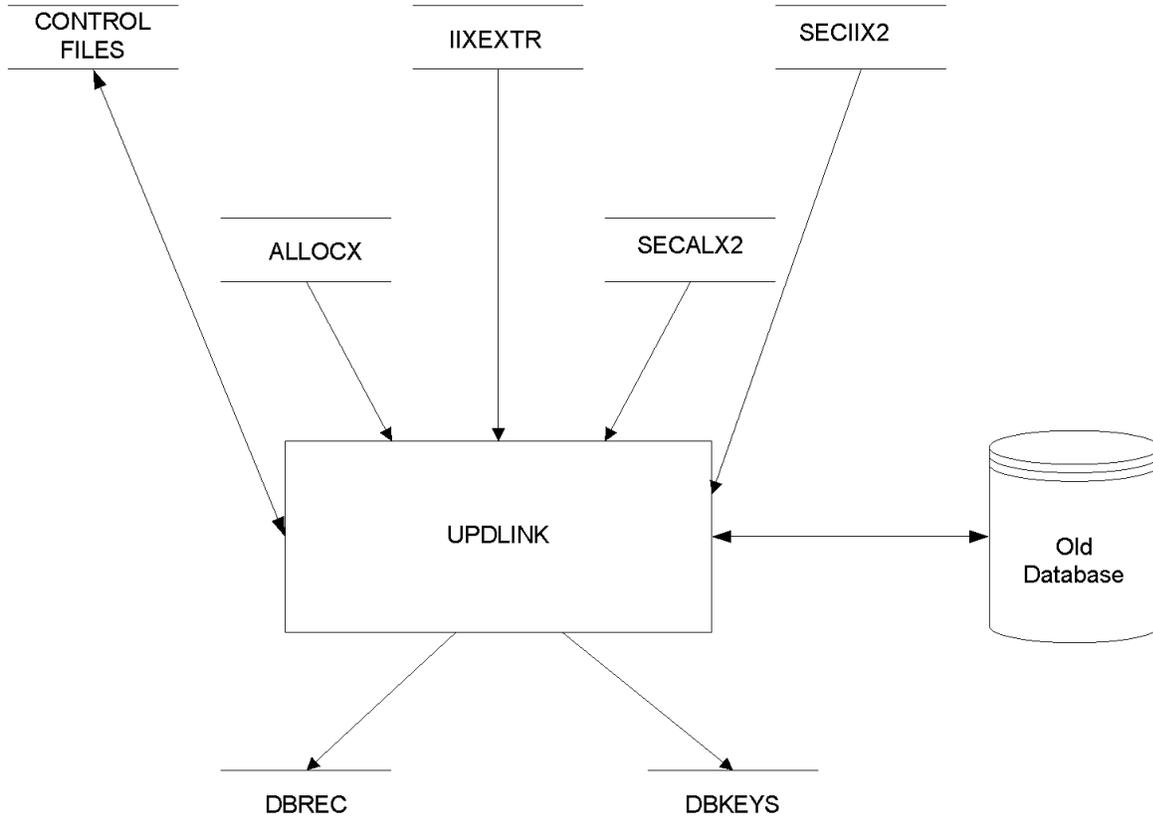


Figure 5-8. Work File Usage for UPDLINK Step

Table 5-8 (Page 1 of 2). Estimating Space for Work File (DBREC, DBKEY)

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
DBREC	VB	24+length of the longest record that fits on a single page in a copy area.	Either LRECL+4 or 6K, whichever is larger.	# of DBREC record = (((24 * # of occurrences) + prefix + data) record-01) + ⋮ (((24 * # of occurrences) + prefix + data) record-nn)

Table 5-8 (Page 2 of 2). Estimating Space for Work File (DBREC, DBKEY)

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
DBKEY	FB	16	BLOCK- NUM parameter	# of DBKEY records = ((# of records in reorganized areas) - (SR8 records)) * 16

5.3.4.8 STORE Step

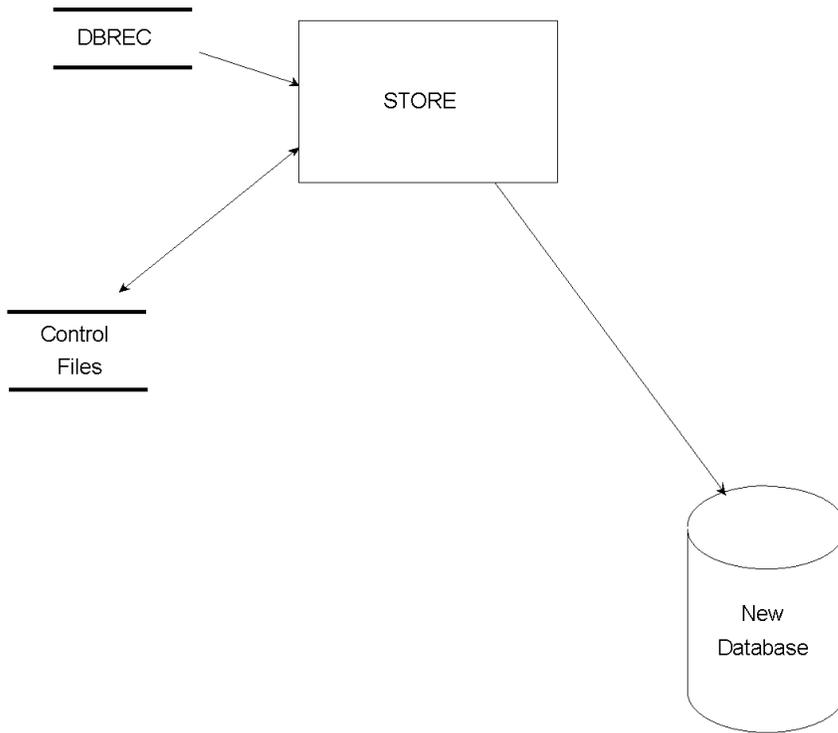


Figure 5-9. Work File Usage For STORE Step

5.3.5 Sort-Work Areas

There are several separate, internal sorts performed by CA-IDMS/DB Reorg. If you decide to assign your sort-work areas to disk, you will have to adjust space requirements to accommodate the largest of the sort input files. Table 5-9 on page 5-23 illustrates which files are needed for each step. For each step, space requirements for sort-work areas can be estimated by adding the space allocations for the sort input files listed with that step. Refer to your sort utility documentation to determine the sort space required for the given file sizes and formats. You can find information about estimating the space allocation for each file in Figure 5-1 on page 5-7 to Figure 5-9.

Table 5-9. Sort Work Areas

Step	Sort Input File(s)
PRIALLOC	PRIOPTA
SECALLOC	SECOPTA +UNALLOC
MATCH	EXTRACT +PRIREAL +SECREAL
IIXSDBK	IIXEXOL
UPDLINK	ALLOCX +IIXEXTR +SECALX2 +SECIIX2
STORE	DBREC

5.3.6 Summary of Procedures for Operating CA-IDMS/DB Reorg

1. Back up database.
2. Define new subschema.
3. Use IDMSBCF to allocate and format COPY areas in new subschema. (Can be done any time before STORE.)

Note: If you do not use DSPLYINT=NO, inform your system console operator of the valid reply options. CA-IDMS/DB Reorg operates with an out-standing operator reply. See 2.3.12, "DSPLYINT Parameter" on page 2-14.

4. Allocate Space for CNTRL1/CNTRL2. Analysis step.

a. PROCESS,
OLDSUB=*subschema-name*,
NEWSUB=*subschema-name*,
OLDDMCL=*dmcl-name*,
NEWDMCL=*dmcl-name*,
STOPAFTER=ANALYSIS
COPY,
AREA= *area-name*

b. First PROCESS stops here.

5. Allocate space for work files. OPTALLOC Step.

If sufficient disk space is not available for any file, it may be assigned to tape (except PAGUTIL). Use information from CA-IDMS/DB Reorg Audit Report, from reports by CA-IDMS/DB Analyzer, and from Figure 5-1 on page 5-7 to

Figure 5-9 on page 5-22. In OS/390, use IEFBR14 for deleting and allocating work files.

a. **PROCESS**,
OLDSUB=*subschemaname*,
NEWSUB=*subschemaname*,
OLDDMCL=*dmclname*,
NEWDMCL=*dmclname*,
STOPAFTER=CURRENT

b. Second Process stops here.

6. If sorted indexes are to be updated, allocate space for three work files. Use information from the OPTALLOC statistics to estimate space for IIXEXOL, IIXSRKY, and IIXEXTR.

a. **PROCESS**,
OLDSUB=*subschemaname*,
NEWSUB=*subschemaname*,
OLDDMCL=*dmclname*,
NEWDMCL=*dmclname*,
RESTART=NEXT,
STOPAFTER=UPDLINK

PRIALLOC Step
SECALLOC Step
MATCH Step
IIXSDBK Step
UPDLINK Step

b. Third PROCESS stops here.

7. If not done previously, use IDMSBCF to allocate and format COPY areas in new subschema. (Can be done anytime before STORE.)

8. **PROCESS**,
OLDSUB=*subschemaname*,
NEWSUB=*subschemaname*,
OLDDMCL=*dmclname*,
NEWDMCL=*dmclname*,
RESTART=NEXT,

STORE Step

Note: If you are using CA-IDMS/DB Reorg to change page ranges for one or more areas, **and** the old and new databases are defined in different DMCL's:

1. Run CA-IDMS/DB Reorg in Step mode.
2. All steps prior to the STORE step must specify the 'olddmcl' in the SYSIDMS parameters.

3. The STORE step must specify the 'newdmcl' in the SYSIDMS parameters.

5.4 CA-IDMS/DB Reorg Processing--OS/390 Environment

The following example illustrates a simple way to execute CA-IDMS/DB Reorg in an OS/390 environment. It shows the JCL needed to validate parameters, to obtain the Audit/Analysis Report for estimating work files, and to reorganize your database. The example is divided into these four processes:

1. Defining the areas which are being copied by running the CA-IDMS utility IDMSBCF.
2. Gathering information by running CA-IDMS/DB Reorg to validate parameters and to obtain the Audit/Analysis Report.
3. Allocating the work files.
4. Reorganizing the database.

See 5.3.6, “Summary of Procedures for Operating CA-IDMS/DB Reorg” on page 5-23 for a summary of the procedures. The summary includes the optional STOPAFTER and RESTART parameters, which allow you to run CA-IDMS/DB Reorg in increments.

Target or Distribution source library members USREXEC1, USREXEC2, USREXEC3, and USREXEC4 provide examples of JCL for executing CA-IDMS/DB Reorg.

5.4.1 Process 1--Defining the New Areas

Follow any of the procedures below that apply to the database you are reorganizing.

Create new COPY area--If you are creating a new area, the area must be defined in both the old and new subschemas in order for CA-IDMS/DB Reorg to execute the COPY statement. Therefore, in the existing (old) database you must allocate and format an area having the new name, using IDMSBCF. The area can be of minimum size (one page is sufficient). This formatting must be done before the OPTALLOC step.

Allocate and format COPY areas--Database areas which are being copied must be allocated and formatted in the new subschema with IDMSBCF, before initiating the STORE step of CA-IDMS/DB Reorg. Here are two suggested alternate times to allocate and format these areas:

1. Before initiating any steps of CA-IDMS/DB Reorg.
2. Immediately before the STORE step. Use STOPAFTER=UPDLINK.

5.4.2 Process 2--Gathering Information

This process gathers the information needed to determine the best strategy for a database reorganization. It initiates CA-IDMS/DB Reorg, examines user parameters and subschemas, stops CA-IDMS/DB Reorg after the ANALYSIS step, and produces the Audit/Analysis Report. In the control parameters which follow the JCL, it is necessary to include a COPY statement for each area that is to be reorganized.

```

/*          SAMPLE JCL TO OBTAIN AUDIT/ANALYSIS REPORT
/*
//USREORG      EXEC    PGM=USRDRVR
//STEPLIB     DD      DSN=your.loadlib,DISP=SHR
//            DD      DSN=idms.loadlib,DISP=SHR
/*          CONTROL WORK FILES
//dbmsdds     DD      DSN=work.files.dbmsdds,DISP=(,CATLG,DELETE),
//            VOL=SER=volser,UNIT=disk,SPACE=(number.tracks)
/*          PRINT FILES
//SYSLST      DD      SYSOUT=a          AUDIT REPORT
//DBMSDUMP    DD      SYSOUT=a          USER-REQUESTED DUMP
//SYSUDUMP    DD      SYSOUT=a          STANDARD ABEND DUMP
//SYSPRINT    DD      SYSOUT=a          INTERNAL DD
//DBMSOUT     DD      SYSOUT=a          INTERNAL DD
/*          PARAMETER INPUT TO DB/REORG
//SYSIDMS     DD      *
//            DMCL=o1ddmcl
//SYSIPT      DD      *
//            PROCESS,  OLDSUB=subschema-name,
//                      NEWSUB=subschema-name,
//                      OLDDMCL=dmc1-name,
//                      NEWDMCL=dmc1-name,
//                      STOPAFTER=ANALYSIS
//            COPY,    AREA=area-name
//
//
/*

```

Figure 5-10. Sample JCL to Obtain AUDIT/ANALYSIS Report

Table 5-10 (Page 1 of 2). Key for Sample JCL to Obtain AUDIT/ANALYSIS Report

Parameter	Description
<u>your.loadlib</u>	The dataset name of the OS/390 PDS library into which CA-IDMS/DB Reorg was installed.
<u>idms.loadlib</u>	The dataset name of the OS/390 PDS library in which your CA-IDMS modules reside. You <u>must</u> supply a separate statement for every OS/390 PDS library needed.
<u>dbmsdds</u>	The DDNAMEs required by the CA-IDMS/DB Reorg Utility. You must supply a DD statement for each of the control work files: CNTRL1, CNTRL2.

Table 5-10 (Page 2 of 2). Key for Sample JCL to Obtain AUDIT/ANALYSIS Report

Parameter	Description
work.files.dbmsdds	The dataset names assigned to the control work files used by the CA-IDMS/DB Reorg Utility. (It is suggested that each dataset name include the required DDNAME.)
volser	The volume serial number of the disk or tape to be used for the control work files.
disk	An appropriate unit designation for the work file.
number.tracks	The number of tracks needed for the control work files if they reside on disk (3 cylinders on a 3350 are usually sufficient).
a	The appropriate SYSOUT class for your installation.
olddmcl	DMCL of the database prior to reorganization.
subschema-name	Two subschema names are required. The first subschema name (OLDSUB=) describes the database before reorganization. The second subschema name (NEWSUB=) describes the database after reorganization.
dmcl-name	Two dmcl names are required. The first dmcl name (OLDDMCL=) describes the database before reorganization. The second dmcl name (NEWDMCL=) describes the database after reorganization.
area-name	The name of an area to be reorganized by CA-IDMS/DB Reorg. (You <u>must</u> include a COPY parameter statement for each area that is to be reorganized.)

5.4.3 Process 3--Allocating Work Files

This process allocates all of the work files to disk. One of CA-IDMS/DB Reorg's files, PAGUTIL, needs to be direct access, either VSAM or relative record BDAM. If the PAGUTIL file is VSAM, use the following control statements for IDCAMS. Refer to Target or Distribution source library member USRIDCAM.

```

DELETE (vsam.work.space) CLUSTER
DEFINE CLUSTER      ( -
                    NAME(vsam.work.space) -
                    RECORDS(record.numbers) -
                    NUMBERED -
                    RECORDSIZE(8168 8168) -
                    usetype -
                    VOLUMES(volume.name) -
                    ) -
DATA              ( -
                    NAME(vsam.work.space.DATA) -
                    )

```

Figure 5-11. Sample JCL to Allocate Work Files

If the PAGUTIL file is to be relative record BDAM, the following allocation JCL applies:

```

/*
//PAGUTIL DD DSN=DBREORG.PAGUTIL,DISP=(,CATLG),
            UNIT=DISK,SPACE=(8168,(record.numbers)),
            DCB=DSORG=DA
//PAGUT@@ DD DSN=*.PAGUTIL,VOL=REF=*.PAGUTIL,
            DISP=SHR,DCB=DSORG=DA

```

Figure 5-12. Sample JCL if PAGUTIL file is to be Relative Record BDAM

You can pre-allocate these data sets but the DCB=DSORG=DA parameter must be coded for each of these DD statements within the JCL of the step that actually accesses the dataset.

If CA-IDMS/DB Reorg is updating an integrated index sorted by db-key, space should be allocated for three additional files after the OPTALLOC step.

```

/*
/*      NON-VSAM WORK FILES
/*
/*      INSURE THAT ANY PREVIOUSLY CREATED WORK FILES ARE DELETED
/*
//DELETE EXEC PGM=IEFBR14
//dbmsdds DD DSN=work.files.dbmsdds,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//ALLOCATE EXEC PGM=IEFBR14
//dbmsdds DD DSN=work.files.dbmsdds,DISP=(,CATLG,DELETE),
//          VOL=SER=volser,UNIT=disk,
//          SPACE=(number.tracks)

```

Figure 5-13. NON-VSAM Work Files

Table 5-11. Key for Allocating Work Files JCL

Parameter	Description
a	The appropriate SYSOUT class for your installation.
vsam.work.space	The dataset name of the <u>one</u> VSAM file to be used by CA-IDMS/DB Reorg.
record.numbers	The number of records to be allocated in the VSAM or BDAM work space. See Table 5-3 on page 5-11
usetype	The appropriate type: REUSE or UNIQUE. If you specify UNIQUE, you must DELETE and DEFINE this cluster prior to each execution of the PRIALLOC step.
volume.name	Volume to contain the cluster or component.
dbmsdds	The DDNAMEs required by the CA-IDMS/DB Reorg Utility. You must supply a DD statement for each of these work files: PRIOPTA, SECOPTA, EXTRACT, PRIREAL, UNALLOC, SECREAL, ALLOCX, SECALX2, SECIIX2, and DBREC. If CA-IDMS/DB Reorg is updating an area containing integrated index sets sorted by db-key, you must allocate space for IIXEXOL, IIXSRKY, and IIXEXTR, after the OPTALLOC step.
work.files.dbmsdds	The dataset names assigned to the work files used by the CA-IDMS/DB Reorg Utility. (It is suggested that each dataset name include the required DDNAME.)
disk	An appropriate unit designation for the work file.
volser	The volume serial number of the disk or tape to be used for the work file.
number.tracks	The number of tracks needed for the work file if it resides on disk. This number is based on the number of records and the blocking factor of the work file and the track capacity of your disk.

5.4.4 Process 4--Reorganizing the Database

This process reorganizes the database by executing the reorganization process of CA-IDMS/DB Reorg as one OS/390 job step.

```

/*          SAMPLE JCL TO REORGANIZE A DATABASE
/*
//USREORG      EXEC    PGM=USRDRVR
//STEPLIB      DD     DSN=your.loadlib,DISP=SHR
              DD     DSN=idms.loadlib,DISP=SHR

/*
/*          YOUR DATABASE FILES
//dbfi01 DD  DSN=user.dbfi01,DISP=SHR

:

//dbfilnn      DD     DSN=user.dbfilnn,DISP=SHR
/*
/*          SORT WORK FILES
//SORTLIB      DD     DSN=your.sortlib,DISP=SHR
//sortwknn     DD     UNIT=disk,SPACE=(CYL,(sort.space))
/*
/*          WORK FILES
//PAGUTIL      DD     DSN=vsam.work.file,vsam-buffer,DISP=SHR
//dbmsdds      DD     DSN=work.files.dbmsdds,DISP=SHR
/*
/*          PRINT FILES
//SYSLSST      DD     SYSOUT=a          AUDIT REPORT
//DBMSDUMP     DD     SYSOUT=a          USER-REQUESTED DUMP
//SYSUDUMP     DD     SYSOUT=a          STANDARD ABEND DUMP
//SORTMSG      DD     SYSOUT=a          SORT MESSAGE OUTPUT
//SYSPRINT     DD     SYSOUT=a          DBMS INTERNAL DD
//DBMSOUT      DD     SYSOUT=a          DBMS INTERNAL DD
/*
/*          PARAMETER INPUT TO DB/REORG
//SYSIDMS      DD *
      DMCL=newdmc1
//SYSIPT       DD *
PROCESS,      OLDSUB=subschema-name,
              NEWSUB=subschema-name,
              OLDDMCL=dmc1-name,
              NEWMCL=dmc1-name,
              RESTART=NEXT

/*

```

Figure 5-14. Sample JCL to Reorganize a Database

Table 5-12 (Page 1 of 3). Key for Sample JCL to Reorganize a Database

Parameter	Description
your.loadlib	The dataset name of the OS/390 PDS library into which CA-IDMS/DB Reorg was installed.
idms.loadlib	The dataset name of the OS/390 PDS library in which your subschema(s) and DMCL(s) reside. You <u>must</u> supply a separate statement for every OS/390 PDS library needed.
dbfi01 /dbfilnn	The DDNAMEs of the database files to be accessed.

Table 5-12 (Page 2 of 3). Key for Sample JCL to Reorganize a Database

Parameter	Description
user.dbfil01 /user.dbfilnn	The dataset name of the file accessed by either the old subschema or the new subschema. You <u>must</u> supply a separate statement for every file that contains, or has set linkage with, an area named in a COPY statement.
your.sortlib	The dataset name of the OS/390 PDS library in which your sort modules reside. (Your environment may not require this DD statement.)
sortwknn	The DDNAMEs of the sort files. You <u>must</u> supply a separate statement for every sort file needed.
disk	The appropriate unit designation for your sort files.
sort.space	The space calculated on sort input needs if sort files are on disk.
vsam.work.file	The dataset name of the <u>one</u> VSAM file to be used by CA-IDMS/DB Reorg. (Refer to Figure 5-12 on page 5-29 if you prefer to use BDAM for this file.)
vsam-buffer	As a default, CA-IDMS/DB Reorg allocates 10 buffers to process the VSAM PAGUTIL file. To override this, code AMP=BUFND=number-of-component-buffers. The space required for these buffers must be available below the line. The maximum number of buffers that can be useful is equal to the number of pages in the largest COPY area divided by 1020.
dbmsdds	The DDNAMEs required by the CA-IDMS/DB Reorg Utility. You must supply a DD statement for each of these work files: CNTRL1, CNTRL2, PRIOPTA, SECOPTA, EXTRACT, PRIREAL, UNALLOC, SECREAL, ALLOCX, SECALX2, SECIIX2, and DBREC. If CA-IDMS/DB Reorg is reorganizing an index set sorted by db-key, or an area containing integrated index sets sorted by db-key, you must include IIXEXOL, IIXSRKY, and IIXEXTR.
work.files.dbmsdds	The dataset names assigned to the work files used by the DB/REORG Utility. (It is suggested that each dataset name include the required DDNAME.)
a	The appropriate SYSOUT class for your installation.
newdmcl	DMCL describing the database after the reorganization.

Table 5-12 (Page 3 of 3). Key for Sample JCL to Reorganize a Database

Parameter	Description
subschema-name	Two subschema names are required. The first subschema name (OLDSUB=) describes the database before reorganization. The second subschema name (NEWSUB=) describes the database after reorganization.
dmcl-name	Two <i>dmcl-names</i> are required. The first <i>dmcl-name</i> (OLDDMCL=) describes the database before reorganization. The second <i>dmcl-name</i> (NEWDMCL=) describes the database after reorganization.

5.5 CA-IDMS/DB Reorg Processing--VSE/ESA Environment

The following example illustrates a simple way to execute CA-IDMS/DB Reorg in a VSE/ESA environment.

It shows the JCL needed to validate parameters, to obtain the Audit/Analysis Report for estimating work files, and to reorganize your database in a VSE/ESA environment. The example is divided into these four processes:

1. Defining the areas which are being copied, by running the CA-IDMS utility, IDMSBCF.
2. Gathering information, by running CA-IDMS/DB Reorg to validate parameters and to obtain the Audit/Analysis Report.
3. Allocating the work files.
4. Reorganizing the database.

See 5.3.6, “Summary of Procedures for Operating CA-IDMS/DB Reorg” on page 5-23 for a summary of the procedures. The summary includes the optional STOPAFTER and RESTART parameters, which allow you to run CA-IDMS/DB Reorg in increments.

TOOLJCL members USREXEC1.S/USREXEC1.S, USREXEC2.S/USREXEC2.S, USREXEC3.S/USREXEC3.S, and USREXEC4.S/USREXEC4.S provide examples of JCL for executing CA-IDMS/DB Reorg.

5.5.1 VSE/ESA File Assignments

Even if you use a storage management tool such as CA-DYNAM, CA-IDMS/DB Reorg requires an ASSGN statement for every file except SORTWK*nn*. This ASSGN is required because CA-IDMS/DB Reorg has its own device-independent support which dynamically builds a DTF based on the device type indicated by the ASSGN. The logical unit for each file is shown in the Table 5-14 on page 5-38. Unless the ASSIGN specifies VSAM or BDAM, the file may be defined with either DLBL or TLBL.

5.5.2 VSE/ESA File Processing Alternate Method

As each file is opened, you can receive a message that a file is not VSAM. The message indicates that the dataset will be processed SAM instead of VSAM because CA-IDMS/DB Reorg was not able to find the dataset in the VSAM catalog. The allocation will not affect processing results. Any CA-IDMS/DB Reorg work files can be assigned as ESDS VSAM or as SAM. Do not use VSAM-managed SAM.

5.5.3 Process 1--Defining the New Areas

Follow the procedures below that apply to the database you are reorganizing.

Create new COPY area--The new area must be defined in both the old and new subschemas in order for CA-IDMS/DB Reorg to execute the COPY statement. Therefore, in the old subschema, you must allocate and format an area having the new name, using IDMSBCF. The area can be of minimum size (one page is sufficient). This initialization must be done before the OPTALLOC step.

Allocate and format COPY areas--Database areas that are being copied must be allocated and formatted in the new subschema with IDMSBCF, before initiating the STORE Step of CA-IDMS/DB Reorg. Here are two suggested alternate times to allocate and format these areas:

1. Before initiating any steps of CA-IDMS/DB Reorg.
2. Immediately before the STORE step. Use STOPAFTER=UPDLINK.

5.5.4 Process 2--Information Gathering

This process gathers the information needed to determine the best strategy for a database reorganization. It initiates CA-IDMS/DB Reorg, examines user parameters and subschemas, stops CA-IDMS/DB Reorg after the ANALYSIS step, and produces the Audit/Analysis Report. In the control parameters that follow the JCL it is necessary to include a COPY statement for each area that is to be reorganized.

```

// JOB DBREORG1
// OPTION LOG,PARTDUMP
*
*   LIBRARY DEFINITIONS
*
// DLBL   DBMS,'your.loadlib'
// EXTENT ,volser
// DLBL   IDMS,'idms.loadlib'
// EXTENT ,volser
// LIBDEF *,SEARCH=(DBMS.sublib,IDMS.sublib)
*
* CONTROL WORK FILES
*   THE NEXT FILES ARE CREATED IN THIS STEP AND USED
*   THROUGHOUT THE REMAINDER OF THE PRODUCT. IT IS SUGGESTED
*   THAT THE ACCOMPANYING SECTION ON FILE CREATION BE CONSULTED
*   IF THE USER WISHES TO ASSIGN THE FILES TO DISK.
*   DBMSNMS SHOULD BE "CNTRL1" AND "CNTRL2".
*
// DLBL   DBMSNMS,'WORK.FILES.dbmsnms',0,SD
// EXTENT SYSnnn
// ASSGN  SYSnnn,DISK,VOL=volser,SHR
*
*   CA INTERNAL FILES
*
// ASSGN  SYS012,SYSLST
// ASSGN  SYS013,SYSLST
*
*   PARAMETER INPUT TO DB/REORG
*
// DLBL  SYSIDMS,'#SYSIPT',0,SD
// EXEC  USRDRVR,SIZE=(USRDRVR,#K)
*
* SYSIDMS PARAMETERS
*INCLUDE DMCL=olddmcl
/*
*   _____
*   ——— DB/REORG CONTROL CARDS ———
*   _____
*
*   CONTROL CARD FORMAT IS FREE FORM. COMMENTS ARE INDICATED BY
*   AN ASTERISK IN COLUMN 1. THE PROCESS STATEMENT CONSISTING OF
*   AT LEAST THE OLDSUBSCHEMA AND NEWSUBSCHEMA PARAMETERS MUST BE
*   INCLUDED FOR THIS INITIAL RUN, AS WELL AS AT LEAST ONE COPY STATEMENT.
PROCESS,  OLDSUB=subschema-name,
          NEWSUB=subschema-name,
          OLDDMCL=dmcl-name,
          NEWDMCL=dmcl-name,
          OLDDBN=db-name,
          NEWDBN=db-name,
          STOPAFTER=ANALYSIS
COPY,    AREA=area-name

:
/*
* $$ E0J

```

Figure 5-15. Sample JCL to Obtain AUDIT/ANALYSIS Report

Table 5-13. Key for Sample JCL to Obtain AUDIT/ANALYSIS Report

Parameter	Description
your.loadlib	The data set name of the core image library into which you download CA-IDMS/DB Reorg.
volser	The volume serial number or the generic assignment of the disk volume on which the file, specified in the preceding DLBL statement, resides.
SYSnnn	The programmer logical units of the database files to be accessed.
idms.loadlib	The data set name of the core image library in which your DMCL and subschema reside.
dbms.sublib /idms.sublib	The sublibrary name of the VSE/ESA library specified in the previous file name.
olddmcl	The DMCL of the old database, before it is reorganized. describes the database before reorganization. The second subschema name (NEWSUB=) describes the
subschema-name	Two subschema names are required. The first subschema name (OLDSUB=) describes the database before reorganization. The second subschema name (NEWSUB=) describes the database after reorganization.
dmcl-name	Two dmcl names are required. The first dmcl name (OLDDMCL=) describes the database before reorganization. The second dmcl name (NEWDMCL=) describes the database after reorganization.
db-name	Two db-names are optional. The first db-name (OLDDBN=) describes the database before reorganization. The second db-name name (NEWDBN=) describes the database after reorganization.
area-name	The name of an area to be reorganized by CA-IDMS/DB Reorg. (You <u>must</u> include a COPY parameter statement for each area that is to be reorganized.)

5.5.5 Process 3--Allocating Work Files

Note: In a VSE/ESA environment, the CA-IDMS/DB Reorg PAGUTIL file needs to be VSAM.

This process allocates all of the work files to disk. When you are allocating space for the work files, you must include DLBL statements for every work file used by

CA-IDMS/DB Reorg and EXTENT and ASSGN statements for each file's corresponding logical unit. See Table 5-14 on page 5-38. If you are reorganizing an index area sorted by db-key, after the OPTALLOC step, you must allocate space for IIXEXOL, IIXSRKY, and IIXEXTR. See Target or Distribution source library member USRIDCAM.

```

*
*   SAMPLE JCL TO ALLOCATE VSAM WORK FILE - PAGUTIL
*
// DLBL   PAGUTIL,'vsam.work.space',0,VSAM
// EXTENT SYS014,volser,,reltrk,amount
// ASSGN  SYS014,DISK,VOL=volser,SHR
// EXEC   IDCAMS,SIZE=AUTO
        DELETE ('vsam.work.space') CLUSTER
        DEFINE CLUSTER ( -
                NAME('vsam.work.space') -
                RECORDS(record.numbers) -
                NUMBERED -
                RECORDSIZE(8168 8168) -
                usetype -
                VOLUMES(volume.name) -
                ) -
        DATA  ( -
                NAME('vsam.work.space.DATA') -
                )
/*
*

```

Figure 5-16. Sample JCL to Allocate VSAM Work File PAGUTIL

Table 5-14. Work Files and Corresponding Logical Units

dbmsnms	SYSnmm
CNTRL1	SYS010
CNTRL2	SYS011
PAGUTIL	SYS014
EXTRACT	SYS015
PRIOPTA	SYS016
SECOPTA	SYS017
PRIREAL	SYS018
UNALLOC	SYS019
SECREAL	SYS020
ALLOCX	SYS021
DBREC	SYS022
IIXEXOL	SYS025
IIXSRKY	SYS026
IIXEXTR	SYS027
SECALX2	SYS028
SECIX2	SYS029

Table 5-15. Key for Sample JCL to Allocate VSAM Work File PAGUTIL

Parameter	Description
vsam.work.space	The file ID of the <u>one</u> VSAM file to be used by CA-IDMS/DB Reorg. (It is suggested that this file ID include the name PAGUTIL.)
volser	The volume serial number or generic assignment of the disk volume on which the file, as specified in the preceding DLBL statement, resides.
reltrk	Relative track or relative block number: the starting position on the DASD for storage of the work file specified in the previous statement.
amount	The space allocation you need to store the file specified in the previous statement.
record.numbers	The number of records to be allocated to the VSAM work file.
usetype	The appropriate type: REUSE or UNIQUE. If you specify UNIQUE, you must DELETE and DEFINE this cluster prior to each execution of the PRIALLOC step.
volume.name	The volume to contain the cluster or component.

5.5.6 Process 4--Reorganizing the Database

This process reorganizes the database by executing the reorganization process of CA-IDMS/DB Reorg as one VSE/ESA job step.

```
*   SAMPLE JCL TO REORGANIZE A DATABASE USING NEW AREAS
*
*   LIBRARY DEFINITIONS
*
// DLBL  DBMS,'your.loadlib'
// EXTENT ,volser
// DLBL  IDMS,'idms.loadlib'
// EXTENT ,volser
// LIBDEF *,SEARCH=(DBMS.sublib,IDMS.sublib)
*
*   SORT WORK FILES
*
*       THERE ARE SIX SEPARATE SORTS IN THIS PORTION OF DB/REORG. IF
*       YOU ELECT TO ASSIGN THE SORTWKNN FILES TO DISK, ADJUST THE
*       SPACE REQUIREMENTS TO ACCOMODATE THE LARGEST OF THESE SORT
*       INPUT FILES. THE REORG STEPS AND SORT INPUTS ARE AS FOLLOWS:
*
// DLBL  SORTWK1,'SORT.WORK1',0,SD
// EXTENT SYS001,,,reltrk,amount
// ASSGN SYS001,DISK,VOL=volser,SHR
// DLBL  SORTWK2,'SORT.WORK2',0,SD
// EXTENT SYS002,,,reltrk,amount
// ASSGN SYS002,DISK,VOL=volser,SHR
```

```

// DLBL  SORTWK3,'SORT.WORK3',0,SD
// EXTENT SYS003,,reltrk,amount
// ASSGN SYS003,DISK,VOL=volser,SHR
// DLBL  SORTWK4,'SORT.WORK4',0,SD
// EXTENT SYS004,,reltrk,amount
// ASSGN SYS004,DISK,VOL=volser,SHR
*
*   USER DATABASE FILES
*
// DLBL  DBFIL01,'user.dbfile01',0,SD
// EXTENT SYSnnn
// ASSGN SYSnnn,DISK,VOL=volser,SHR

:
// DLBL  DBFILNN,'user.dbfilenn',0,SD
// EXTENT SYSnnn
// ASSGN SYSnnn,DISK,VOL=volser,SHR
*
*   VSAM WORK FILE
*
// DLBL  PAGUTIL,'vsam.work.space',0,SD
// EXTENT SYS014
// ASSGN SYS014,DISK,VOL=volser,SHR
*
*   DB/REORG WORK FILES
*
*   THERE ARE A NUMBER OF WORK FILES THAT ARE USED BY DB/REORG.
*   THESE FILES VARY IN SIZE SO IT IS SUGGESTED THAT THE USER
*   GUIDE OPERATION SECTION ON WORK FILES BE CONSULTED IF THE
*   USER WISHES TO ASSIGN THE FILES TO DISK.
*
*   A REMINDER:
*   THE FOLLOWING FILES ARE SEQUENTIAL AND SO TAPE MAY PROVE A
*   BETTER CHOICE IF DISK SPACE IS PROBLEMATIC.  ALSO, STANDARD
*   DCB PARAMATERS LIKE RECFM, LRECL AND BLKSIZE ARE HANDLED
*   INTERNALLY BY THE UTILITY AND SHOULD NOT BE CODED ON THE JCL.
*
// DLBL  dbmsnms,'work.files.dbmsnms',0,SD
// EXTENT SYSnnn
// ASSGN SYSnnn,DISK,VOL=volser,SHR
*
*   PARAMETER INPUT TO DB/REORG
*
// DLBL  DBMSNMS,'WORK.FILES.dbmsnms',0,SD
// EXTENT SYSnnn
// ASSGN SYSnnn,DISK,VOL=volser,SHR
*
*   CA INTERNAL FILES
*
// ASSGN SYS012,SYSLST
// ASSGN SYS013,SYSLST
*
*   PARAMETER INPUT TO DB/REORG
*
// DLBL SYSIDMS,'#SYSIPT',0,SD
// EXEC  USRDRVR,SIZE=(USRDRVR,#k)
*
*   SYSIDMS PARAMETERS
*   INCLUDE DMCL=newdmcl

```

```

/*
PROCESS, OLDSUB=subschema-name,
         NEWSUB=subschema-name,
         OLDDMCL=dmc1-name,
         NEWDMCL=dmc1-name,
         OLDDBN=db-name,
         NEWDBN=db-name,
         RESTART=NEXT
/*
/*
* $$ E0J

```

Table 5-16 (Page 1 of 2). Key for Sample JCL to Reorganize a Database

Parameter	Description
your.loadlib	The data set name of the core image library into which you downloaded CA-IDMS/DB Reorg.
idms.loadlib	The data set name of the core image library in which your DMCL and subschema reside.
volser	The volume serial number or generic assignment of the disk volume on which the file, specified in the previous statement, resides.
dbms.sublib /idms.sublib.	The sublibrary name of the VSE/ESA library specified in the previous file name.
reltrk	The relative track or relative block number: the starting position on the DASD for storage of the work file specified in the previous statement.
amount	The space allocation you need to store the file specified in the previous statement.
user.dbfile01 /user.dbfilenn	The file ID of each of the database files to be accessed.
dbmsnms	The file names required by the CA-IDMS/DB Reorg utility described earlier in this chapter.
work.files.dbmsnms	The file ID assigned to the work files used by the CA-IDMS/DB Reorg utility.
SYSnnn	The programmer logical units of the database files to be accessed.
#k	The number of kilobytes of storage to be made available to the sort phase.
newdmcl	DMCL describing the database after the reorganization.
subschema-name	Two subschema names are required. The first subschema name (OLDSUB=) describes the database before reorganization. The second subschema name (NEWSUB=) describes the database after reorganization.

Table 5-16 (Page 2 of 2). Key for Sample JCL to Reorganize a Database

Parameter	Description
dmcl-name	Two dmcl names are required. The first dmcl name (OLDDMCL=) describes the database before reorganization. The second dmcl name (NEWDMCL=) describes the database after reorganization.
area-name	The name of an area to be reorganized by CA-IDMS/DB Reorg. (You <u>must</u> include a COPY parameter statement for each area that is to be reorganized.)

5.6 CA-IDMS/DB Reorg Processing--VM/ESA Environment

The following example illustrates a sample way to execute CA-IDMS/DB Reorg. It shows the steps needed to validate parameters, to obtain the Audit/Analysis Report for estimating work files, and to reorganize your database. The example is divided into these four processes:

1. Defining the areas which are being copied by running the CA-IDMS utility IDMSBCF.
2. Gathering information by running CA-IDMS/DB Reorg to validate parameters and to obtain the Audit/Analysis Report.
3. Allocating the work files.
4. Reorganizing the database.

See 5.3.6, “Summary of Procedures for Operating CA-IDMS/DB Reorg” on page 5-23 for a summary of the procedures. The summary includes the optional STOPAFTER and RESTART parameters, that allow you to run CA-IDMS/DB Reorg in increments.

5.6.1 Process 1--Defining the New Areas

Follow procedures below that apply to the database you are reorganizing.

Create new COPY area--The new area must be defined in both the old and new subschemas in order for CA-IDMS/DB Reorg to execute the COPY statement. Therefore, in the existing (old) database, you must allocate and format an area having the new name, using IDMSBCF. The area can be of minimum size (one page is sufficient). This formatting must be done before the OPTALLOC step.

Allocate and format COPY areas--Database areas which are being copied must be allocated and formatted in the new subschema with IDMSBCF, before initiating the STORE step of CA-IDMS/DB Reorg. Here are two suggested alternate times to allocate and format these areas:

1. Before initiating any steps of CA-IDMS/DB Reorg.
2. Immediately before the STORE step. Use STOPAFTER=UPDLINK.

5.6.2 Process 2--Information Gathering

This process gathers information needed to determine the best strategy for a database reorganization. It initiates CA-IDMS/DB Reorg, examines user parameters and subschemas, stops CA-IDMS/DB Reorg after the ANALYSIS step, and produces the Audit/Analysis Report. A sample EXEC is supplied as member USREXEC. Sample control parameters for the parameter input file (FILEDEF SYSIPT) are shown below. Note that the display interval parameter, DSPLYINT, **must not** be specified in a VM/ESA environment.

```

PROCESS,OLDSUB=subschema-name,
NEWSUB=subschema-name,
OLDDMCL=dmc1-name,
NEWDMCL=dmc1-name,
STOPAFTER=ANALYSIS,
COPY,AREA=area-name

:
/* */
TRACE OFF; SIGNAL ON ERROR
/* */
CA_LOADLIB_FN      = 'yourlib'
IDMS_LOADLIB_FN    = 'idmslib'
SORTLIB_FN         = 'sortlib'
/* */
/* Link and access the Minidisks containing the required librari(e)s */
/* and database file(s). */
/* */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'GLOBAL LOADLIB ' CA_LOADLIB_FN IDMS_LOADLIB_FN
'GLOBAL TXTLIB ' SORTLIB_FN
/* */
/* Insert FILEDEFS as needed to define all required database files. */
/* */
'FILEDEF dbfil01 DISK fn ft fm (dcb'
'FILEDEF dbfilnn DISK fn ft fm (dcb'
/* */
/* Print and Dump files. */
/* */
'FILEDEF SYSLSST PRINTER'
'FILEDEF SYSUDUMP PRINTER'
'FILEDEF SNAPIT PRINTER'
'FILEDEF DBMSOUT PRINTER'
'FILEDEF SYSPRINT PRINTER'
'FILEDEF SORTMSG PRINTER'
'FILEDEF DBMSDUMP PRINTER'
/* */
/* Control work files. */
/* */
'FILEDEF CNTRL1 DISK USREXEC CNTRL1 fm'
'FILEDEF CNTRL2 DISK USREXEC CNTRL2 fm'
'FILEDEF PRIOPTA DISK USREXEC PRIOPTA fm'
'FILEDEF SECOPTA DISK USREXEC SECOPTA fm'
'FILEDEF EXTRACT DISK USREXEC EXTRACT fm'
'FILEDEF PRIREAL DISK USREXEC PRIREAL fm'
'FILEDEF SECREAL DISK USREXEC SECREAL fm'
'FILEDEF UNALLOC DISK USREXEC UNALLOC fm'
'FILEDEF SECALX2 DISK USREXEC SECALX2 fm'
'FILEDEF SECIIX2 DISK USREXEC SECIIX2 fm'
'FILEDEF ALLOCX DISK USREXEC ALLOCX fm'
'FILEDEF IIXEXOL DISK USREXEC IIXEXOL fm'
'FILEDEF IIXSRKY DISK USREXEC IIXSRKY fm'
'FILEDEF IIXEXTR DISK USREXEC IIXEXTR fm'
'FILEDEF DBREC DISK USREXEC DBREC fm'
/* */
/* You must create a file 'SYSIDMS INPUT A' containing the SYSIDMS */
/* parameters you use to specify your runtime environment. */
/** Include statement DMCL='newdmc1' in your SYSIDMS parameters */

```

```

/*
'FILEDEF SYSIDMS DISK SYSIDMS INPUT A'
/*
/* You must create a file 'USREXEC SYSIPT A' containing the input */
/* parameter statements prior to executing this EXEC. */
/*
'FILEDEF SYSIPT DISK USREXEC SYSIPT A'
/*
/* Insert FILEDEF statements for SORT work space as required by */
/* your SORT product. */
/*
/* Decide now whether the PAGUTIL file will be VSAM or BDAM. */
/* To use VSAM for the PAGUTIL file : */
/* REMEMBER — VSAM files must be predefined using AMSERV control */
/* statements. See sample EXEC USRVSAM, and also the */
/* Operations section of the CA-IDMS/DB Reorg Users */
/* Guide. */
/* 1) Change: */
/* vsam_fm To the filemode of the accessed minidisk containing */
/* your VSAM catalog(s) and file(s). */
/* vsamcat To the dataset name of your VSAM catalog. */
/* pagutil To the dataset name of your VSAM PAGUTIL file. */
/* 2) Remove the comments from around the next 3 statements. */
/* 'DLBL IJSYSCT vsam_fm DSN vsamcat'
/* 'DLBL IJSYSUC vsam_fm DSN vsamcat'
/* 'DLBL PAGUTIL vsam_fm DSN pagutil ( VSAM' */
/* To use BDAM for the PAGUTIL file : */
/* 1) Change: */
/* cuu To the appropriate minidisk device address. */
/* nn To the number of cylinders required to hold PAGUTIL. */
/* fm To the filemode letter for the temp disk. */
/* 2) Remove the comments from around the next 9 statements. */
/* 'DEF T3380 cuu CYL nn'
/* PUSH 'PAGUTL'
/* PUSH 'YES'
/* 'FORMAT cuu fm ( BLK 4K'
/* 'RESERVE USREXEC PAGUTIL fm6'
/* 'FILEDEF PAGUTIL DISK USREXEC PAGUTIL fm6' ,
/* ( DSORG DA'
/* 'FILEDEF PAGUT@@ DISK USREXEC PAGUTIL fm6' ,
/* ( DSORG DA' */
SAY 'STARTING DATABASE REORGANIZATION'
SIGNAL OFF ERROR
'EXECOS OSRUN USRDRVR'
USREORG_RC = RC
IF USREORG_RC > 4
THEN DO
CALL ERROR
END
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME DBREORG LISTING'
'CP SPOOL PRINTER OFF'
SAY 'USREXEC FINISHED WITH A RETURN CODE OF' USREORG_RC
'GLOBAL LOADLIB'
'GLOBAL TXTLIB'
'FILEDEF * CLEAR'
EXIT USREORG_RC
/*****/
ERROR:

```

```

/*****/
ERROR_RC = RC
TRACE OFF; SIGNAL OFF ERROR
SAY 'NON-ZERO RETURN CODE ENCOUNTERED IN EXEC AT LINE' SIGL
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME DBREORG LISTING'
'CP SPOOL PRINTER OFF'
'GLOBAL LOADLIB'
'GLOBAL TXTLIB'
'FILEDEF * CLEAR'
EXIT ERROR_RC
/*

```

Table 5-17 (Page 1 of 2). Key for Sample EXEC to Run CA-IDMS/DB Reorg

Parameter	Description
yourlib	The file name of the load library into which you downloaded CA-IDMS/DB Reorg.
idmslib	The file name of the load library containing your CA-IDMS subschema and DMCL modules.
sortlib	The file name of the text library containing your sort modules.
dbfil01 /dbfilnn	The name of each of the database files to be accessed.
fn ft m	The file name, file type, and file mode (respectively) of the relevant file. The file mode number of your database file(s) must be 6.
dcb	The data control block (DCB) information for the files.
cuu	The appropriate minidisk device address for the BDAM PAGUTIL file.
fm ft fm6	The file name, file type, and file mode (respectively) of the BDAM PAGUTIL file. The file mode number must be 6.
newdcm1	DMCL describing the database after the reorganization.
subschemaname	Two subschema names are required. The first subschema name (OLDSUB=) describes the database before reorganization. The second subschema name (NEWSUB=) describes the database after reorganization.
dmcl-name	Two dmcl names are required. The first dmcl name (OLDDMCL=) describes the database before reorganization. The second dmcl name (NEWDMCL=) describes the database after reorganization.

Table 5-17 (Page 2 of 2). Key for Sample EXEC to Run CA-IDMS/DB Reorg

Parameter	Description
area-name	The name of an area to be reorganized by CA-IDMS/DB Reorg. (You <u>must</u> include a COPY parameter statement for each area that is to be reorganized.)

Note: You must decide whether you will use VSAM or BDAM for the PAGUTIL file. Review and alter the sample EXEC appropriately.

5.6.3 Process 3--Allocating Work Files

The CA-IDMS/DB Reorg's file PAGUTIL needs direct access, either VSAM or BDAM. If the PAGUTIL file is VSAM, the following JCL explanation applies.

VSAM files written under VM/ESA are written using VSE/ESA VSAM. To use VSAM under VM/ESA, you must have OS/390 or VSE/ESA format disks defined and accessible in your VM/ESA configuration. A sample EXEC USRVSAM to access the appropriate minidisk and create the required VSAM space appears below.

```
TRACE OFF; SIGNAL ON ERROR
/*                                     */
DELETE          -
                (vsam.work.space) -
                CLUSTER
DEFINE CLUSTER ( -
                NAME(vsam.work.space) -
                RECORDS(record.numbers) -
                NUMBERED -
                RECSZ(8168 8168) -
                usetype -
                VOLUMES(volser) -
                ) -
DATA           ( -
                NAME(vsam.work.space.DATA) -
                )
```

Figure 5-17. Sample EXEC to Create VSAM Space on VM/ESA

Figure 5-18 shows a sample input file to the AMSERV command shown in Figure 5-17. Name this file **mastcat AMSERV fm**.

```
DEFINE MASTERCATALOG
      (NAME(mastcat)--
      VOLUME (dsk____)--
      CYL (n)--
      FILE (IJSYSCT))
```

Figure 5-18. Sample Input File to the AMSERV Command

Once you have VSAM space defined and allocated, the VSAM PAGUTIL file can be defined with the command:

```
AMSERV vsamwork
```

Figure 5-19. Sample AMSERV Command

Figure 5-20 shows a sample input file for the AMSERV command to define the VSAM PAGUTIL file. The file is named: *vsamwork AMSERV fm*. See sample statements in member USRDCAM.

```
DEL      (vsam.work.space)CLUSTER
DEFINE  CLUSTER (NAME(vsam.work.space)--
        RECORDS (record.numbers)
        NUMBERED--
        RECORDSIZE (8168 8168)--
        usetype--
        VOLUMES (dskxxx))
        DATA (NAME(vsam.work.space.DATA))
```

Figure 5-20. AMSERV Command to Define VSAM PAGUTIL File

If the PAGUTIL file is to be relative BDAM, use the JCL in Figure 5-21.

```
DEF T3380 cuu CYL nn
PUSH 'PAGUTL'
FORMAT cuu fm (BLK 4K
RESERVE USREXEC PAGUTIL fm6
FILEDEF PAGUTIL fm ft fm6 (DSORG DA
FILEDEF PAGUT00 fm ft fm6 (DSORG DA
```

Figure 5-21. JCL if PAGUTIL File is to be Relative Record BDAM

Table 5-18 (Page 1 of 2). Key for VM/ESA JCL

Parameter	Description
vsam.work.space	The file name of the VSAM PAGUTIL file to be used by CA-IDMS/DB Reorg.
record.numbers	The number of records to be allocated in the VSAM work space. See Table 5-3 on page 5-11.
usetype	The appropriate type: REUSE or UNIQUE. If you specify UNIQUE, you must DELETE and DEFINE this cluster prior to each execution of the PRIALLOC step.
mastcat	The file name of the VSAM master catalog.
dskxxx	The name of the minidisk on which to allocate a the VSAM files.

Table 5-18 (Page 2 of 2). Key for VM/ESA JCL

Parameter	Description
n	The number of cylinders to be allocated for the master catalog.
cuu	The appropriate minidisk device address for the BDAM PAGUTIL file.
nn	The number of cylinders required to hold the BDAM PAGUTIL file, based upon a block size of 8168. See Table 5-3 on page 5-11 for the number of PAGUTIL records.
fm	The file mode for the relevant file.
fm ft fm6	The file name, file type, and file mode (respectively) of the BDAM PAGUTIL file. The file mode number must be 6.

Note: PAGUT@@ must have the same file id as PAGUTIL.

5.6.4 Process 4--Reorganizing the Database

This process reorganizes the database. Sample parameters for the parameter input file (FILEDEF SYSPIPT) are shown below. Note that the display interval parameter, DSPLYINT, must not be specified in a VM/ESA environment.

```
PROCESS, OLDSUB=subschema.name,
NEWSUB=subschema-name,
OLDDMCL=dmc1-name,
NEWMCL=dmc1-name,
RESTART=NEXT
```

Figure 5-22. Sample Parameters for the Parameter Input File

This process uses the same sample EXEC to run CA-IDMS/DB Reorg as shown in 5.6.2, "Process 2--Information Gathering" on page 5-43.

Chapter 6. Messages

6.1 Condition Codes	6-4
6.2 Audit Report	6-5
6.3 VSE/ESA File Processing Alternate Method	6-6
6.4 Message Text	6-7

This chapter lists all messages generated by CA-IDMS/DB Reorg that appear in the Audit Report and on the Status Display. All messages listed in the Audit Report are preceded by a unique eight-character code. The message code is in an *xxxxnnms* format, where *xxxx* is the ID of the module issuing the message, *nnn* is a number for the message within the module, and *s* is the severity code for the message. The severity code will be either I, W, or E. Severity codes are explained below. Condition codes are explained on the following page.

(I) Informative--A message with a severity code of I is an informative message. Informative messages need no remedial action.

(W) Warning--A message with a severity code of W is a warning message. Warning messages report conflicting parameter data or various processing conditions.

In most cases, when a warning message is issued, CA-IDMS/DB Reorg will finish executing the step in which the warning was generated, but it will not execute any subsequent reorganization steps. However, if the warning refers to an invalid value in a parameter statement, CA-IDMS/DB Reorg will execute the Analysis step so that you will receive an Audit/Analysis report.

(E) Error--A message with a severity code of E is an error message. An error message reports an irrecoverable condition. In most cases, when CA-IDMS/DB Reorg encounters an error, all processing terminates immediately. If the error is syntactic or semantic, however, CA-IDMS/DB Reorg will continue processing until it reaches the end of the current step.

6.1 Condition Codes

In an OS/390 environment, the following condition codes are set at the end of the job:

0--Only informational messages were produced. This indicates successful processing.

4--Warning messages were produced. Processing may have been successful but the circumstances must be evaluated.

8--Parameter error messages were produced. Processing was unsuccessful.

12 or 16--Other errors were detected. Processing was unsuccessful.

6.2 Audit Report

Messages are listed on the Audit Report in the sequence in which they were generated by CA-IDMS/DB Reorg.

Parameter-related messages appear in a two-column format. The left column contains the message code and text; the right column shows user input. Part of the information in the right column may be underscored by asterisks (*****) to show the precise location of an error. Information may also be underscored by x's (xxxx) to indicate parameters that were not validated because an error(s) was encountered. Processing messages appear in the left column. For more information about the format of the Audit Report, see Chapter 4, "System Output" on page 4-1.

6.3 VSE/ESA File Processing Alternate Method

Occasionally, you will receive a message that a file is not VSAM. The message indicates that the dataset will be processed SAM instead of VSAM because CA-IDMS/DB Reorg was not able to find the dataset in the VSAM catalog. The allocation will not affect processing results. Note that PAGUTIL must be a VSAM file.

6.4 Message Text

AUDT001I STEP: step-name, STARTED

Reason: The reorganization step indicated began executing.

Action: None.

AUDT002I STEP: step-name, ENDED, CC=n

Reason: The reorganization step that has just ended and its return code.

Action: None.

**AUDT003I CURRENT STEP: step-name, IN SUBSTEP substep-name, DUR
hh.mm.ss**

Reason: The current reorganization step being processed, the substep it is in, and the amount of time it has been there.

Action: None.

AUDT004I AREA: area-name,PAGE RANGE=nnnnnnnnnn-nnnnnnnnnn

Reason: The page range for the area being processed.

Action: None.

AUDT005I CURRENT PAGE -nnnnnnnnnn (nnn%)

Reason: The number of the current page and the percentage of pages that have been processed.

Action: None.

AUDT006I number-of-records RECORDS IN INPUT FILE

Reason: The number of records contained in the input file.

Action: None.

AUDT007I number-of-records RECORDS READ (nnn%)

Reason: The number of the current record and the percentage of records that have been read.

Action: None.

AUDT008I INDEX SET nnn OF nnn, index-set-name

Reason: The index set being processed and its sequence in the total number of index sets.

Action: None.

AUDT009I number-of-records IXDET RECORDS READ

Reason: The number of IXDET records read.

Action: None.

AUDT010E UNABLE TO FIND MESSAGE NUMBER message-id IN PROGRAM module-name MESSAGE TABLE

Reason: A message number cannot be found in the designated module message table.

Action: Contact Computer Associates Product Support.

AUDT011E INVALID COMMUNICATION SWITCH nnnn PASSED TO PROGRAM USRAUDT

Reason: USRAUDT received an invalid communication switch.

Action: Contact Computer Associates Product Support.

AUDT012I DB/REORG IS BETWEEN STEPS PERFORMING A GENERAL SERVICE FUNCTION

Reason: The utility is still executing, but it is not performing one of the primary steps.

Action: None.

CALL901E PARAMETER LIST ERROR - PGM=module-name

Reason: Module could not be called because of an internal problem.

Action: Contact Computer Associates Product Support.

CALL902E NOT FOUND IN LIBRARY - PGM=module-nam

Reason: Module cannot be located in the available load or core-image libraries.

Action: Locate the load or core-image library that contains module and make it available to the job.

CALL903E NOT ENOUGH STORAGE - PGM=module-name

Reason: Module cannot be loaded due to insufficient GETVIS area (VSE/ESA only).

Action: Resubmit the job with a smaller SIZE= parameter or run in larger partition.

CALL904E ANCHOR TABLE LOAD FAILURE - PGM=GSSANKR

Reason: An internal error has been detected.

Action: Contact Computer Associates Product Support.

CALL905E UNANTICIPATED RETURN CODE=n PGM=module-name

Reason: An internal error has been detected.

Action: Contact Computer Associates Product Support.

DBST001E PROGRAM NAME program-name

Reason: A DBIO error has occurred. The program indicated was executing at the time of the error.

Action: The action required depends on interpretation of the DBIO error code given in Message: DBST002E.

DBST002E ERROR STATUS error-code

Reason: A DBIO error has occurred.

Action: To find the meaning of the first two digits of the error code, refer to the CA-IDMS major error codes listed in the *CA-IDMS Error Codes and Messages Guide* or the supplemental list in Figure 6-1 on page 6-11. If the third and fourth digits of the error code fall within the ranges 00-99, see the list of CA-IDMS minor error codes found in the *CA-IDMS Error Codes and Messages Guide*. Otherwise, see Figure 6-1 on page 6-11.

DBST003E RECORD NAME record-name

Reason: A DBIO error has occurred. The record indicated was the last one processed successfully before the error occurred.

Action: The action required depends on interpretation of the DBIO error code given in Message: DBST002E.

DBST004E AREA NAME area-name

Reason: A DBIO error has occurred. The area indicated was the last one processed successfully before the error occurred.

Action: The action required depends on interpretation of the DBIO error code given in Message: DBST002E.

DBST005E ERROR SET set-name

Reason: A DBIO error has occurred. The set indicated was being processed when the error occurred.

Action: The action required depends on interpretation of the DBIO error code given in Message: DBST002E.

DBST006E ERROR RECORD record-name

Reason: A DBIO error has occurred. The record indicated was being processed when the error occurred.

Action: The action required depends on interpretation of the DBIO error code given in Message: DBST002E.

DBST007E ERROR AREA area-name

Reason: A DBIO error has occurred. The area indicated was being processed when the error occurred.

Action: The action required depends on interpretation of the DBIO error code given in Message: DBST002E.

DBST008E CURRENT DBKEY.....db-key

Reason: A DBIO error has occurred. The value of the current db-key when the error occurred is indicated.

Action: The action required depends on interpretation of the DBIO error code given in Message: DBST002E.

DBST009E DIRECT DBKEY.....db-key

Reason: A DBIO error has occurred. The value of the direct db-key when the error occurred is indicated.

Action: The action required depends on interpretation of the DBIO error code given in Message: DBST002E.

DBIO MAJOR ERROR STATUS CODES

GP = FIND/OBTAIN PAGE

PR = PAGE RETURN

DBIO MINOR ERROR STATUS CODES

A1 = New page returned

A2 = Function not supported

A3 = Invalid page range

A4 = Invalid qualifier

A5 = Page not current of transaction

A6 = Area not bound

A7 = Page return not on

A8 = Page return already on

A9 = Fast scan not on

B1 = Fast scan already on

B2 = SR4 chain error

B3 = SR2/SR3 chain error

B4 = HDR/FTR page number mismatch

B5 = Error closing file

B6 = Record not logically deleted

B7 = SMP currency error with page return on

B8 = Bad 'line space used' length in footer

B9 = No space in page

C1 = Duplicate DBKEY

C2 = Invalid Store

C3 = Skipped index

C4 = Broken chain

C5 = IXDET member of multiple sets

C6 = Ignored page reserve

C7 = Invalid record length. Discrepancy in RDW lengths encountered during attempt to MODIFY RECORD. Run CLEANUP utility to remove LDELS from all areas to be processed by REORG.

C8 = DMCL/SCHEMA CA-IDMS release mismatch

C9 = Decompress error

D1 = Invalid record length

D3 = FSTRIGGER not less than FSBUFFER, or FSTRIGGER equal to 0

D4 = Invalid page displacement

D5 = Invalid request

D7 = GSDTPARM cannot be loaded

D8 = Record not found in PSUB

D9 = Segment names in the DBNAME table entry is mismatched with current DMCL

DA = DBNAME not valid or segment name cannot be used as DBNAME

Minor return codes prefixed with an I to indicate an internal error:

I1 = Invalid parameter list

I2 = Function not supported

I3 = File already open/closed

I4 = Insufficient work space--record read is larger than subschema description

I5 = Invalid currency data

I6 = VSAM block error

I7 = Bad fast scan DMCL block

I8 = file not assigned

Figure 6-1. DBIO Error Codes

**DBST010E SUBSCHEMA CONTAINS AREAS WITH MIXED PAGE GROUPS
OR MIXED RADIX - NOT SUPPORTED**

Reason: An attempt was made to ready an area with a different page group to those areas already readied.

Action: CA-IDMS/DB Reorg cannot be run against a subschema that contains mixed page groups.

**DEXT001E ERROR FOUND BY STEP INITIALIZATION
ROUTINE--USERSUBS**

Reason: An irrecoverable error condition has been detected by USRDEXT.

Action: Contact Computer Associates Product Support.

DEXT002I IIX SORTED DBKEYS/EXTRACTS READ number-of-records

Reason: The number of records read from the IIXEXOL file.

Action: None.

DEXT003I ALLOCATION EXTRACT RECORDS WRITTEN number-of-records

Reason: The number of records written to the IIXEXTR file.

Action: None.

DEXT004I IIX SORTED DBKEYS READ number-of-records

Reason: The number of records read from the IIXSRKY file.

Action: None.

DEXT005I IIX SORTED DBKEYS WRITTEN number-of-records

Reason: The number of records written to the IIXSRKY file.

Action: None.

**DEXT006E COULD NOT CREATE FORMAT2 ALLOCATION EXTRACT
RECORD BECAUSE EXTRACT INVALID**

Reason: The allocation extract file was created by CA-IDMS/DB Reorg, but has been corrupted.

Action: Verify that the allocation extract file (IIXEXOL) is valid and resubmit the job.

DRV001E ERROR CALLING module-name, CALL PARAMETER ERROR

Reason: The specified module cannot be called because of an internal error.

Action: Contact Computer Associates, Inc. Product Support.

DRV002E ERROR CALLING module-name, PROGRAM NOT FOUND

Reason: The specified program cannot be located in the available library.

Action: Locate the library that contains the module and make it available to the job. Resubmit the job.

DRV003E ERROR CALLING module-name, NOT ENOUGH STORAGE TO LOAD PROGRAM

Reason: The specified program cannot be loaded because of insufficient region size.

Action: Increase the region size and resubmit the job.

DRV004E UNANTICIPATED RETURN CODE r PGM = module name

Reason: An internal error has been detected.

Action: Contact Computer Associates Product Support.

GSFL001E ERROR OCCURRED DURING OPEN OF FILE file-name - GSSFILE RETURN CODE IS n1,n2,n3,n4

Reason: The file handler is unable to open the file.

Action: See Table 6-1 on page 6-47 and Table 6-2 on page 6-48 for an explanation and appropriate action for the return codes indicated.

GSFL002E ERROR OCCURRED DURING READ OF FILE file-name - GSSFILE RETURN CODE IS n1,n2,n3,n4

Reason: The file handler is unable to read the file.

Action: See Table 6-1 on page 6-47 and Table 6-2 on page 6-48 for an explanation and appropriate action for the return codes indicated.

GSFL003E ERROR OCCURRED DURING WRITE TO FILE file-name - GSSFILE RETURN CODE IS n1,n2,n3,n4

Reason: The file handler is unable to write to the file.

Action: See Table 6-1 on page 6-47 and Table 6-2 on page 6-48 for an explanation and appropriate action for the return codes indicated.

GSFL004E ERROR OCCURRED DURING UPDATE OF FILE file-name - GSSFILE RETURN CODE IS n1,n2,n3,n4

Reason: The file handler is unable to update the file.

Action: See Table 6-1 on page 6-47 and Table 6-2 on page 6-48 for an explanation and appropriate action for the return codes indicated.

**GSFL005E ERROR OCCURRED DURING CLOSE OF FILE file-name -
GSSFILE RETURN CODE IS n1,n2,n3,n4**

Reason: The file handler is unable to close the file.

Action: See Table 6-1 on page 6-47 and Table 6-2 on page 6-48 for an explanation and appropriate action for the return codes indicated.

GSFL999I file-id IS NOT VSAM - WILL TRY QSAM

Reason: The indicated file is not a VSAM file. The message is preceded by a system message indicating an open error for a VSAM file. If the attempt to open the file for QSAM processing is successful, CA-IDMS/DB Reorg will continue with normal processing.

Note: The CA-IDMS/DB Reorg file PAGUTIL needs direct access, either VSAM or BDAM.

Action: None.

LMSG901E MESSAGE NOT IN TABLE.....message-id module-name

Reason: Message table module does not contain an entry with the named message-id.

Action: Contact Computer Associates Product Support.

LMSG902E MESSAGE TABLE NOT FOUND.....module-name

Reason: Module cannot be located in the available load or core-image libraries.

Action: Locate the load or core-image library that contains module and make it available to the job.

LMSG903E NOT ENOUGH STORAGE FOR TABLE.....module-name

Reason: Module cannot be loaded due to insufficient GETVIS area (VSE/ESA only).

Action: Resubmit the job with a smaller SIZE= parameter or run with a larger partition.

LMSG904E BAD MESSAGE PARM OR TABLE.....module-name

Reason: An internal processing error has been detected while processing the named message table.

Action: Contact Computer Associates Product Support.

LMSG905E UNANTICIPATED RETURN CODE.....nnnnn

Reason: An internal processing error has been detected by the message table processor.

Action: Contact Computer Associates Product Support.

**MTCH001E ERROR FOUND BY STEP INITIALIZATION ROUTINE -
USRSUBS**

Reason: An irrecoverable error condition has been detected by USRMTCH.

Action: Contact Computer Associates Product Support.

MTCH002I ALLOCATION EXTRACT RECORDS READ.....number-of-record

Reason: The number of allocation extract records read from the EXTRACT, PRIREAL, and SECREAL files.

Action: None.

**MTCH003I ALLOCATION EXTRACT RECORDS WRITTEN
....number-of-records**

Reason: The number of allocation extract records written.

Action: None.

**MTCH004I IIX DUPLICATE DBKEYS/EXTRACTS WRITTEN number-of-
records**

Reason: The number of IIXEXOL records written.

Action: None.

MTCH007E MISSING ALLOCATION EXTRACT RECORDS

Reason: The allocation extract file was not created by CA-IDMS/DB Reorg, or it is a legitimate file that has been corrupted.

Action: Verify that the allocation extract file is valid. The allocation extract file consists of the EXTRACT, PRIREAL, and SECREAL files, sorted on first six bytes.

**MTCH008E POINTER EXTRACT DOES NOT MATCH ANY ALLOCATION
OR LDEL RECORD**

Reason:

1. The allocation extract file was created by CA-IDMS/DB Reorg, but has been corrupted.
2. The old database file contains integrity errors.
3. The old or new subschema is a partial description of the database.

Action:

1. Verify that the allocation extract file is valid. The allocation extract file consists of the EXTRACT, PRIREAL, and SECREAL files.
2. Use CA-IDMS/DB AUDIT to ensure that there are no integrity errors in the old database.
3. Verify that all records and sets in the old database are defined in the subschemas.

MTCH009E COULD NOT CREATE FORMAT2 ALLOCATION EXTRACT RECORD BECAUSE FORMAT1 ALLOCATION INVALID

Reason: The allocation extract file was created by CA-IDMS/DB Reorg, but has been corrupted.

Action: Verify that the allocation extract file (EXTRACT, PRIREAL, and SECREAL files) is valid and resubmit the job.

MTCH010E NO SET RECORD FOUND WHOSE ID MATCHES THE SET ID ON THE INPUT RECORD. SAVE SNAP AND CALL COMPUTER ASSOCIATES.

Reason: An extract record on either of the EXTRACT, PRIREAL, or SECREAL input files contains a set-ID that does not match to any subschema set-ID's.

Action: Contact Computer Associates Product Support.

NKWP001E UNRECOGNIZABLE MAJOR KEYWORD

Reason: The major keyword underscored by asterisks (*****) in the Audit Report is misspelled or is in error and cannot be processed.

Action: Correct the spelling of the keyword and resubmit the job.

NKWP002E DUPLICATE MINOR KEYWORD ENTERED

Reason: The minor keyword underscored by asterisks (*****) in the Audit Report has already been entered on the parameter statement underscored.

Action: Delete the duplicate minor keyword phrase and resubmit the job.

NKWP003E AN EQUAL SIGN (=) AND VALUE MUST FOLLOW THE KEYWORD

Reason: The required equal sign and value after the keyword are missing.

Action: Enter an equal sign followed by an appropriate value at the location underscored by an asterisk (*) in the Audit Report and resubmit the job.

NKWP005E MISSING RIGHT PARENTHESIS; PHRASE TERMINATED

Reason: A set of open and closed parentheses is required.

Action: Enter a right parenthesis to enclose the phrase or list, then resubmit the job.

NKWP006E REPEATED LIST IMPROPERLY CONTINUED

Reason: The last phrase on this statement is a list enclosed in parentheses that ends with a comma. However, the next statement entered does not continue the list.

Action: Supply a continuation card and resubmit the job.

NKWP007E MISSING CONTINUATION CARD

Reason: The previous statement ended with a comma, but this statement begins with a major keyword.

Action: Supply a continuation card and resubmit the job.

NKWP008E PRIORITY MUST CONTAIN NO MORE THAN 500 VALUES

Reason: You supplied more than 500 record types and/or record names in the priority list.

Action: Correct and resubmit.

NKWP009E VALUE MUST BE xxxxxxxxxxxxxxxxxxxxxxxx

Reason: Value underscored by an asterisk (*) in the Audit Report is incorrect for the keyword.

Action: Supply the appropriate value specified in the message and resubmit the job.

NKWP010E KEYWORD PHRASES NOT SEPARATED BY COMMA

Reason: Each keyword phrase must be separated by a comma.

Action: Supply a comma at the location underscored by an asterisk (*) in the Audit Report and resubmit the job.

NKWP011E UNRECOGNIZABLE MINOR KEYWORD

Reason: The minor keyword underscored by asterisks (*****) in the Audit Report is misspelled or is in error and was not processed.

Action: Correct the spelling of the minor keyword and resubmit the job.

NKWP012E MINOR KEYWORD OMITTED

Reason: A minor keyword was omitted at the location underscored by an asterisk (*) in the Audit Report.

Action: Enter the missing minor keyword and resubmit the job.

NKWP015E THESE KEYWORD(S) MUST BE SPECIFIED:xxxxxxxxxxxxxxxxxxxx

Reason: The minor keyword(s) listed in the message was omitted. CA-IDMS/DB Reorg cannot continue processing unless the keyword(s) shown is entered.

Action: Supply the missing keyword(s) and resubmit the job.

NKWP018E MISSING VALUE NEED xxxxxxxxxxxxxx

Reason: A keyword was entered without a value. The variable portion of the error message lists the acceptable values.

Action: Enter the correct value at the location underscored by an asterisk (*) in the Audit Report and resubmit the job.

NKWP019E VALUE ENTERED PREVIOUSLY

Reason: The value underscored by asterisks (****) on the Audit Report was already entered in the list. CA-IDMS/DB Reorg cannot process this parameter unless you supply two different values.

Action: Correct the value and resubmit the job.

NKWP020E VALUE MUST BE ASCENDING

Reason: The value underscored by asterisks (***** on the Audit Report must be in ascending numerical sequence from the previous value in the list.

Action: Rearrange the values in the list in ascending sequence and resubmit the job.

NKWP021E VALUE MUST BE DESCENDING

Reason: The value underscored by asterisks (***** on the Audit Report must be in descending numerical sequence from the previous value in the list.

Action: Rearrange the values in the list in descending sequence and resubmit the job.

NKWP024E SECOND-LEVEL PRIORITY MUST CONTAIN AT LEAST ONE VALUE.

Reason: A list or sublist cannot be empty. It must contain at least one valid record type or record name.

Action: Insert one or more record types or record names in the PRIORITY statement at the location underscored by an asterisk (*) in the Audit Report.

NKWP027E SECOND-LEVEL-PRIORITY MUST CONTAIN NO MORE THAN 10 VALUES

Reason: You supplied more than 10 record types and/or record names in a single, second-level-priority list. The eleventh entry is underscored by an asterisk (*) in the Audit Report.

Action: Break the second-level-priority list into several second-level-priority lists and resubmit the job.

NKWP050E GSSNKWP INTERNAL ERROR. NOTIFY COMPUTER ASSOCIATES PRODUCT SUPPORT

Reason: An internal error has been detected.

Action: Contact Computer Associates Product Support.

**NKWP051E ERROR OCCURRED DURING CALL OF xxxxxxxx - GSSCALL
RETURN CODE IS nn**

Reason: Module call processor error occurred.

Action: Possible return codes are:

- 4--Improper parameter list. See CALL901E.
- 8--Module not found in library. See CALL902E.
- 12--Not enough storage. See CALL903E.
- 16--Anchor table load failure. See CALL904E.

**NKWP052E ERROR OCURRED DURING LOAD OF xxxxxxxx - GSSLOAD
RETURN CODE IS nn**

Reason: Module load processor error occurred.

Action: Contact Computer Associates Product Support.

**NKWP053E ERROR OCCURRED DURING WRITE OF xxxxxxxx - GSSPRINT
RETURN CODE IS nn**

Reason: Error occurred during call of printer module.

Action: Contact Computer Associates Product Support.

NKWP061E DD STATEMENT MISSING FOR DDNAME: xxxxxxxx

Reason: A JCL DD card must be supplied.

Action: Correct the JCL and resubmit the job.

NKWP062E LOGICAL UNIT NOT ASSIGNED: xxxxxxxx

Reason: An ASSSGN card must be supplied.

Action: Correct the JCL and resubmit the job.

NKWP063E SYSIPT LRECL MUST BE 80.

Reason: A parameter file with an LRECL other than 80 was supplied.

Action: Create a parameter file with an LRECL of 80 and resubmit the job.

**NKWP064E ERROR OCCURRED DURING OPEN OF xxxxxxxx FILE -
GSSFILE RETURN CODE IS n1,n2,n3,n4**

Reason: The file handler is unable to open the file.

Action: See Table 6-1 on page 6-47 and Table 6-2 on page 6-48 for an explanation and appropriate action for the indicated return codes.

**NKWP065E ERROR OCCURRED DURING READ OF xxxxxxxx FILE -
GSSFILE RETURN CODE IS n1,n2,n3,n4**

Reason: The file handler is unable to read the file.

Action: See Table 6-1 on page 6-47 and Table 6-2 on page 6-48 for an explanation and appropriate action for the indicated return codes.

**NKWP066E ERROR OCCURRED DURING CLOSE OF xxxxxxxx FILE -
GSSFILE RETURN CODE IS n1,n2,n3,n4**

Reason: The file handler is unable to close the file.

Action: See Table 6-1 on page 6-47 and Table 6-2 on page 6-48 for an explanation and appropriate action for the indicated return codes.

NKWP080W SKIPPING UNTIL NEXT RECOGNIZABLE STATEMENT

Reason: Because of a previous error, the phrase(s) underscored by the capital letter X in the Audit Report has not been processed.

Action: None.

NKWP081W SKIPPING UNTIL NEXT RECOGNIZABLE MINOR KEYWORD

Reason: Because of a previous error, the value(s) underscored by the capital letter X in the Audit Report has not been processed.

Action: None.

NKWP091I INPUT PARAMETER STATEMENT

Reason: A CA-IDMS/DB Reorg input parameter statement has been read. The entire 80-byte image is displayed. Columns 1-72 will be processed.

Action: None.

NKWP092I CONTINUATION STATEMENT

Reason: The parameter statement continues a previous statement entered.

Action: None.

NKWP093I COMMENT STATEMENT

Reason: The parameter statement contains an asterisk (*) as the first non-blank character. The asterisk indicates that this is a comment statement, and it will not be processed.

Action: None.

NKWP094I END OF PARAMETER INPUT

Reason: All parameters that were entered have been read and processed by CA-IDMS/DB Reorg.

Action: None.

NKWP095I BLANK PARAMETER STATEMENT

Reason: The parameter statement contains blanks in columns 1-72 and will not be processed.

Action: None.

NKWP096E UNMATCHED APOSTROPHE. NEED xxxxxxxxxxxx

Reason: A value was entered that contains an odd number of apostrophes. Literal values must begin and end with single apostrophes. To represent a single apostrophe within a literal, you must enter two apostrophes. The variable portion of the error message lists the acceptable values.

Action: Correct the value underscored by asterisks (****) and resubmit the job.

NKWP097E BAD LITERAL FORMAT. NEED xxxxxxxxxxxx

Reason: A literal value was entered with improperly placed apostrophes. Literal values must begin and end with single apostrophes. To represent a single apostrophe within a literal, you must enter two apostrophes. The variable portion of the error message lists the acceptable values.

Action: Correct the value underscored by asterisks (****) and resubmit the job.

**NLYZ001E WRONG VERSION OF USRNLYZ MODULE, DB/REORG
release-number-1 USRNLYZ release-number-2**

Reason: The release numbers of CA-IDMS/DB Reorg's main program (USREORG) and USRNLYZ do not match. This error occurs if a new release of CA-IDMS/DB Reorg is partially installed over a prior release, or if two releases of CA-IDMS/DB Reorg are present and the available load or core-image libraries caused mixed releases of the program modules to be used.

Action: Verify that CA-IDMS/DB Reorg is correctly installed and that the proper load or core-image libraries are used. Resubmit the job.

**NLYZ002E ERROR FOUND BY STEP INITIALIZATION ROUTINE -
USRSUBS**

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

NLYZ003E NOSWEEP REQUESTED FOR AREA area-name. OWNER AND MEMBER(S) OF SET set-name ARE IN A NOSWEEP AREA

Reason: You requested a NOSWEEP for an area containing a record owning a multi-member set that also has a member record in a NOSWEEP area.

Action: Remove the NOSWEEP parameter statement for the area and resubmit the job.

NLYZ004E SUBSCHEMA DESCRIPTION OF record-name RECORD MISSING

Reason: The subschema does not have a description of the record.

Action: Correct the subschema and resubmit the job.

NLYZ005I END OF ANALYSIS REPORTS

Reason: The Analysis Reports have been completed.

Action: None.

NLYZ006I MAXIMUM RECORD SIZE FOR NEW SUBSCHEMA.....nnnn

Reason: Size information.

Action: None.

NLYZ007I MAXIMUM CALC KEY SIZE FOR OLD SUBSCHEMA.....nnnn

Reason: Size information

Action: None.

NLYZ008I MAXIMUM IX-DETAIL SIZE FOR OLD SUBSCHEMA.....nnnn

Reason: Size information

Action: None.

NLYZ009I MAXIMUM IX-DETAIL SIZE FOR NEW SUBSCHEMA.....nnnn

Reason: Size information.

Action: None.

NLYZ010I new/old SUBSCHEMA subschema-name WAS COMPILED compile-date compile-time, DMCL dmcl-name, DBNAME dbname.

Reason: Subschema compile information.

Action: None.

NLYZ012I MAXIMUM SYMBOLIC KEY SIZE FOR INDEXED SET.....nnnn

Reason: Size information.

Action: None.

NLYZ013I MAXIMUM RECORD SIZE FOR AN SR8 RECORD.....nnnn

Reason: Size information

Action: None.

NLYZ014E IIX SET index-set-name CONTAINS HI-LO PAGE RANGE NOT CORRESPONDING TO RANGE OF ANY AREA

Reason: The OR52 subschema control block, for the indicated index set, points to an AM57 subschema control block whose page ranges do not correspond to any AC56 page ranges in the subschema.

Action: Contact Computer Associates Product Support. To verify the page ranges, you can browse the subschema load module to find the OR52 control block for the indicated set. The last fullword in the OR52 indicates the offset to the AM57. The HI-LO page ranges are in the fifth and sixth fullwords of the AM57. Compare these page ranges to the page ranges in the AC56 control blocks.

OPTX001I OBTAINS FOR AREA SWEEP number-of-records

Reason: The number of records obtained through area sweeps.

Action: None.

OPTX002I OBTAINS FOR SET WALKnumber-of-records

Reason: The number of records obtained through set walks.

Action: None.

OPTX007I EXTRACT RECORDS WRITTEN number-of-records

Reason: The number of EXTRACT records written.

Action: None.

OPTX008I PRIOPTA RECORDS WRITTEN number-of-records

Reason: The number of PRIOPTA records written.

Action: None.

OPTX009I SECOPTA RECORDS WRITTEN number-of-records

Reason: The number of SECOPTA records written.

Action: None.

**OPTX010E WRONG VERSION OF USROPTX MODULE, DB/REORG
release-number-1 USROPTX release-number-2**

Reason: The release numbers of CA-IDMS/DB Reorg's main program (USREORG) and USROPTX do not match. This error occurs if a new release of CA-IDMS/DB Reorg is partially installed over a prior release, or if two releases of CA-IDMS/DB Reorg are present and the available load or core-image libraries cause mixed releases of the program modules to be used.

Action: Verify that CA-IDMS/DB Reorg is correctly installed and that the proper load or core-image libraries are used. Resubmit the job.

OPTX011E ERROR FOUND BY STEP INITIALIZATION ROUTINE - USRSUBS

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

OPTX012E record-name REC SYM KEY CALCED TO page-number WHICH IS OUTSIDE REC PG RANGE OF page-number--page-number

Reason: CALC record cannot be stored on suggested page because page number returned is outside record type's page range.

Action: If GSSCALC was relinked to include IDMSCLCX, validate that link executed correctly. Validate also that IDMSCLCX routine is returning valid page numbers. If no problems exist, call Computer Associates Product Support.

OPTX013I NUMBER OF SORTED DBKEYS AT BOTTOM LEVEL number-of-records

Reason: The number of bottom level records for IIX sorted db-keys with duplicate symbolic keys.

Action: Use this figure in the calculation of the space needed for the IIXEXOL and IIXEXTR work files.

OPTX014I NUMBER OF IIXSRKY RECORDS THAT WILL BE CREATED number-of-records

Reason: The number of records that will be created for the IIXSRKY work file.

Action: Use the number of records to estimate the space needed for the IIXSRKY work file.

OPTX015I NUMBER OF IIX SORTED DBKEYS POINTING TO UPPER LEVELS number-of-records

Reason: The number of IIX sorted db-key containing duplicates that point to upper levels.

Action: Use the number in the calculation to estimate space needed for the IIXEXOL and IIXEXTR files.

**OPTX016I STEP IIXSDBK WILL BE EXECUTED AND DD STATEMENTS
1) IIXEXOL 2) IIXSRKY 3) IIXEXTR MUT BE DEFINED**

Reason: CA-IDMS/DB Reorg found an area containing integrated index sets sorted by db-key. The IIXDBK step will be executed.

Action: You must allocate space for the work files IIXEXOL, IIXSRKY, and SPFEXTR. See 5.1, "Overview" on page 5-3.

OPTX017E SYSTEM OWNED IIX SET set-name DOES NOT CORRESPOND TO ANY SET DEFINITION

Reason: While sweeping the old database CA-IDMS/DB Reorg encountered an SR7 record whose name (or CALC key) does not match any set name in any OR52 in the subschema.

Action: Check the database to be sure it is correct. Ensure that the old subschema contain a definition for set-name.

OPTX018E IIX SET index-set-name CONTAINS HI-LO PAGE RANGE NOT CORRESPONDING TO RANGE OF ANY AREA

Reason: The OR52 subschema control block, for the indicated index set, points to an AM57 subschema control block whose page ranges do not correspond to any AC56 page ranges in the subschema.

Action: Contact Computer Associates Product Support. To verify the page ranges, you can browse the subschema load module to find the OR52 control block for the indicated set. The last fullword in the OR52 indicates the offset to the AM57. The HI-LO page ranges are in the fifth and sixth fullwords of the AM57. Compare these page ranges to the page ranges in the AC56 control blocks.

OPTX019E record-name/set-name PAGE RANGE IS ONLY ONE PAGE page-number AND THAT IS SMP SO CANNOT STORE

Reason: The indicated record or set has a page range of one page and that page is a space management page. The record or set cannot be stored.

Action: Modify the schema to increase the page range.

OPTX020E record-name RETRIEVED WALKING SET set-name BUT record-name NOT DEFINED IN THIS SET

Reason: The indicated record was retrieved while walking the indicated SR8 set, but the record is not defined as a member of the set.

Action: Contact Computer Associates Product Support.

OPTX021E SR8 RECORD IN IIX SET set-name OVERFLOWED BUFFER DURING DECOMPRESSION OF SYMBOLIC KEYS

Reason: Decompression of symbolic keys in an SR8 record required more buffer space than is available.

Action: Check whether the SR8 records are correct. Contact Computer Associates Product Support.

OPTX022E SR8 RECORD IN IIX SET set-name TARGETED PAGE page-number IS NOT IN IIX SET PAGE RANGE

Reason: The number of the target page for an SR8 record does not fall within the page range of the owner of the indicated integrated index set.

Action: Contact Computer Associates Product Support.

OPTX023W THE DATABASE CONTAINS nnn OCCURRENCES OF RECORD ID id, WHICH IS NOT DEFINED IN THE OLD SUBSCHEMA

Reason: While performing an area sweep, these records were encountered on the old database.

Action: Proceed with the reorganization and these records will be eliminated from the new database, or begin the job again, providing an OLD SUBSCHEMA that defines these records.

If the record is a member of the CALC set, *nnn* will be twice as large as the actual number of record occurrences on the database.

PARM001I STATEMENT IGNORED WHEN RPT OR RESTART SPECIFIED ON THE PROCESS STATEMENT

Reason: PROCESS statement keywords RPT or RESTART cause all following statements to be ignored.

Action: None.

PARM002W RESTART KEYWORD IGNORED WHEN RPT KEYWORD IS SPECIFIED

Reason: The RESTART parameter is being used with the RPT parameter in the same run.

Action: Initiate another run of CA-IDMS/DB Reorg using either the RPT parameter or the RESTART parameter.

PARM003E ERROR DETECTED IN module-name ERROR CODE RETURNED IS error-code n

Reason: An irrecoverable error condition has been detected. When the module name is GSDBLDP, possible error codes are:

100--Unable to build DMCL

104--Unable to load subschema

108--Unable to obtain storage for PSUB

2xx--GSSCALL error xx=CALLRC. See NKWP051E.

Action: Correct the situation or contact Computer Associates Product Support.

PARM004E ONLY ONE PROCESS STATEMENT CAN BE SPECIFIED

Reason: More than one PROCESS statement was found in file SYSIPT. Only one PROCESS statement is allowed.

Action: Correct and resubmit the job.

PARM005E STATEMENT FLUSHED DUE TO INVALID OR MISSING PROCESS STATEMENT

Reason: CA-IDMS/DB Reorg did not find a valid PROCESS parameter statement in file SYSIPT.

Action: A valid PROCESS statement is required and must be the first parameter statement in file SYSIPT. Correct and resubmit the job.

PARM006E NO VALID COPY STATEMENTS ENCOUNTERED - AT LEAST ONE COPY STATEMENT MUST BE SPECIFIED

Reason: CA-IDMS/DB Reorg did not find a valid COPY parameter statement in file SYSIPT.

Action: At least one valid COPY parameter statement must be given in file SYSIPT. Correct and resubmit the job.

PARM008E MEMBER LOADED FOR new/old SUBSCHEMA IS NOT A SUBSCHEMA

Reason: Member loaded is not a subschema. If subschema name is correct, then the subschema has been corrupted in the load or core-image library.

Action: Verify subschema name. If subschema has been corrupted, then recreate it and resubmit the job.

PARM010E MEMBER LOADED FOR DMCL OF new/old SUBSCHEMA IS NOT A DMCL

Reason: Member loaded for DMCL of subschema is invalid. If DMCL name is correct, the DMCL has been corrupted in the load or core-image library.

Action: Verify subschema name. If DMCL has been corrupted, recreate it and resubmit the job.

PARM011E NOT ENOUGH STORAGE TO BUILD INTERNAL PROCESSING TABLE

Reason: There is insufficient storage for CA-IDMS/DB Reorg to execute.

Action: Under VSE/ESA, use a larger partition or a smaller SIZE= parameter. Under OS/390, use a larger REGION. Resubmit the job.

PARM012E SUBSCHEMA subschema-name-1 FROM PARAMETER INPUT DOES NOT MATCH SUBSCHEMA subschema-name-2 FROM LAST RUN

Reason: The subschema name from the parameter input file and the control file of a previous run must be the same. When this error occurs, it is likely that the wrong parameter input or control file from a previous run is being used.

Action: Verify that parameter input and control files are correct. Verify that the load module core image library name and the CSECT name of the subschema are equal. Resubmit the job.

PARM013E ALL DB/REORG STEPS HAVE SUCCESSFULLY EXECUTED

Reason: An attempt was made to restart CA-IDMS/DB Reorg after it finished executing.

Action: None.

PARM014E NAME SPECIFIED FOR RPT/RESTART IS NOT A VALID STEP NAME

Reason: The step name specified in not a valid CA-IDMS/DB Reorg step.

Action: Correct the step name and resubmit the job.

PARM015E ALL STEPS PRIOR TO RPT/RESTART STEP HAVE NOT SUCCESSFULLY EXECUTED

Reason: CA-IDMS/DB Reorg has not completed all steps prior to the one specified by the RESTART parameter.

Action: Specify a step for RESTART that has completed processing in a prior run or the step that immediately follows the last step processed in a prior run.

PARM016E NOSWEEP AREA area-name MISSING FROM new/old SUBSCHEMA

Reason: The specified NOSWEEP area only exists within one of the subschemas.

Action: NOSWEEP can only be specified for areas that exist in both the old and new subschemas. Correct and resubmit the job.

PARM017E NOSWEEP AREA area-name HAS NO SET LINKAGE WITH ANY COPY AREAS

Reason: The specified NOSWEEP area does not have set linkage with any COPY areas.

Action: A NOSWEEP parameter statement is not necessary for an area that does not have set linkage to at least one COPY area. Correct and resubmit the job.

PARM018E set-name SET NOT IN new/old SUBSCHEMA**Reason:**

1. The specified set exists within only one of the subschemas.
2. The specified set exists within both of the subschemas, but the record ID or record names of the participating records are not the same in both subschemas.

Action: The specified set must exist with the same participating records in both old and new subschemas. Correct and resubmit the job.

PARM019E pointer-type DBKEY DISPLACEMENTS DO NOT MATCH BETWEEN SUBSCHEMAS FOR owner-or-member RECORD record-name IN SET set-name

Reason: Specified db-key displacements for a given record type did not match between the old and new subschemas.

Action: Db-key displacements for member and owner records must match between the old and new subschemas. Correct and resubmit the job.

PARM020E MEMBER member-name OF SET set-name NOT IN new/old SUBSCHEMA

Reason: Member of the set designated does not exist in both the old and new subschemas.

Action: Member of the set must exist in both the old and new subschemas. Correct and resubmit the job.

PARM021E new/old SUBSCHEMA record-name RECORD NOT IN old/new SUBSCHEMA

Reason: The record exists in only one subschema.

Action: Verify the subschemas. Correct and resubmit the job.

PARM022E AREA area-name NEEDS TO BE A COPY AREA FOR RECORD record-name IN new/old SUBSCHEMA

Reason: The record indicated does not reside in the same area in both the old and new subschemas. Both areas must be copied to move the record to a different area.

Action: Supply a copy statement for the additional area indicated in the message and resubmit the job.

PARM023E STOPAFTER STEP CANNOT PRECEDE RESTART STEP

Reason: The step specified in the STOPAFTER parameter is processed before the step specified in the RESTART parameter.

Action: Change the incorrect STOPAFTER or RESTART value. Resubmit the job.

PARM027E NO record-name RECORDS FOR INDEX SET index-set-name IN THE DATABASE

Reason: Record-name records for index set are missing.

Action: Correct this condition and resubmit the job.

PARM033E AREA NAME NOT IN new/old SUBSCHEMA

Reason: An area was specified in a COPY or NOSWEEP statement that does not exist in both the old and the new subschemas.

Action: Correct this condition and resubmit the job.

PARM034E AREA NAME PREVIOUSLY SPECIFIED IN A COPY STATEMENT

Reason: The same area cannot be specified in two copy statements.

Action: Correct the input parameters and resubmit the job.

PARM035E MULTIPLE OCCURRENCES OF SAME PARAMETER STATEMENT

Reason: For NOSWEEP or COPY, this parameter statement has already been given for this area.

Action: Remove the conflicting NOSWEEP or COPY parameter statement. Resubmit the job.

PARM039E RECORD record-name NOT IN AREA area-name IN NEW SUBSCHEMA

Reason: The COPY statement PRIORITY parameter specifies a record that does not reside in the area named in the copy statement area parameter for the subschema named.

Action: Correct the record name. Resubmit the job.

PARM041E AREA NAME PREVIOUSLY SPECIFIED IN A NOSWEEP STATEMENT

Reason: A COPY statement and a NOSWEEP statement specify the same area.

Action: Remove the conflicting NOSWEEP or COPY parameter statement. Resubmit the job.

PARM042E PAGE RESERVE VALUE GREATER THAN MAXIMUM FREE SPACE ON A PAGE

Reason: The COPY statement PAGERESERVE parameter specifies a value that is too large.

Action: Correct the PAGERESERVE value. Resubmit the job.

PARM043I NO PARAMETER ERRORS DETECTED

Reason: The parameter statement just processed did not contain any errors.

Action: None.

PARM044I ERROR(S) DETECTED IN PARAMETER STATEMENT

Reason: The parameter statement just processed contained one or more errors.

Action:

None.

**PARM046E ERROR FOUND BY STEP INITIALIZATION ROUTINE -
USRSUBS**

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

PARM047E PARAMETER INPUT FILE CONTAINS NO STATEMENTS

Reason: File SYSIPT was empty.

Action: Place parameter statements in file SYSIPT. Resubmit the job.

**PARM050E INDEX SET *index-set-name* RESIDES IN COPY AREA IN *new/old*
SUBSCHEMA BUT NOT IN *old/new* SUBSCHEMA**

Reason: The named IIX index set resides within a COPY area for only one of the subschemas.

Action: Give a COPY parameter statement to resolve the error condition. Resubmit the job.

**PARM051E LOCATION MODE CHANGE FOR *record-name* RECORD. AREA
area-name AND AREA *area-name* MUST BE COPY AREAS**

Reason: The named record had its location mode changed. The named areas must be copy areas.

Action: Give two COPY parameter statements to resolve the error condition. Resubmit the job.

**PARM052E LOCATION MODE CHANGE FOR *record-name* RECORD. AREA
area-name MUST BE A COPY AREA**

Reason: The named record had its location mode changed. The named area must be a copy area.

Action: Give a COPY parameter statement to resolve the error condition. Resubmit the job.

PARM053E FORMAT AND/OR LENGTH OF record-name CHANGED

Reason: The record indicated had its format and/or length changed from the old subschema to new subschema. Additionally, for a variable length record, the control length may have changed. This type of change is a restructure and is not supported.

Action: Correct this condition and resubmit the job.

PARM057E MEMBER LOADED FOR new/old SUBSCHEMA CONTAINS INTEGRITY ERRORS

Reason: The subschema specified in the PROCESS statement has a format or content error.

Action: Correct the subschema or verify its name in the PROCESS statement and resubmit the job.

PARM058E MEMBER LOADED FOR new/old SUBSCHEMA CONTAINS ONE OR MORE NATIVE VSAM FILES

Reason: The current release of CA-IDMS/DB Reorg does not support the processing of native VSAM files.

Action: Create a subschema (NEW or OLD) which does not include native VSAM files and resubmit the job using the changed subschema.

PARM059E PRIVACY LOCK ON FOR level-name level-identifier IN new/old SUBSCHEMA

Reason: A requested reorganization cannot be executed because a privacy lock is active for the named area, set, or record.

Action: Create a subschema without a privacy lock and make it available to CA-IDMS/DB Reorg. Resubmit the job.

PARM062E RECORD record-name PARTICIPATES IN SETS NOT DEFINED BY THE NEW SUBSCHEMA

Reason: The indicated record is a member or an owner of sets that are not defined in the new subschema.

Action: Define a subschema that includes all sets in which the records participate. After making the necessary changes, resubmit the job.

PARM064E INFORMATION/REPLY ROUTE CODE PARAMETER DOES NOT APPLY WHEN USER SPECIFIES (NO) ON DSPLYINT PARAMETER

Reason: Both DSPLYINT=NO and the INFORROUTE or REPLYROUTE parameter were specified.

Action: Specify another option for DSPLYINT or delete the INFORROUTE and/or REPLYROUTE.

PARM065E INFORMATION/REPLY ROUTE CODE CLASS 11 IS INFORMATIONAL ONLY--USER MUST SPECIFY ROUTE CODE THAT ALLOWS REPLIES

Reason: A class 11 route code was specified. Class 11 is not acceptable because it only allows display of information and does not allow the operator to reply.

Action: Specify a route code that allows the operator to reply.

PARM066E RECORD record-name IN NEW SUBSCHEMA IS VARIABLE WITHOUT AN OCCURS DEPENDING ON CLAUSE

Reason: The indicated record is defined with a MINIMUM FRAGMENT clause, but does not have an OCCURS DEPENDING ON clause.

Action: Remove the FRAGMENT clause from the new subschema or add an OCCURS DEPENDING ON clause.

PARM067E INDEX SET index-set-name DEFINED IN old/new SUBSCHEMA BUT NOT IN new/old SUBSCHEMA

Reason: The old and new subschemas do not correspond.

Action: Define the index set in either the old or the new subschema as required.

PARM068E MEMBER record-name IS IN OLD SUBSCHEMA AND MEMBER record-name IS IN NEW SUBSCHEMA FOR INDEX SET index-set-name

Reason: The member record is inconsistent between the old and new subschema.

Action: Change the member record in either the old or the new subschema to match the other.

PARM069E NAME OF INDEX SET index-set-name IN OLD SUBSCHEMA DIFFERS FROM INDEX SET NAME index-set-name IN NEW SUBSCHEMA

Reason: Index set names are inconsistent between the old and new subschema.

Action: Change the index set name in either the old or the new subschema to match the other.

PARM070E area-name AREA CANNOT BE NOSWEEP SINCE OWNER OF SET set-name IN COPY AREA

Reason: The indicated area was designated NOSWEEP and it contains records that are linked (as members) to an SR8 set in a copy area.

Action: Remove the NOSWEEP statement for the indicated area. Resubmit the job.

PARM071E IIX SET index-set-name CONTAINS HI-LO PAGE RANGE NOT CORRESPONDING TO RANGE OF ANY AREA

Reason: The OR52 subschema control block, for the indicated index set, points to an AM57 subschema control block whose page ranges do not correspond to any AC56 page ranges in the subschema.

Action:

Contact Computer Associates Product Support. To verify the page ranges, you can browse the subschema load module to find the OR52 control block for the indicated set. The last fullword in the OR52 indicates the offset to the AM57. The HI-LO page ranges are in the fifth and sixth fullwords of the AM57. Compare these page ranges to the page ranges in the AC56 control blocks.

PARM072E IIX SET index-set-name IS IN A COPY AREA IN THE new/old SUBSCHEMA BUT IS NOT IN A COPY AREA IN THE new/old SUBSCHEMA

Reason: The integrated index set owner was found in a copy area in one subschema, but not in the other subschema.

Action: Add another COPY statement that includes the area containing the integrated index in the other subschema.

PARM073E OPERATOR COMMUNICATIONS REQUESTED--NOT PERMITTED IN A VM/ESA ENVIRONMENT

Reason: Unsupported function.

Action: The DSPLYINT parameter is not functional in a VM/ESA environment.

PARM074E record-name, MEMBER OF UNDEFINED SET(S) IN old/new SUBSCHEMA

Reason: The SET(S) this record participates in is not defined in the noted subschema.

Action: Add the SETS(S) to the appropriate subschema.

PARM075E INVALID USER WORK AREA SIZE PASSED

Reason: Invalid user work area size passed in call to GSDBLDP.

Action: Size must be from 64 to 256 inclusive

PARM076E SUBSCHEMA AREA NOT FOUND IN THE DMCL

Reason: An area was specified in the subschema which was not found in the DMCL.

Action: Use a DMCL that contains all areas in the subschema.

PARM077E SUBSCHEMA subschema-name DEFINES AN SQL DATABASE - NOT SUPPORTED

Reason: CA-IDMS/DB Reorg does not currently support SQL databases.

Action: Do not run a CA-IDMS/DB Reorg specifying

PARM078E BLOCKNUM nnn MUST BE IN RANGE 1:32

Reason: The BLOCKNUM parameter which specifies the blocking factor for the CNTRL1, CNTRL2, and PAGUTIL work files must be in the range from 1 to 32.

Action: Correct the BLOCKNUM value and re-run the job.

PARM079E BLOCKNUM nnn FROM PARAMETER INPUT DOES NOT MATCH BLOCKNUM yyy FROM LAST RUN.

Reason: The BLOCKNUM parameter specified in a RESTART run must match that from the previous run.

Action: Correct the BLOCKNUM value and re-run the job.

PARM080W FSBUFFERS KEYWORD WAS BYPASSED - REFER TO SYSIDMS PREFETCH CAPABILITY.

Reason: The FSBUFFER parameter was included on the PROCESS statement. This is no longer valid. It has been replaced by the SYSIDMS prefetch capability.

Action: None- the parameter is ignored.

PRIA001E ERROR FOUND BY STEP INITIALIZATION ROUTINE - USRSUBS

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

PRIA002I PRIOPTA RECORDS READ..... number-of-records

Reason: The number of PRIOPTA records that have been read.

Action: None.

PRIA003I PRIREAL RECORDS WRITTEN number-of-records

Reason: The number of PRIREAL records written.

Action: None.

PRIA004I UNALLOC RECORDS WRITTEN number-of-records

Reason: The number of UNALLOC records written.

Action: None.

PRIA005I PAGUTIL RECORDS WRITTEN number-of-records

Reason: The number of PAGUTIL records written.

Action: None.

PRIA006I PAGUTIL RECORDS UPDATEDnumber-of-records

Reason: The number of PAGUTIL records updated.

Action: None.

PRIA012E MISSING PRIOPTA RECORDS

Reason: The PRIOPTA file was not created by CA-IDMS/DB Reorg or it is a legitimate file that has been corrupted.

Action: Verify that the PRIOPTA file is from the present run. Resubmit the job.

PRIA013E RECORD-ID record-id NOT IN new/old SUBSCHEMA

Reason: The record-ID does not exist in the subschema designated.

Action: Correct the subschema or verify its name in the PROCESS statement and resubmit the job.

PRIA014E RECORD-ID record-id IS CALC RECORD MISSING ITS CALC SET

Reason: PRIOPTA contains the record defined as CALC but the program is unable to locate its CALC set.

Action: Correct the subschema or verify its name in the PROCESS statement and resubmit the job.

PRIA015E RECORD-ID record-id IS A CALC RECORD WITH DUPLICATE SYMBOLIC KEYS BUT THE NEW SUBSCHEMA PROHIBITS DUPLICATES

Reason: The location mode of the record is being changed from VIA to CALC with a DUPLICATES NOT ALLOWED option, but duplicate symbolic keys exist in the old database.

Action: Eliminate duplicates for the record or redefine the record with DUPLICATES ALLOWED option in the new subschema and resubmit the job. CA-IDMS/DB Reorg cannot delete records from the database.

PRIA016E SYSTEM OWNED IIX SET index-set-name DOES NOT CORRESPOND TO ANY SET DEFINITION

Reason: An SR7 was obtained whose set name (or CALC key) does not match any set name in the OR52 subschema control blocks in the subschema.

Action: Contact Computer Associates Product Support.

PRIA017E IIX SET index-set-name CONTAINS HI-LOW PAGE RANGE NOT CORRESPONDING TO RANGE OF ANY AREA

Reason: The OR52 subschema control block, for the indicated index set, points to an AM57 subschema control block whose page ranges do not correspond to any AC56 page ranges in the subschema.

Action: Contact Computer Associates Product Support. To verify the page ranges, you can browse the subschema load module to find the OR52 control block for the indicated set. The last fullword in the OR52 indicates the offset to the AM57. The HI-LO page ranges are in the fifth and sixth fullwords of the AM57. Compare these page ranges to the page ranges in the AC56 control blocks.

PRNT001E ERROR IN GENERAL SERVICE PRINT PROGRAM

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

PRST001I RESTART DATA.....old/new SUBSCHEMA subschema-name

Reason: CA-IDMS/DB Reorg subschema and step status information for the RESTART parameter.

Action: None.

PRNT910E A SEVERE ERROR OCCURRED DURING OUTPUT TO THE SYSLST FILE

Reason: GSSCALL experienced a problem invoking GSSPRNT.

Action: Ensure all necessary programs are available in your steplib or LIBDEF.

PRST002I RESTART WITHstep-name

Reason: The CA-IDMS/DB Reorg step where you should continue processing after interrupting CA-IDMS/DB Reorg.

Action: None.

PRST003I DATASET DISPOSITION FOR RESTARTfile-name disposition step-name

Reason: CA-IDMS/DB Reorg file status information for RESTART parameter.

Action: None.

RCTL001E INVALID RECORD ID. IT SHOULD READ record-id-1 BUT INSTEAD IT READ record-id-2

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

RCTL002E INADEQUATE STORAGE AVAILABLE FOR REQUEST OF n

Reason: There is insufficient storage for CA-IDMS/DB Reorg to execute. The amount of storage requested was n bytes.

Action: Under VSE/ESA, use a larger partition or smaller SIZE= parameter. Under OS/390, use a larger REGION. Resubmit the job.

RCTL003E NBR OF BYTES OF PSUB LOADED BY USRRCTL EXCEEDS PSUB LENGTH VALUE CONTAINED IN CONTROL AREA

Reason: The integrity of CNTRL1/CNTRL2 files has been compromised.

Action: Contact Computer Associates Product Support.

RCTL004E NBR OF BYTES OF PSUB LOADED BY USRRCTL IS LESS THAN PSUB LENGTH VALUE CONTAINED IN CONTROL AREA

Reason: The integrity of CNTRL1/CNTRL2 files has been compromised.

Action: Contact Computer Associates Product Support.

RCTL005I SINCE ERROR DETECTED IN FILE file-name-1, USRRCTL WILL TRY TO READ FILE file-name-2

Reason: Control file-1 could not be read successfully, so backup control file-2 will be tried.

Action: None.

RCTL006I ERROR DETECTED IN FILE file-name AND THERE ARE NO MORE FILES TO READ

Reason: Control file designated could not be read successfully.

Action: None.

RCTL007I USRRCTL SUCCESSFULLY COMPLETED USING FILE file-name

Reason: Control file designated was successfully processed by USRRCTL.

Action: None.

RORG001E ERROR DETECTED IN module-name ERROR CODE RETURNED IS n

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

RORG002E ERROR LINKING TO module-name, LINK PARAMETER ERROR

Reason: USREORG is unable to call this module due to internal error.

Action: Contact Computer Associates Product Support.

RORG003E ERROR LINKING TO program-name, PROGRAM NOT FOUND

Reason: Module cannot be located in the available load or core image libraries.

Action: Locate the load or core image library that contains the module and make it available to the job. Resubmit the job.

RORG004E ERROR LINKING TO module-name, NOT ENOUGH STORAGE TO LOAD PROGRAM

Reason: The module cannot be loaded due to insufficient GETVIS area (VSE/ESA only).

Action: Resubmit the job with a smaller SIZE= parameter. A larger partition may also be required.

RORG005E ERROR IN SORT PROGRAM module-name INVALID SORT FIELD PARAMETER

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

RORG006E ERROR IN SORT PROGRAM module-name INVALID SORT RECORD PARAMETER

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

RORG007E ERROR IN SORT PROGRAM module-name E35 PROGRAM NOT FOUND

Reason: The module cannot be located in the available load or core image libraries.

Action: Locate the load or core image library that contains the module and make it available to the job. Resubmit the job.

RORG008E ERROR IN SORT PROGRAM module-name NOT ENOUGH STORAGE TO LOAD E35 PROGRAM

Reason: Module cannot be loaded due to insufficient GETVIS area (VSE/ESA only).

Action: Resubmit the job with a smaller SIZE= parameter. A larger partition may also be required.

RORG009E ERROR IN SORT PROGRAM module-name SORT UTILITY ERROR

Reason: The call to the installation-provided sort program failed with a non-zero return code.

Action: See your installation's sort program documentation for an explanation of the return code and any other sort program messages. Correct and resubmit the job.

RORG010E ERROR IN SORT PROGRAM module-name ERROR PROCESSING INPUT FILE

Reason: GSSFILE failed while processing the input file. This error could also be caused by incorrect JCL.

Action: Verify that the JCL is appropriate and resubmit the job. If no JCL or other user errors can be found, report the error to Computer Associates Product Support.

RORG011E ERROR IN SORT PROGRAM module-name REPORTED IN PREVIOUS MESSAGE

Reason: Module detected an error. The error was reported in the preceding message.

Action: Verify that the JCL is appropriate and resubmit the job. If no JCL or other user errors can be found, report the error to Computer Associates Product Support.

RORG012I STEP INTENTIONALLY BYPASSED

Reason: The current step was bypassed because it was not needed.

Action: None.

RORG013I DATABASE REORGANIZATION COMPLETE

Reason: The database reorganization process is complete.

Action: None.

RORG014E ERROR OCCURRED ON CALL TO RESET UTILITY USRSUBS

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

SECA001E ERROR FOUND BY STEP INITIALIZATION ROUTINE - USRSUBS

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

SECA002I SECOPTA RECORDS READ number-of-records

Reason: The sum of the SECOPTA and UNALLOC records read.

Action: None.

SECA003I SECREAL RECORDS WRITTEN number-of-records

Reason: The number of SECREAL records written.

Action: None.

SECA004I PAGUTIL RECORDS READ number-of-records

Reason: The number of PAGUTIL records read.

Action: None.

SECA005I PAGUTIL RECORDS WRITTEN number-of-records

Reason: The number of PAGUTIL records written.

Action: None.

SECA006I PAGUTIL RECORDS UPDATED number-of-records

Reason: The number of PAGUTIL records updated.

Action: None.

SECA012E MISSING SECONDARY ALLOCATION RECORDS

Reason: Secondary allocation file was not created by CA-IDMS/DB Reorg, or it is a legitimate file that has been corrupted.

Action: Verify that the secondary allocation file is valid. The secondary allocation file consists of the SECOPTA and UNALLOC files. Resubmit the job.

SECA013E RECORD-ID record-id NOT IN new/old SUBSCHEMA

Reason: The record-ID is not defined in the subschema designated. Probable cause is a CNTRL1/CNTRL2 file that has been corrupted.

Action: Correct this condition and resubmit the job.

SECA014E NO SPACE LEFT IN AREA area-name FOR MINIMUM ROOT OF RECORD record-name

Reason: The area is full.

Action: Correct this condition by creating adequate space in the new subschema for the area indicated, by decreasing the page reserve parameter value, by redistributing records to other areas, or by using a combination of these techniques. Resubmit the job.

SECA015E NO SPACE LEFT IN AREA area-name FOR RECORD record-name, WHICH CANNOT BE FRAGMENTED.

Reason: The area is full.

Action: Correct this condition by creating adequate space in the new subschema for the area indicated by decreasing the page reserve parameter value, by redistributing records to other areas, or by using a combination of these techniques. Resubmit the job.

SECA016E AREA area-name IS FULL

Reason: This area is full.

Action: Correct this condition by creating adequate space in the new subschema for the area indicated, by decreasing the page reserve parameter value, by redistributing records to other areas, or by using a combination of these techniques. Resubmit the job.

SECA017E NO SET RECORD FOUND WHOSE ID MATCHES THE SET ID set-id ON INPUT RECORD. SAVE SNAP AND CALL Computer Associates International, Inc.

Reason: No set record was found with the an ID that matches the ID found in the system-owned integrated index set.

Action: Save the SNAP/dump and contact Computer Associates Product Support.

SECA018E NO AREA FOUND THAT CONTAINS IIX SET NUMBER set-id. SAVE ACCOMPANYING SNAP AND CALL Computer Associates International, Inc.

Reason: CA-IDMS/DB Reorg could not find an area containing the indicated index set.

Action: Save the SNAP/dump and contact Computer Associates Product Support.

SECA019E PROGRAM ATTEMPTING TO CREATE UPPER LEVEL IIX STRUCTURE WHEN NO BOTTOM LEVEL EXISTS

Reason: The program encountered a serious logic flaw and could not continue.

Action: Save the SNAP/dump and contact Computer Associates Product Support.

SETL001E ERROR OCCURRED ON CALL TO RESET UTILITY USRSUBS

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

SETL002I ALLOCX READ number-of-records

Reason: The number of ALLOCX records read.

Action: None.

SETL003I DBREC WRITTEN number-of-records

Reason: The number of DBREC records written.

Action: None.

SETL005I DATABASE RECORDS MODIFIED number-of-recor

Reason: The number of database records modified.

Action: None.

**SETL006I RECORD PREFIXES HAVE BEEN ALTERED IN NON-COPY
AREA area-name**

Reason: Records have been modified in the area indicated. The area was not specified in a copy statement.

Action: None.

**SNAP001E ERROR FOUND DURING SNAP DUMP - GSSSNAP RETURN
CODE IS nnnnnnnnnn**

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

IIXX001I IXSET READ number-of-records

Reason: The number of IXSET records read.

Action: None.

IIXX002I IXDET READ number-of-records

Reason: The number of IXDET records read.

Action: None.

IIXX003I IIXNDX2 WRITTEN number-of-records

Reason: The number of IIXNDX2 records written.

Action: None.

**IIXX004E WRONG VERSION OF USRIIXX MODULE, DB/REORG
release-number-1 USRIIXX release-number-2**

Reason: The release numbers of CA-IDMS/DB Reorg's main program (USREORG) and USRIIXX do not match. This error occurs if a new release of CA-IDMS/DB Reorg is partially installed over a prior release, or if two releases of CA-IDMS/DB Reorg are present and the available load or core image libraries cause mixed releases of the program modules to be used.

Action: Verify that CA-IDMS/DB Reorg is correctly installed and that the proper load or core-image libraries are used. Resubmit the job.

IIXX008E ERROR OCCURRED ON CALL TO RESET UTILITY USRSUBS

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

STAT001I CA-IDMS STATISTICS FOLLOW - - - - -

Reason: CA-IDMS statistics are reported following this message.

Action: None.

STAT002I step-name STATISTICS FOLLOW - - - - -

Reason: Statistics for the step are reported following the message.

Action: None.

STOR001E ERROR OCCURRED ON CALL TO RESET UTILITY USRSUBS

Reason: An irrecoverable error condition has been detected.

Action: Contact Computer Associates Product Support.

STOR002I DBREC READ number-of-records

Reason: The number of DBREC records read.

Action: None.

STOR003I DBREC STORED number-of-records

Reason: The number of database records stored.

Action: None.

**SYNC001E MODULE VERSIONS DO NOT MATCH. DB/REORG
release-number-1 module-name release-number-2**

Reason: The release numbers of CA-IDMS/DB Reorg's main program (USREORG) and module-name do not match. This error occurs if a new release of CA-IDMS/DB Reorg is partially installed over a prior release, or if two releases of CA-IDMS/DB Reorg are present and the available load or core-image libraries cause mixed releases of the program modules to be used.

Action: Verify that CA-IDMS/DB Reorg is correctly installed and that the proper load or core-image libraries are used. Resubmit the job.

SYNC002E FILE SYNC CONTROL RECORD MISSING FROM file-name FILE

Reason: The file designated was not created by CA-IDMS/DB Reorg or it is a legitimate file that has been corrupted.

Action: Verify that the file is valid. Resubmit the job.

**SYNC003E file-name FILE NOT FOR CURRENT RUN OF CA-IDMS/DB
REORG**

Reason: The file designated was created by CA-IDMS/DB Reorg but is not to be used with this run of CA-IDMS/DB Reorg.

Action: Verify that the file is valid. Resubmit the job.

**SYNC004E EXTRANEIOUS FILE SYNC CONTROL RECORD READ FROM
file-name FILE**

Reason: The file indicated was created by CA-IDMS/DB Reorg but appears to be corrupted with extraneous records.

Action: Verify that the file is valid. Resubmit the job.

**SYNC005E file-name FILE SHOULD CONTAIN n1 RECORDS. CURRENT
FILECOUNT IS n2 RECORDS**

Reason: The file is corrupt because of missing or extraneous records.

Action: Verify that the file is valid and resubmit the job.

UTIL001E INVALID PARAMETER LIST PASSED TO MODULE module-name

Reason: Module indicated has received an invalid parameter list.

Action: Contact Computer Associates Product Support.

**UTIL002E UNANTICIPATED ERROR OCCURRED IN PROGRAM module-
name RETURN CODE WAS nnnn**

Reason: Module ended with the indicated return code.

Action: Contact Computer Associates Product Support.

**UTIL901E program-name PROGRAM WAS UNABLE TO PRINT MESSAGE
message-number**

Reason: See the message following this one on the Audit Report.

Action: See the message following this one on the Audit Report.

**WCTL001E INVALID PSUB BLOCK ENCOUNTERED IN PROGRAM
program-name**

Reason: An unusual processing situation has occurred and the utility cannot continue.

Action: Contact Computer Associates Product Support.

WTOR000I =====> REPLY WITH REQUEST TO DB/REORG *****

Reason: The Console Communication Facility has been activated. CA-IDMS/DB Reorg is ready to accept console requests.

Action: Enter a console communication command if desired.

WTOR001E ERROR - BAD SYNTAX, NO COMMAND GIVEN WITH REPLY

Reason: A reply to WTOR000I was given but it did not contain a command.

Action: Enter a console communication command.

WTOR002E ERROR - BAD SYNTAX, INTERVAL REQUEST GREATER THAN MAXIMUM(120)

Reason: The value in the SET INTERVAL request is too large.

Action: Enter a SET INTERVAL request in the range 0-120.

WTOR003E ERROR - BAD SYNTAX, INTERVAL REQUEST LESS THAN MINIMUM (0)

Reason: The value in the SET INTERVAL request is too small.

Action: Enter a SET INTERVAL request in the range 0-120.

WTOR004E ERROR - BAD SYNTAX, INVALID COMMAND/OPERAND IN REPLY

Reason: A reply to WTOR000I was given but it did not contain a valid command/operand.

Action: Enter a console communication command with a valid command/operand.

WTOR005I INTERVAL AND STEP DISPLAYS CANCELED, DISPLAY IS ONREQUEST ONLY

Reason: A SET INTERVAL REQUEST command was successfully processed.

Action: None.

WTOR006E ERROR - UNABLE TO ATTACH A DISPLAY INTERVAL RC = hex-word

Reason: The display interval could not be established by USRINTV. An ABEND or internal error may be the cause. In VSE/ESA, lack of timer support in the partition used by CA-IDMS/DB Reorg could also cause the error.

Action: If no user error can be found, contact Computer Associates Product Support.

WTOR007I INTERVAL REQUEST ACCEPTED, CURRENT DISPLAY INTERVAL IS NOW nnn MINUTES

Reason: A SET INTERVAL command was successfully processed.

Action: None.

WTOR008I STEP REQUEST ACCEPTED, END OF STEP DISPLAY IS NOW ACTIVE

Reason: A SET INTERVAL STEP command was successfully processed.

Action: None.

WTOR009E DISPLAY TIMER IS NOT ACTIVE

Reason: A SET INTERVAL STEP command was issued and the interval timer was either turned off by a previous SET INTERVAL console command or it was established through the DISPLAY parameter or a SET INTERVAL command.

Action: Establish a timer display interval with the SET INTERVAL console command. (VSE/ESA installations only: If this message appears in the Audit Report at the end of the job, the interval timer ended before the interval established in the SET INTERVAL command. Contact Computer Associates Product Support.)

WTOR010E subtask-name HAS ABENDED, SUBTASK ECB=hex-word

Reason: The subtask designated has ended unexpectedly and the timer display cannot be reestablished once it has abended.

Action: Contact Computer Associates Product Support.

WTOR011I DISPLAY INTERVAL HAS BEEN CANCELLED, DISPLAY IS ONLY AT END OF STEP

Reason: This message is displayed in response to a SET INTERVAL operator command.

Action: None.

Two types of errors can be reported by the return codes of n1, n2, n3, and n4--non-VSAM file errors and VSAM file errors. The error is described by n2 and n4. For VSAM file errors, n4 is always equal to 28. The error is described by n1, n2, and n3. A general return code is given by n4 for both non-VSAM and VSAM errors. All return codes are decimal values.

Table 6-1 (Page 1 of 2). n4 Return Codes

n4	Reason	Action
4	End-of-file	Call Product Support.
8	Open error or file is not open	Look for JCL errors or for the use of improper files.
12	An I/O error has occurred	Find cause for I/O error.
16	Request not recognized	Call Product Support.
20	File was already opened	Call Product Support.
24	Parameter list error	Call Product Support.

Table 6-1 (Page 2 of 2). n4 Return Codes

n4	Reason	Action
28	VSAM error n1=R15 return code from VSAM n2=low order byte from R0 GENCB/MODCB type of error n3=VSAM feedback byte error in I/O request	Use n1, n2, and n3 to check for possible user errors. If there are no user errors, call Product Support.
32	Insufficient storage	Increase storage for job step.
36	SYNAD error occurred	Call Product Support.
40	BPAM FIND error n1=R15 n2=R0	Use n1 and n2 (as described in Data Management Macro Instructions) to check for errors.
44	BPAM STOW error n1=R15 n2=0	Use n1 and n2 (as described in Data Management Macro Instructions) to check for errors.

Table 6-2 (Page 1 of 2). n2 Return Codes

n2	Reason	Action
0	n4=8, use of unopened file n4=24, parameter list error	Call Product Support. Call Product Support.
1	JCL/label override parm list	Remove DCB information from JCL and ensure that the correct files are referenced.
2	Parm list override JCL/label	Remove DCB information from JCL and ensure that the correct files are referenced.
3	Unrecognized request	Call Product Support.
4	OS/390 x13 ABEND trapped at open	Fix cause for x13 ABEND.
5	Tried to updat seq. file	Call Product Support.
6	VSAM write at other than load	Call Product Support.

Table 6-2 (Page 2 of 2). n2 Return Codes

n2	Reason	Action
7	SOS table could not expand	Call Product Support.
8	OS/390 DCB open failed	Call Product Support.
9	SOS table buffer pointer lost	Call Product Support.
10	SOS table file CB not built	Call Product Support.
11	OS/390 DD statement Missing	Supply missing DD statement.
12	VSAM ACB open failed	Call Product Support.
13	Record format invalid	Call Product Support.
14	Macro format invalid	Call Product Support.
15	Record length not numeric	Call Product Support.
16	Record length too large	Call Product Support.
17	Block size not numeric	Call Product Support.
18	Block size too large	Call Product Support.
19	Invalid VSE/ESA sysname table	Assemble a valid sysname table.
20	VSE/ESA sysname table entry missing	Assemble a sysname table with an entry for the missing one.
21	VSE/ESA LU number too large	Use an LU number within range.
22	VSE/ESA sysname is not numeric or is misspelled	Correct to a valid sysname.
23	VSE/ESA sysname blank	Do not use blank sysname.
24	VSE/ESA LU not assigned	Call Product Support.
25	VSE/ESA DTF prototype missing	Call Product Support.
26	VSE/ESA logic module missing	Generate missing logic module.
27	VSE/ESA CCW mismatch	Call Product Support.
28	File is not a PDS	Allocate file to a PDS.

Index

A

Audit Report 4-5, 6-5

C

CA-IDMS/DB Reorg Output 4-4
CA-IDMS/DB Reorg Processing--OS/390
 Environment 5-26
CA-IDMS/DB Reorg Processing--VM/ESA
 Environment 5-43
CA-IDMS/DB Reorg Processing--VSE/ESA
 Environment 5-34
CA-IDMS/DB Reorg Reorganization Capabilities 3-14
CA-IDMS/DB Reorg's Inputs 3-5
CA-IDMS/DB Reorg's Output 1-12
CA-IDMS/DB Reorg's Record Allocation
 Techniques 3-7
Condition Codes 6-4
Console Communications Facility 2-25

D

Database Access 5-4
DBKEYS Interface File 4-13

H

How CA-IDMS/DB Reorg Performs Reorganization 1-8

I

Improve System and CA-IDMS/DB Performance 1-7

M

Message Text 6-7

N

NOSWEEP Statement 2-23

O

Operational Flexibility During Reorganization 1-11
Operational Flow and Considerations 5-5
Organization ix
Overview 2-3, 3-3, 4-3, 5-3

P

Parameters and Their Functions 2-4
PROCESS Statement 2-6
Publications x
Purpose viii

R

Reorganization Steps 3-21

S

Select Any Area or Entire Database 1-5
Status Display 4-12

U

Understanding CA-IDMS/DB Reorg Processing 3-4
Understanding syntax diagrams xi

V

VSE/ESA File Processing Alternate Method 6-6

W

Why Reorganize a Database? 1-4

