

CA IDMS™/DB Reorg

User Guide
Release 18.5.00, 2nd Edition



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

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

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

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

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

The manufacturer of this Documentation is CA.

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

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

CA Technologies Product References

This document references the following CA Technologies products:

- CA IDMS™/DB Database Reorg Option (CA IDMS Reorg)
- CA OLQ™ Online Query for CA IDMS™ (CA OLQ)
- CA Culprit™ for CA IDMS™ (CA Culprit)
- CA IDMS/DC™ Transaction Server Option (CA IDMS/DC)
- CA IDMS™ Database Universal Communications Facility Option (CA IDMS UCF)
- CA IDMS™ DC Transaction Server Option or CA IDMS™ Database Universal Communications Facility Option (DC/UCF)
- CA ADS™ for CA IDMS™ (CA ADS)

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

Documentation Changes

The following documentation updates were made for the 18.5.00, 2nd Edition release of this documentation:

- [Allocating Work Files](#) (see page 110)—Added DBKEYS to the Work Files and Corresponding Logical Units table.

Contents

Chapter 1: Introduction	9
Understanding Syntax Diagrams	9
Sample Syntax Diagram.....	11
Chapter 2: General Information	13
Why Reorganize a Database?.....	13
Select Any Area or Entire Database	13
Updates Set Linkage Automatically	14
Leaves Relationships and Records Unaltered	14
Recognizes SR8 Index Areas	14
Improve System and CA IDMS/DB Performance	15
How CA IDMS/DB Reorg Performs Reorganization.....	15
Reorganization Controlled by Parameter	16
Reorganization Controlled by Subschema and DMCL	17
Reorganization Controlled by Internal Logic.....	18
Operational Flexibility During Reorganization.....	18
STOPAFTER and RESTART Parameters.....	18
Console Communication Facility	19
CA IDMS/DB Reorg's Output	19
Audit Report Designed for Operations Flexibility.....	19
CA IDMS/DB Reorg Interface Capabilities.....	20
Chapter 3: Parameters and Commands	21
Parameters and Their Functions.....	21
PROCESS Statement Overview.....	21
COPY Statement Overview	21
NOSWEEP Statement Overview	22
PROCESS Statement.....	23
PROCESS Parameter.....	24
OLDSUB Parameter	25
NEWSUB Parameter.....	25
OLDDMCL Parameter.....	25
NEWDMCL Parameter	25
OLDDBN Parameter	26
NEWDBN Parameter	26
RPT Parameter	26

RESTART Parameter	28
STOPAFTER Parameter	30
DSPLYINT Parameter	32
TSOUSERS Parameter	35
SORTFLAG Parameter	35
SORTSIZE Parameter	35
BLOCKNUM Parameter	36
COPY Statement	36
Copy Parameter	36
AREA Parameter	37
PRIORITY Parameter	37
PAGERESERVE Parameter	39
TARGETROOT Parameter	39
NOSWEEP Statement	40
Console Communications Facility	41
DISPLAY STATUS Command	41
SET INTERVAL Command	41
STOP Command	42
CANCEL Command	42

Chapter 4: Concepts	43
Understanding CA IDMS/DB Reorg Processing	43
Reorganization--Not Restructuring	43
How Is CA IDMS/DB Reorg Different From Other Utilities?	43
CA IDMS/DB Reorg's Inputs	44
CA IDMS/DB Reorg's Record Allocation Techniques	45
Minimize Record Overflow	45
Reduce Fragmented Records	46
Allocating DIRECT Records	46
Allocating CALC Records	46
Allocating VIA Records	47
Sorted VIA Records	48
Allocation and the PRIORITY Parameter	49
Allocation and Page Reserve	50
CA IDMS/DB Reorg Reorganization Capabilities	51
Reorganization Capabilities Controlled by Parameters	52
Direct an Efficient Reorganization Through Parameters	53
Reorganization Capabilities Controlled by Subschema and DMCL Definition	54
Automatic Reorganization Controlled by Internal Logic	56
Reorganization Steps	57
Functions of the Reorganization Steps	59

Chapter 5: System Output 63

CA IDMS/DB Reorg Output	63
Audit Report	63
Status Display.....	64
Interface File	64
Audit Report	64
Audit Subreport--Area Recap for Old Subschema.....	66
Audit Subreport--Set Linkage Recap for Area	67
Restart Information Messages.....	70
Status Display.....	73
Status Display Fields	73
DBKEYS Interface File.....	73

Chapter 6: Operations 75

Database Access.....	75
Operational Flow and Considerations	75
Integrated Indexes	76
File Space Not Needed	76
Allocating Space for Work Files and Sort Work Areas	76
Work Files.....	76
Sort Work Areas	90
Summary of Procedures for Operating CA IDMS/DB Reorg	90
CA IDMS/DB Reorg Processing--z/OS or OS/390 Environment.....	92
Process 1--Defining the New Areas.....	93
Process 2--Gathering Information.....	93
Process 3--Allocating Work Files	95
Process 4--Reorganizing the Database.....	97
CA IDMS/DB Reorg Processing--VM/ESA Environment	99
Process 1--Defining the New Areas.....	99
Process 2--Information Gathering.....	100
Process 3--Allocating Work Files	104
Process 4--Reorganizing the Database.....	106
CA IDMS/DB Reorg Processing--VSE/ESA Environment	107
VSE/ESA File Assignments	107
VSE/ESA File Processing Alternate Method	108
Process 1--Defining the New Areas.....	108
Process 2--Information Gathering.....	108
Process 3--Allocating Work Files	110
Process 4--Reorganizing the Database.....	112

Chapter 7: Messages 117

Condition Codes	117
Audit Report	118
VSE/ESA File Processing Alternate Method	118
Message Text	118
Return Codes	180

Index 183

Chapter 1: Introduction

This section contains the following topics:

[Understanding Syntax Diagrams](#) (see page 9)

Understanding Syntax Diagrams

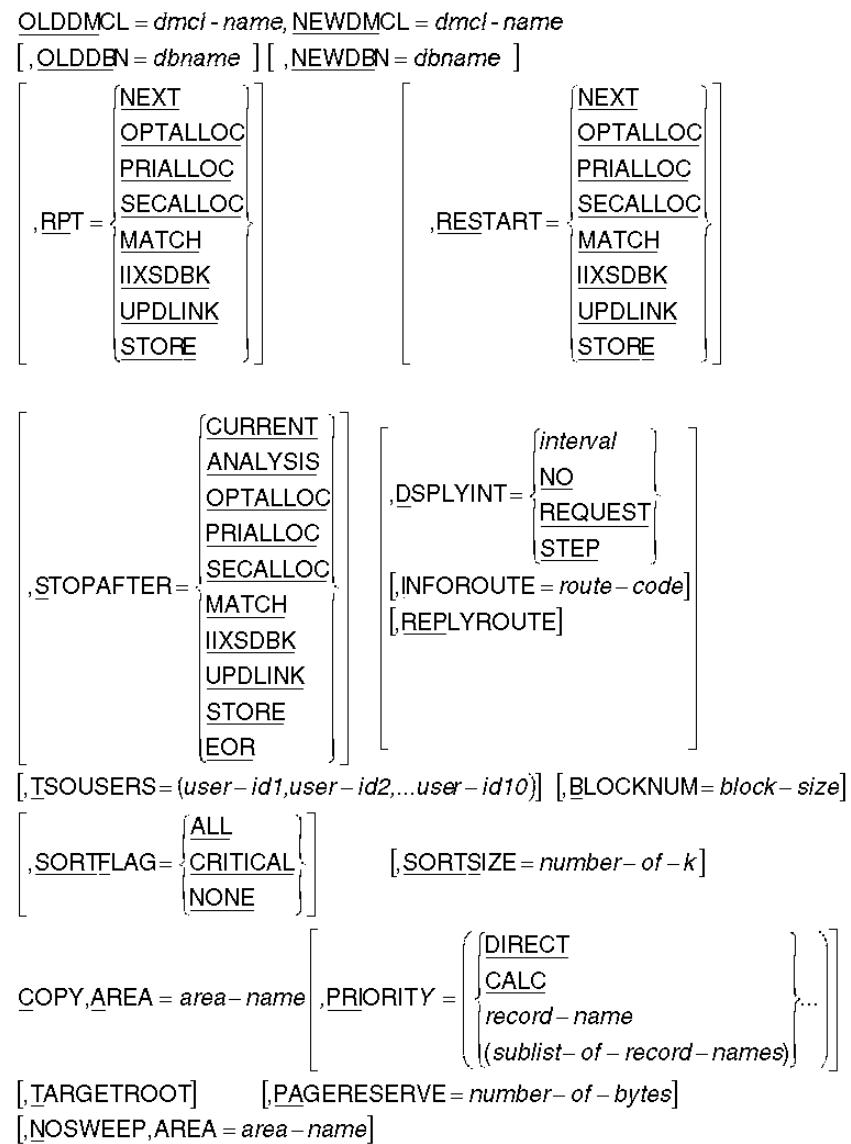
Look at the list of notation conventions to see how syntax is represented in this manual. The example following the list shows how the conventions are used.

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 of the keyword.
AREA= <i>area-name</i>	Variables appear in lowercase <i>italics</i> . 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

Example	Function
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



Chapter 2: General Information

CA IDMS/DB Reorg is a comprehensive, time-saving tool for database administrators and systems programmers who want to reorganize an CA IDMS 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.

This section contains the following topics:

- [Why Reorganize a Database?](#) (see page 13)
- [Select Any Area or Entire Database](#) (see page 13)
- [Improve System and CA IDMS/DB Performance](#) (see page 15)
- [How CA IDMS/DB Reorg Performs Reorganization](#) (see page 15)
- [Operational Flexibility During Reorganization](#) (see page 18)
- [CA IDMS/DB Reorg's Output](#) (see page 19)

Why Reorganize a Database?

Your CA IDMS 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 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.

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 the schema in the [Recognizes SR8 Index Areas](#) (see page 14) section.

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.

Leaves Relationships and Records Unaltered

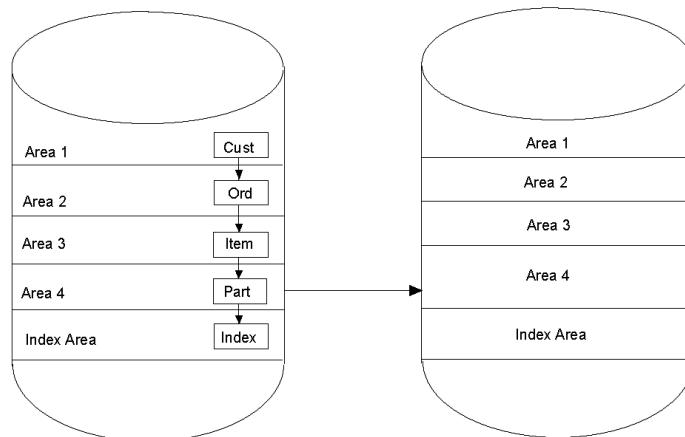
CA IDMS/DB Reorg will not alter the logical relationships between records or change the size or content of any record.

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.

Reorganizing a Single Area in a Database:

When reorganizing Area 4, Advantage CA-IDMS/DB Reorg expands the page ranges in Area 4 and updates set linkages in Area 3 and the Index Area.



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.

After you have run CA IDMS/DB Reorg against your database, you are able to achieve better system performance. Programs run more efficiently because I/Os are 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 database.

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. 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 are 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 the *CA IDMS Database Administration Guide* and *CA IDMS System Operations Guide*.

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

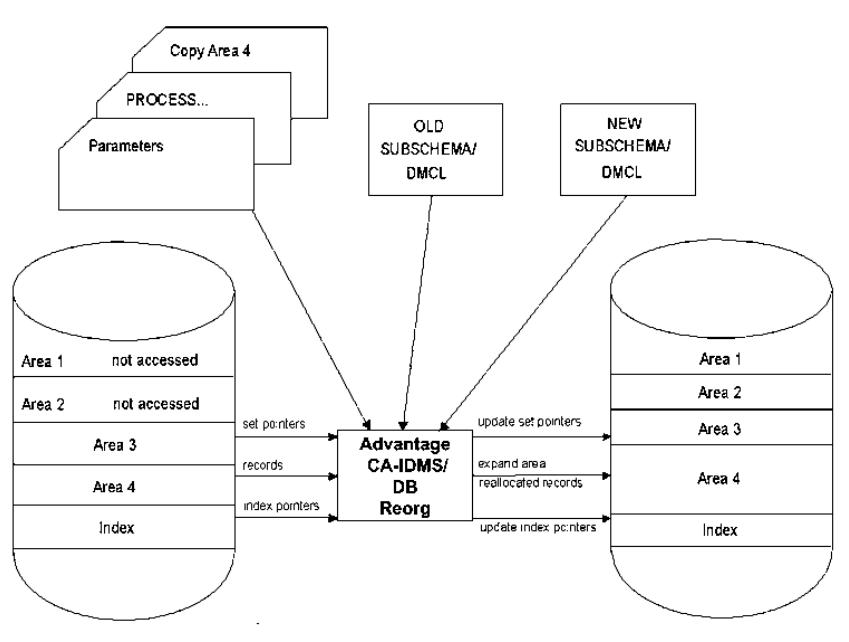
Reorganization Controlled by Parameter

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

Parameters and Commands include details about parameter-controlled reorganization capabilities.

CA IDMS/DB Reorg Process:**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, CA IDMS/DB Reorg:

- Updates integrated indexes when member records are reorganized
- Retargets CALC records
- Retargets VIA records
- Expands or reduces an area's page range
- Expands or reduces an area's pagesize
- Expands or reduces a record's page range
- Changes a record's location mode
- Moves a record to another area

Concepts includes details about these subschema-controlled reorganization capabilities.

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

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

Concepts contains details about these reorganization functions.

Operational Flexibility During Reorganization

CA IDMS/DB Reorg performs a reorganization on a step-by-step basis (steps are discussed in detail in Concepts). 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.

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 Using RESTART with STOPAFTER 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

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 an 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 Console Communications Facility.

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.

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 executes against your database, analyzing the reorganization that occurs 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.

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 3: Parameters and Commands

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.

This section contains the following topics:

- [Parameters and Their Functions](#) (see page 21)
- [PROCESS Statement](#) (see page 23)
- [COPY Statement](#) (see page 36)
- [NOSWEEP Statement](#) (see page 40)
- [Console Communications Facility](#) (see page 41)

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.

PROCESS Statement Overview

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.

COPY Statement Overview

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.

NOSWEEP Statement Overview

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.

CA IDMS/DB Reorg Parameter Summary:

OLDDMCL = *dmcl-name*, NEWDMCL = *dmcl-name*

[, OLDDBN = *dbname*] [, NEWDBN = *dbname*]

<u>RPT</u> = { NEXT OPTALLOC PRIALLOC SECALLOC MATCH IIXSDBK UPDLINK STORE }	<u>RESTART</u> = { NEXT OPTALLOC PRIALLOC SECALLOC MATCH IIXSDBK UPDLINK STORE }
--	--

<u>STOPAFTER</u> = { CURRENT ANALYSIS OPTALLOC PRIALLOC SECALLOC MATCH IIXSDBK UPDLINK STORE EOR }	<u>DSPLYINT</u> = { interval NO REQUEST STEP } [, <u>INFOROUTE</u> = <i>route-code</i>] [, <u>REPLYROUTE</u>]
--	---

[, TSOUSERS = {*user-id1*, *user-id2*, ..., *user-id10*}] [, BLOCKNUM = *block-size*]

<u>SORTFLAG</u> = { ALL CRITICAL NONE }	<u>SORTSIZE</u> = <i>number-of-k</i>
--	--------------------------------------

<u>COPY, AREA</u> = <i>area-name</i>	<u>PRIORITY</u> = { DIRECT CALC record-name (sublist-of-record-names) } ...
--------------------------------------	---

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

[, NOSWEEP, AREA = *area-name*]

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 z/OS or 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 Statement Syntax:

```

PROCESS, OLDSUB = subschema-name, NEWSUB = subschema-name,
OLDDMCL = dmcl-name, NEWDMCL = dmcl-name
[ ,OLDBN = 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]

[ ,ISOUSERS = {user-id1,user-id2,...user-id10} ] [ ,BLOCKNUM = block-size ]
[ ,SORTFLAG = {ALL
               CRITICAL
               NONE} ] [ ,SORTSIZE = number-of-k ]

```

PROCESS Parameter**PROCESS**

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

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 are affected by the reorganization. All records must be assigned to the same page group.

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/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.

OLDDMCL Parameter

Rules

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

,OLDDMCL=*dmcl-name*

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

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

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.

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.

RPT Parameter

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 helps you restart CA IDMS/DB Reorg correctly.

RPT Syntax and Options

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

[,RPT= {step-name}]

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 Concepts).

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

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.

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 db-key (IIXSDBK) Step.

UPDLINK

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

STORE

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

Rules

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

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 continues processing reorganization steps in their logical sequence (see Concepts 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

RESTART Syntax and Options

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

[,RESTART={step-name}]

This option indicates that a RESTART option follows. You select the reorganization step at which CA IDMS/DB Reorg resumes execution (steps are detailed in Concepts).

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

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.

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 db-key (IIXSDBK) step.

UPDLINK

Resumes with the Update Set Linkage (UPDLINK) step.

STORE

Resumes with the Store (STORE) step.

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

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. Concepts explains the function of each step.

STOPAFTER Syntax and Options

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

STOPAFTER=

This option indicates that a STOPAFTER option follows. You select the reorganization step to be completed before processing stops. Concepts explains the function of each step. The default is EOR.

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.

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 executes all remaining reorganization steps in sequence.

Rules

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

Using 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. 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 IPL, 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.

Examples of STOPAFTER and RESTART:

- 3) PROCESS, OLDSUB= *subschema-name*,
NEWSUB= *subschema-name*,
OLDDMCL= *dmcl-name*,
NEWDMCL= *dmcl-name*,
RESTART=NEXT

DSPLYINT Parameter

The DSPLYINT (display interval) parameter establishes the Console Communication Facility and indicates the times at which processing information is 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 INFOROUTE parameter or the REPLYROUTE parameter.

DSPLYINT Syntax and Options

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

[,DSPLYINT= {display interval}]

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

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. 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.

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.

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.

Rules

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

Route Code Syntax and Options

For a z/OS or 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 *route-code*

Indicates the route code for WTO messages.

route-code

Identifies the functional area that is to receive the status display at the intervals specified in the DSPLYINT parameter.

The default value for INFOROUTE is 2.

/REPLYROUTE *route-code*

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.

The default value for REPLYROUTE is 1.

Rules

- INFOROUTE and REPLYROUTE function only under z/OS and 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 are automatically routed for programmer information in addition to any route specified by using INFOROUTE or REPLYROUTE.

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

Code	Functional Area
12	Emulator Information
13	User Routing Code
14	User Routing Code
15	User Routing Code
16	(reserved for future use)

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 z/OS and OS/390.

SORTFLAG Parameter

[,SORTFLAG= {*ALL*
CRITICAL◀
STEP}]

[,SORTFLAG={*sort message level*}]

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

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. The default value is 200K.

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.

Rules

- 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.

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.

COPY Statement Syntax:

```
COPY AREA=area-name [ ,PRIORITY=( { DIRECT  
CALC  
record-name  
(sublist-of-record-names) } ... ) ]  
[ ,TARGETROOT ] [ ,PAGERESERVE=number-of-bytes ]
```

Copy Parameter

COPY

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

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 named area must exist within the old and 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 RPT or RESTART is specified on the PROCESS statement, the COPY statement is treated as a comment and ignored.
- A COPY statement can only be specified once for a given area.

PRIORITY Parameter

The optional PRIORITY parameter allows you to specify the order in which records are 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 store the highest priority records on their target pages before storing other records. The second priority listed takes precedence over the third listed, and so on.

Specifying 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 Concepts for a 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.

PRIORITY Syntax and Options

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

PRIORITY

Indicates priority options are listed in the order they are to receive priority. The default for priority is DIRECT, CALC, and VIA records after any user specified priorities.

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.

DIRECT

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

record-name

The records of this name are 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 receives storage priority of equal value. A sublist must be enclosed in parentheses and each record name must be separated by a comma.

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.

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 reserves 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. The default value is zero (no reserve).

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 stores 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. Minimally, the root is as large as the minimum ROOT defined in the new subschema and as large as can 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.

NOSWEET Statement

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

The NOSWEET 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 NOSWEET is used, CA IDMS/DB Reorg resolves 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.

Using NOSWEET

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 NOSWEET 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 automatically sets the NOSWEET parameter.

NOSWEET Syntax and Options

[NOSWEET,AREA=*area-name*]

NOSWEET

Initiates NOSWEET statement processing and indicates that an area name parameter follows.

AREA=*area-name*

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

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.

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]*

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.

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 is 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 is displayed when each step starts and ends.

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.

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 appears on the Audit Report. The DUMP part of the command is optional. It directs CA IDMS/DB Reorg to produce a dump.

Chapter 4: Concepts

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.

This section contains the following topics:

- [Understanding CA IDMS/DB Reorg Processing](#) (see page 43)
- [CA IDMS/DB Reorg's Inputs](#) (see page 44)
- [CA IDMS/DB Reorg's Record Allocation Techniques](#) (see page 45)
- [CA IDMS/DB Reorg Reorganization Capabilities](#) (see page 51)
- [Reorganization Steps](#) (see page 57)

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.

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.

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 updates 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:

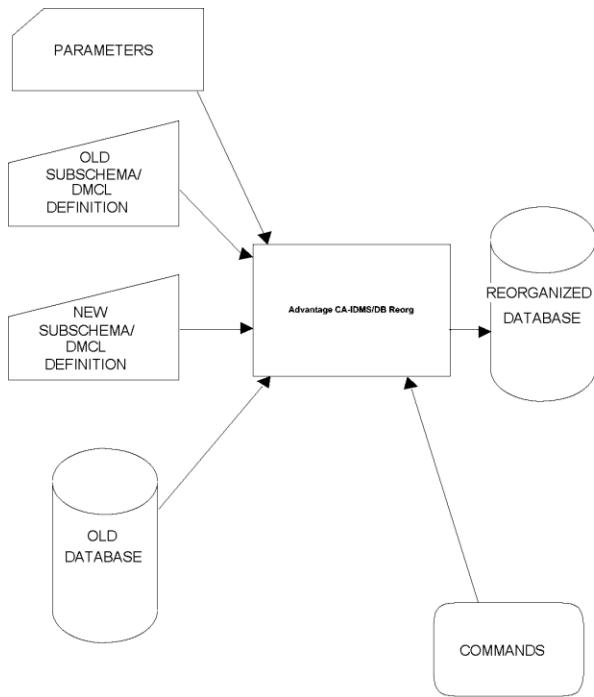
- Inputs
- Record allocation techniques
- Reorganization capabilities
- Reorganization steps.

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.

- **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.
- **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 are 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.

Inputs to CA IDMS/DB Reorg:



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 performs fewer I/Os and response time improves significantly.

Minimize Record Overflow

CA IDMS/DB Reorg overflows 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.

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 fits. 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, creates 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 TARGETROOT Parameter.

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 allocates the direct record to the next available page where the entire record fits. If there is no page in the DIRECT record's page range and the record is variable length, CA IDMS/DB Reorg fragments 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.

Allocating CALC Records

The target page for a CALC record is determined by the IDMSCALC routine. A CALC record is 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 finds the next available page in the record's page range that can accommodate the entire record.

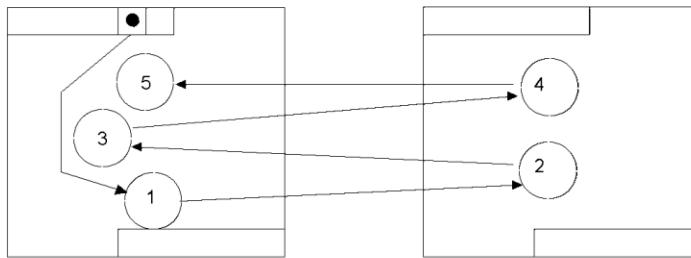
If CALC records must overflow to another page, CA IDMS/DB Reorg overflows the CALC records with symbolic keys that are higher than those of records stored on the home page. This reduces the average number of I/Os required to retrieve CALC records.

When a variablelength 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 stores 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 firstavailable page which can fit the entire fragment.

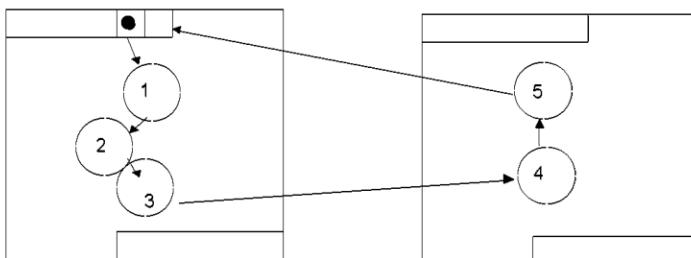
In the database, duplicate CALC keys maintain the same order as in the old database.

Optimizing the CALC Set:

Before Advantage CA-IDMS/DB Reorg:
Three I/Os are required to retrieve the CALC record with the symbolic key 3. The maximum number of records per page is 3.



After Advantage CA-IDMS/DB Reorg:
One I/O is required to retrieve the CALC record with the symbolic key 3.



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 is allocated after all primary allocations are made.

VIA records are also allocated in logical set order (the physical and logical order are the same). This can reduce the number of I/Os required to walk the set.

In an active online environment where buffer use is low, the number of I/Os to walk a VIA set is the same as the number of page changes encountered when walking the set. The simple example in the following figure 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.

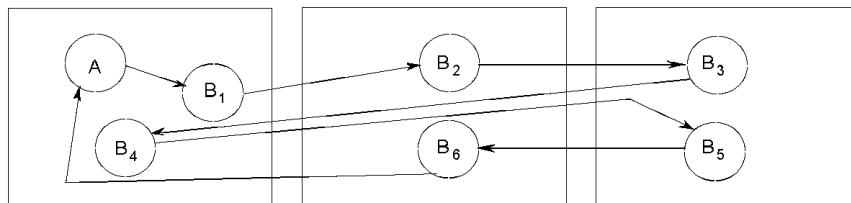
Sorted VIA Records

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. CA IDMS/DB Reorg overflows 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.

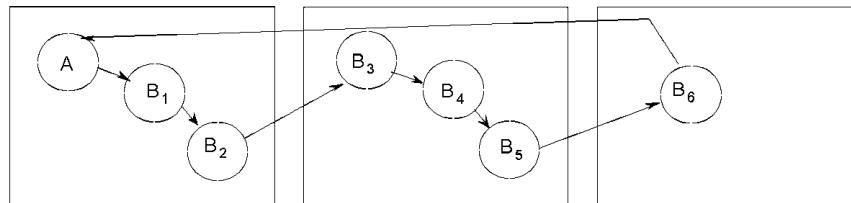
A variable length, sorted VIA record that does not fit on its owner's page has its root portion stored there if the TARGETROOT parameter was specified. This reduces the average number of I/Os required to retrieve a sorted VIA member when using the FIND/OBTAIN USING command because CA IDMS/DB looks at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

Optimizing VIA Records:

Before Advantage CA-IDMS/DB Reorg:
When the logical and physical VIA set order are not the same, excessive I/Os are required to walk the set. In this example, six I/Os are required to walk the set. The maximum number of records per page is 3.



After Advantage CA-IDMS/DB Reorg:
The logical and physical VIA set order are identical. Three I/Os are required to walk the set.



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) are 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 the previous figure, the type B records are clustered near their type A owners.

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

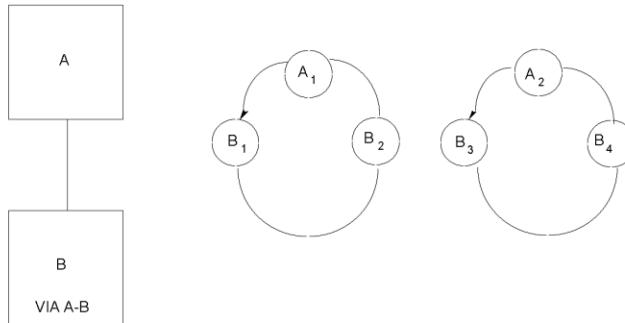
You can rank individual records or groups of record types. For example, if you specify a sublist of record types as first priority, and DIRECT records as second priority, you are overriding CA IDMS/DB Reorg's automatic record allocation techniques. By using the PRIORITY parameter you can tailor the physical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

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 PAGERESERVE Parameter.)

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.

User-Defined Clustering:

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))



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.

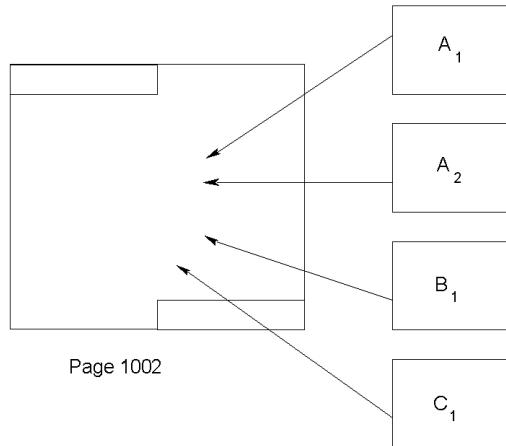
Reorganization Capabilities Controlled by Parameters

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:

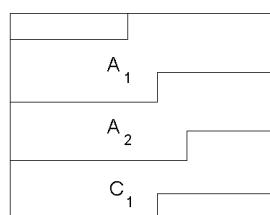
- **Resolve record/page conflicts.** The PRIORITY parameter allows you to specify which record types (or group of record types) are to be stored on the target page and which ones should overflow if the page is full. The priorities you assign take effect only when records overflow. The priorities you 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.
- **Define clustering.** The PRIORITY parameter also allows you to define how VIA clusters are restored. The schema which follows this list shows how you can direct CA IDMS/DB Reorg to cluster VIA records in the order in which they are 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 rebuilds 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.

Resolving Record/Page Conflicts:

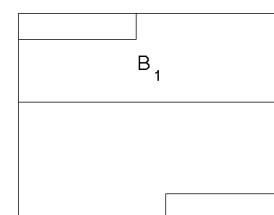
These four record occurrences target to page 1002. The page can only accommodate three records.



Given the priority statement for record types A, B, and C:
PRIORITY=(A,C,B)
 Advantage CA-IDMS/DB Reorg allocates the records this way:



Page 1002



Page 1003

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 sweeps 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 is turned on automatically, and no sets are 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.

Parameters and Commands includes details for each parameter and command.

Reorganization Capabilities Controlled by Subschema and DMCL Definition

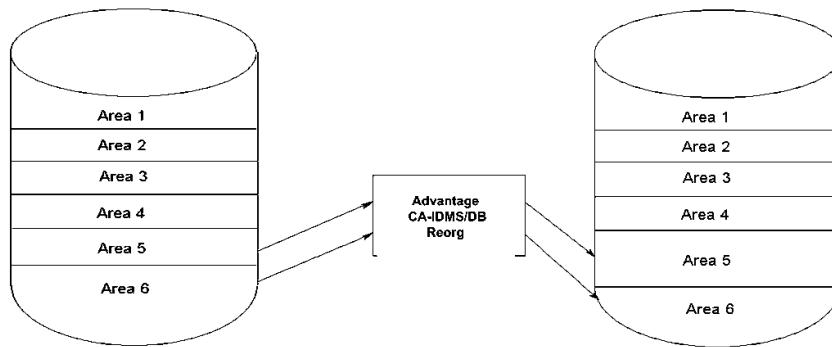
Many reorganization capabilities depend on how you have defined the new subschema and DMCL. During reorganization, CA IDMS/DB Reorg compares the area, set, index, and record definitions in the new subschema with those in the old subschema. After comparing the old subschema with the new subschema, CA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

Changing the Page Range of Two Areas:

Advantage CA-IDMS/DB Reorg allows you to copy only two areas of the database. You can change the page ranges for both areas.



- **Eliminate logically deleted records.** When CA IDMS logically deletes records, the data is removed 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.

Automatic Reorganization Controlled by Internal Logic

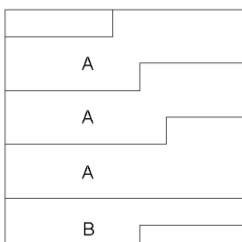
Many of CA IDMS/DB Reorg's capabilities are controlled automatically by its own internal processing logic. For these functions you do not have to enter a parameter or define subschemas. Through its own internal processing logic CAIDMS/DB Reorg has the capabilities to:

- **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.
- **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.
- **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.

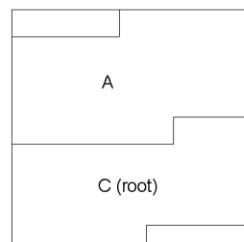
Minimizing Fragmented Records:

Before Advantage CA-IDMS/DB Reorg:

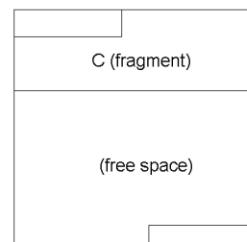
Three I/Os are required to retrieve the CALC record with the symbolic key 3. The maximum number of records per page is 3.



Page 2001



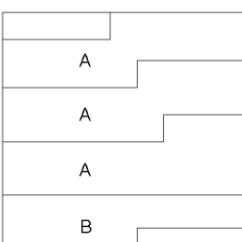
Page 2002



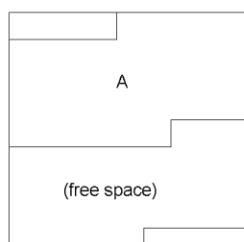
Page 2003

After Advantage CA-IDMS/DB Reorg:

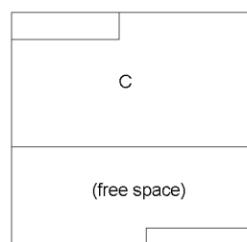
One I/O is required to retrieve the CALC record with the symbolic key 3.



Page 2001



Page 2002



Page 2003

Reorganization Steps

CA IDMS/DB Reorg performs reorganization incrementally by executing a series of steps. This incremental approach to reorganization makes possible several operational benefits.

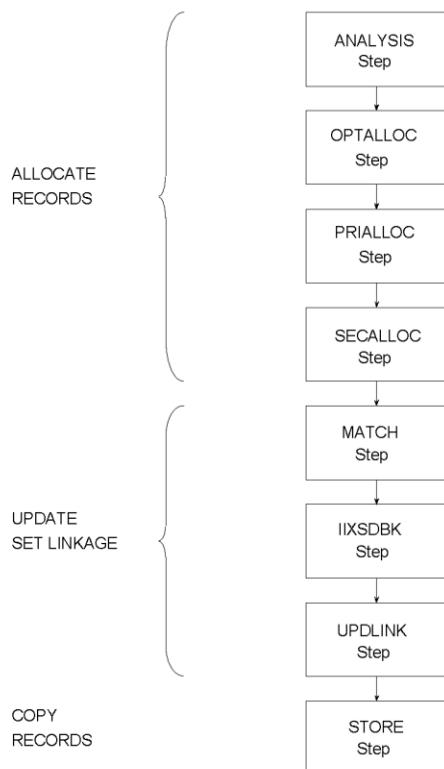
Step-by-step reorganization permits you to stop and restart the utility after completion of any step. By using the restart feature (see STOPAFTER and RESTART Parameters), you can interrupt processing and restart it later.

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 takes. See System Output for complete information on the status display available at the operator console.

CA IDMS/DB Reorg must execute each reorganization step sequentially. 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

Reorganization Steps and Their Logical Sequence:



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 data base 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:

- Reconstructs the integrated index
- Allocates real db-keys for the new SR8 records
- Creates 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.

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.		
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.		

Step/Function	Allocates Records to Area Specified in COPY Statement	Updates Set Linkage	Accesses Old Database	Access New Database
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 5: System Output

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.

This section contains the following topics:

- [CA IDMS/DB Reorg Output](#) (see page 63)
- [Audit Report](#) (see page 64)
- [Status Display](#) (see page 73)
- [DBKEYS Interface File](#) (see page 73)

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 utilities.

Audit Report

The Audit Report is produced automatically every time you run CA IDMS/DB Reorg. This report traces all CAIDMS/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.

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 Status Display Fields.

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 DBKEYS Interface File

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. The following figure shows two pages from an Audit Report with the following information:

Report Field	Description
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	Lists 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).

Report Field	Description
Processing Messages	<p>Lists 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.</p> <p>The 'Audit Report - OPTALLOC Step' schema at the end of this section shows processing messages generated by the routine that checks parameter input and the OPTALLOC step.</p>

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 Messages for an explanation of the three types of severity codes and information about specific messages.

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
			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	COPY, AREA=LOC-REGION		
		NKWP091I INPUT PARAMETER STATEMENT	COPY, AREA=DEPT-REGION		
		PARM043I NO PARAMETER ERRORS DETECTED	COPY, AREA=STUDENT-REGION		
		NKWP091I INPUT PARAMETER STATEMENT	COPY, AREA=CLASS-REGION		
		PARM043I NO PARAMETER ERRORS DETECTED	COPY, AREA=MISC2-REGION		
		NKWP091I INPUT PARAMETER STATEMENT			
		NKWP094I END OF PARAMETER INPUT			
		AUDT002I STEP: PARM, ENDED, CC= 0000	DATE: mm/dd/yy	TIME: hh:mm:ss	

Audit Report - OPTALLOC Step:

ID	RELEASE	CA-IDMS/DB REORG	DATE	TIME	PAGE
		AUDIT REPORT	mm/dd/yy	hh:mm:ss	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 IIXSKRY 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	

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.

The following figure shows an Area Recap for Old Subschema with the following information:

Report Field	Description
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.

Report Field	Description
PROCESSING-TYPE	<p>The type of processing CA IDMS/DB Reorg performs, according to your parameter specifications, against each area in the old subschema. For each area listed one of five processing types may appear:</p> <ul style="list-style-type: none"> ■ SWEEP--Indicates that an area sweep is performed. ■ NOT ACCESSED--Indicates that CA IDMS/DB Reorg will not access the area. ■ NOSWEEP - SET WALKING--Indicates CA IDMS/DB Reorg walks 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 is copied to the new database.

Audit Subreport--Area Recap for Old Subschema:

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 -----																																		
<table> <thead> <tr> <th>----- AREA -----</th><th>NBR-PAGES</th><th>PROCESSING-TYPE</th></tr> </thead> <tbody> <tr> <td>BOGUS-1-REGION</td><td>30</td><td>NOT ACCESSED</td></tr> <tr> <td>BOGUS-2-REGION</td><td>30</td><td>NOT ACCESSED</td></tr> <tr> <td>BOGUS-3-REGION</td><td>30</td><td>NOT ACCESSED</td></tr> <tr> <td>CLASS-REGION</td><td>30</td><td>COPY</td></tr> <tr> <td>DEPT-REGION</td><td>15</td><td>COPY</td></tr> <tr> <td>LOC-REGION</td><td>5</td><td>COPY</td></tr> <tr> <td>MISC1-REGION</td><td>20</td><td>SWEEP</td></tr> <tr> <td>MISC2-REGION</td><td>20</td><td>COPY</td></tr> <tr> <td>STUDENT-REGION</td><td>80</td><td>COPY</td></tr> </tbody> </table>					----- 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
----- 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																																

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 the [Operations](#) (see page 75) chapter for instructions on how to calculate file sizes.

The following figure illustrates a Set Linkage Recap for Area with the following information:

Report Field	Description
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 is 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 uses 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 is stored in a noncopy area or has a location mode which is not VIA.

Audit Subreport--Set Linkage Recap for Area:

ID	RELEASE Rnn.nnn	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 dmcl-name, DBNAME dbname					
NLYZ005I END OF ANALYSIS REPORTS					
AUDT002I STEP: ANALYSIS, ENDED, CC=0000			DATE: mm/dd/yy	TIME: hh:mm:ss	

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. The following figure shows a page from the sample Audit Report which contains Restart Information Messages.

Message ID--Restart Information Message ID. Messages are described in [Messages.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 (z/OS and OS/390 DDnames).
- File status--Disposition of the files as though each step were a separate z/OS or OS/390 step.
 - **OLD/NEW**--Indicates the current status of each dataset. NEW indicates the file is 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.

Restart Information Messages:

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 hh:mm:ss	
		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 UPDATED BY SECALLOC STEP		
	UNALLOC	OLD KEEP	CREATED BY PRIALLOC STEP READ BY SECALLOC STEP		
	PRIREAL	OLD KEEP	CREATED BY PRIALLOC STEP READ BY MATCH STEP		
	SECREAL	OLD KEEP	CREATED BY SECALLOC STEP READ BY MATCH STEP		
	ALLOCX	OLD KEEP	CREATED BY MATCH STEP READ BY UPDLINK STEP		
	IIXSRKY	OLD KEEP	CREATED BY IIXSDBK STEP READ BY IIXSDBK STEP		
	IIXEXOL	OLD KEEP	CREATED BY MATCH STEP READ BY IIXSDBK STEP		
	IIXEXTR	OLD KEEP	CREATED BY IIXSDBK STEP READ BY UPDLINK STEP		
	DBREC	OLD KEEP	CREATED BY UPDLINK STEP READ BY STORE STEP		

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 a z/OS or 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 helps you gauge CA IDMS/DB Reorg run time.

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. Parameters and Commands provides a complete explanation of the Status Display parameters and commands.

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 Messages.

```

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

```

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 utilities or with DB-EZ Reorg. The file is described in the following table.

DBKEYS Interface File.:

Field Name	Element Description	Explanation
OLDDBKEY	PIC S9(8) COMP SYNC	Old db-key

Field Name	Element Description	Explanation
NEWDBKEY	PIC S9(8) COMP SYNC	New db-key
SR3DBKEY	PIC S9(8) COMP SYNC	SR3 db-key. See note.
RECORDID	PIC S9(4) COMP	Record id
SR3RECID	PIC S9(4) COMP	Original record id. 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 6: Operations

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

This section contains the following topics:

[Database Access](#) (see page 75)

[Operational Flow and Considerations](#) (see page 75)

[CA IDMS/DB Reorg Processing--z/OS or OS/390 Environment](#) (see page 92)

[CA IDMS/DB Reorg Processing--VM/ESA Environment](#) (see page 99)

[CA IDMS/DB Reorg Processing--VSE/ESA Environment](#) (see page 107)

Database Access

CA IDMS/DB Reorg utilizes CA IDMS 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 update jobs are accessing the database. CA IDMS/DB Reorg will not attempt to run against an area locked for update.

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 DSPLYINT Parameter.
- 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.

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 SECREAL and SECALX2.

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.

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. The following figures contain information to help you allocate space to the files.

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.

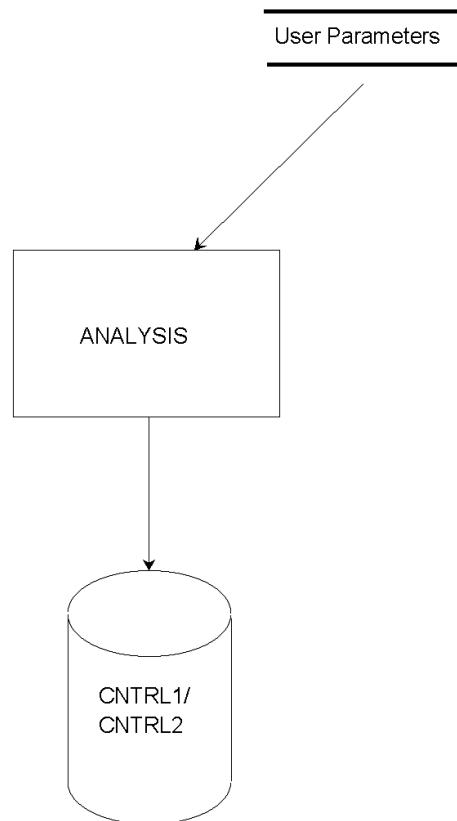
Each work file requires a certain amount of space when you assign it to disk. Calculate the space for every file, except the EXTRACT file, by using the number of records (from the CA IDMS/DB Analyzer reports or from your own sources) and the information given for estimating work file space in the following figures. 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 (Audit Subreport-Set Linkage Recap for Area) in addition to the number of records listed on the CA IDMS/DB Analyzer Set Report.

When you use the information in the following figures 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 the following figures.

ANALYSIS Step

Work File Usage For Analysis Step:

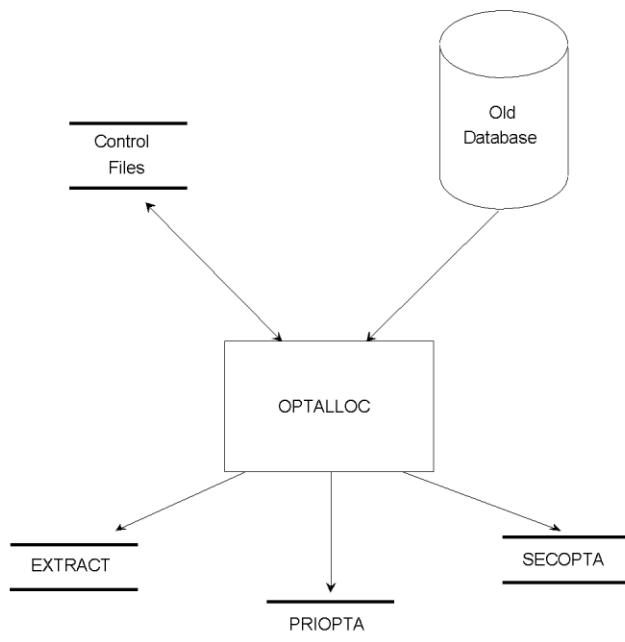


Estimating Space for Work Files (CNTR1 CNTR2):

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
CNTRL1	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.
CNTRL2				

OPTALLOC Step

Work File Usage For OPTALLOC Step:

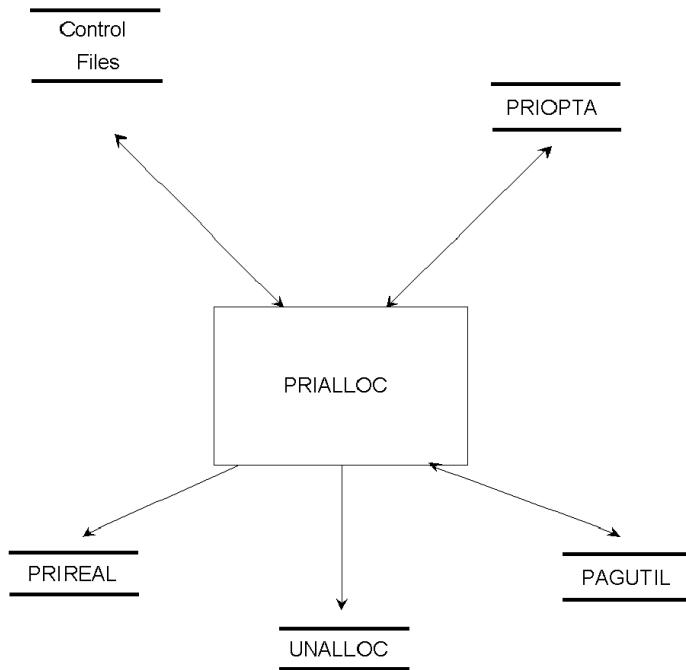


Estimating Space for Work Files (EXTRACT, PRIOPTA,SECOPTA):

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
EXTRACT	FB	22	6138	<p># of EXTRACT records = (# 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 Audit Subreport--Set Linkage Recap for Area 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 = (# of VIAs not in the same page range as the owner) + (# of all BLSR8 records)</p>

PRIALLOC Step

Work File Usage For PRIALLOC Step:



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.

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
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.

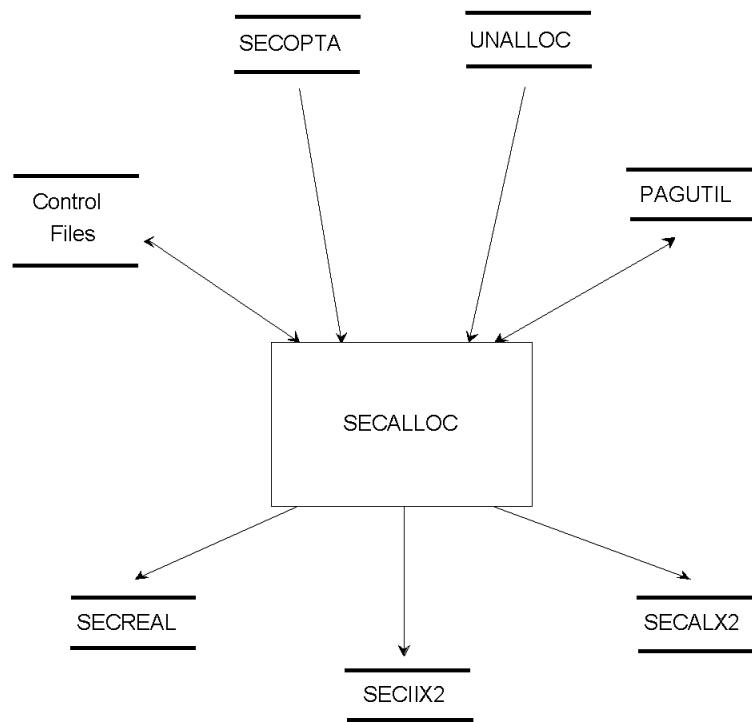
Example:

AREA-1 has 200 pages
AREA-2 has 2000 pages
PAGUTIL = 3 + (200/1020) + (2000/1020)
= 3 + 1 + 2
= 6 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 z/OS or OS/390 when this is a VSAM file.

SECALLOC Step

Work File Usage for SECALLOC Step:



Estimating Space for Work Files (SECREAL):

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
SECREAL	FB	22	6138	<p># of SECREAL records =</p> <p>(# 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)</p> <p>Note: This calculation must be done for each indexed set in the area.</p> <p>BL - Bottom Level</p> <p>UL - Upper Level</p>

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 **For SECREAL and SECALX2 (optional detailed calculation)**.

Estimating Space for Work Files (SECALX2, SECIIIX2):

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
SECIIIX2	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 is not 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>

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
SECALX 2	VB	18	6138	<p>The space required is 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 * the # of BL SR8 records) + (3 * UL SR8 record) + (1 * # of UL SR8 entries).</p> <p>where:</p> <p>UL SR8 entries = (# of SR8 records - 1)</p> <p>See For SECREAL and SECALX2 (optional detailed calculation) for more information. :tnote.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.</p> <p>BL - Bottom Level SR8</p> <p>UL - Upper Level SR8</p>

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.

Bottom Level Statistics--Used Table Entries (existing DB)

BL = 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.

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:

Bottom level SR8 records 2100
10 = 210

Upper level SR8 records 210
10 = 21

21
10 = 3 (rounded up)

(Top level SR8 record) 3
10 = 1 (rounded up)

SECREAL

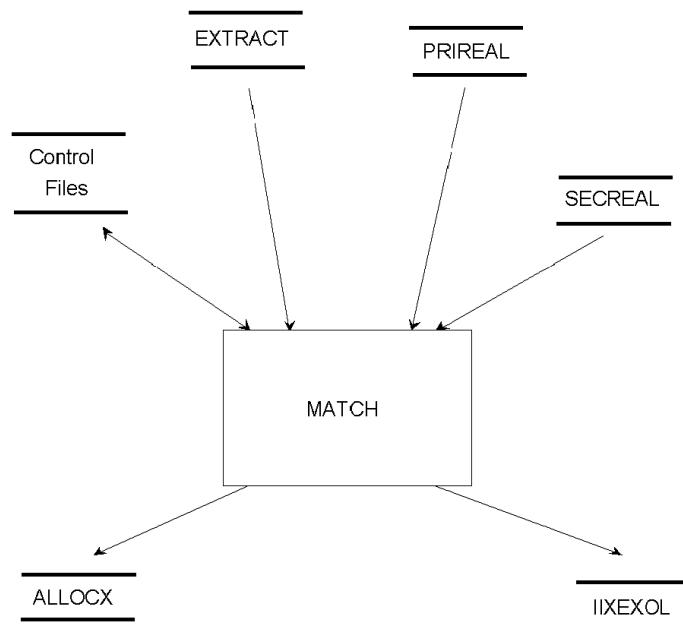
number of SECOPTA records
+ number UNALLOC records
+ (210 + 21 + 3 + 1) total SR8 records
+ 2100 SR8 bottom level entries

SECALX2

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

MATCH Step

Work File Usage For Match Step:



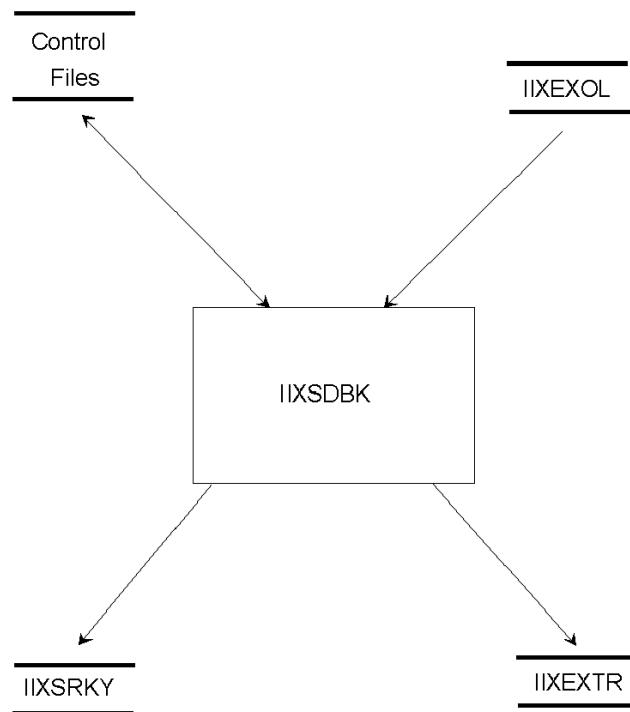
Estimating Space for Work Files (ALLOCX, IIXEXOL):

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).

Work File(s)	REC FM	LRECL	BLKSIZE	Space Allocation
IIXEXOL	FB	20	6140	<p>See note.</p> <p># of IIXEXOL records = $(3) + (2 * \text{BL SR8 records}) + (1 * \text{UL SR8 records})$</p> <p>Note: This file is required only if the area being copied contains an indexed set sorted by DB-Key.</p> <p>BL - Bottom Level SR8</p> <p>UL - Upper Level SR8</p>

IIXSDBK Step

Work File Usage for IIXSDBK Step:

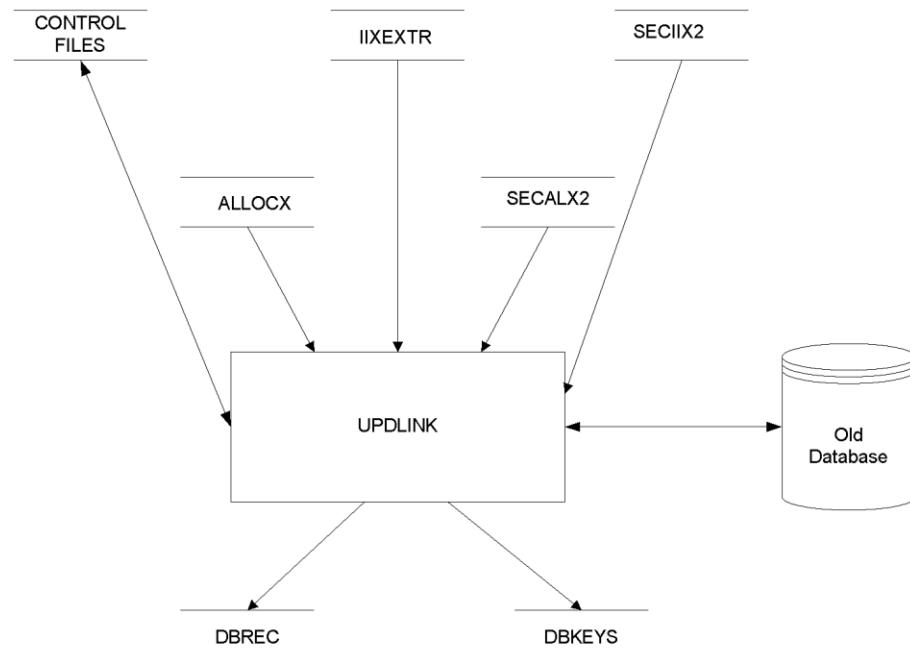


Estimating Space for Work Files (IIXSRKY IIXEXTR):

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	<p># of IIXEXTR records = $(3 + (2 * \text{BL SR8 record}) + (1 * \text{UL SR8 records}))$</p> <p>Note: These files are needed only if the area being updated contain index sets sorted by db-key.</p> <p>BL - Bottom Level SR8 UL - Upper Level SR8</p>

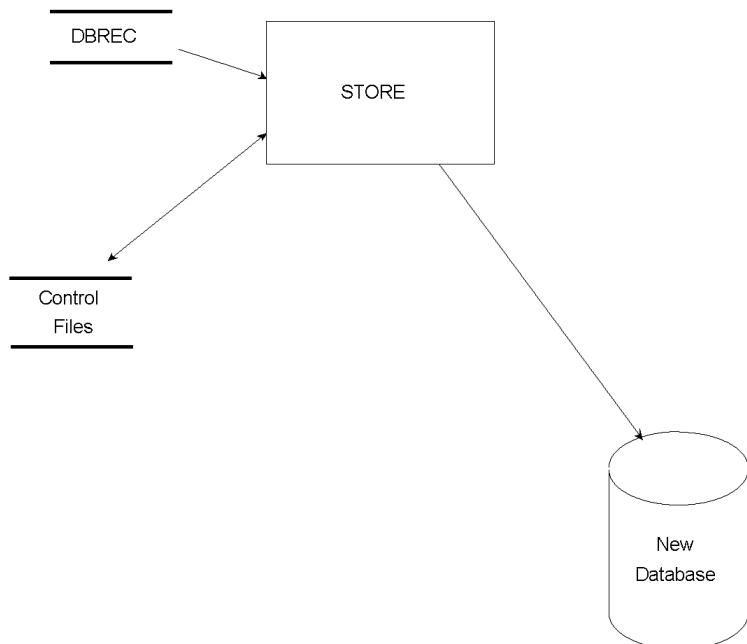
UPDLINK Step

Work File Usage for UPDLINK Step:



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 = $(((24 * \# \text{ of occurrences}) + \text{prefix} + \text{data}) \text{ record-01}) + \…$ $(((24 * \# \text{ of occurrences}) + \text{prefix} + \text{data}) \text{ record-nn})$
DBKEY	FB	16	BLOCK- NUM parameter	# of DBKEY records = $(\# \text{ of records in reorganized areas}) - (\text{SR8 records}) * 16$

STORE Step**Work File Usage For STORE Step:**

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 must adjust space requirements to accommodate the largest of the sort input files. The following figure 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 the following figures.

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

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 outstanding operator reply. See DSPLYINT Parameter.

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

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

- a. First PROCESS stops here.

5. Allocates 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. In z/OS or OS/390, use IEFBR14 for deleting and allocating work files.

```
PROCESS,
OLDSUB=subschema-name,
NEWSUB=subschema-name,
OLDDMCL=dmcl-name,
NEWDMCL=dmcl-name,
STOPAFTER=CURRENT
```

- a. 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.

```
PROCESS,
OLDSUB=subschema-name,
NEWSUB=subschema-name,
OLDDMCL=dmcl-name,
NEWDMCL=dmcl-name,
RESTART=NEXT,
STOPAFTER=UPDLINK
```

PRIALLOC Step
SECALLOC Step
MATCH Step
IIXSDBK Step
UPDLINK Step

- a. 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.)

```
PROCESS,  
  OLDSUB=subschema-name,  
  NEWSUB=subschema-name,  
  OLDDMCL=dmcl-name,  
  NEWDMCL=dmcl-name,  
  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:

8. Run CA IDMS/DB Reorg in Step mode.
9. All steps prior to the STORE step must specify the 'olddmcl' in the SYSIDMS parameters.
10. The STORE step must specify the 'newdmcl' in the SYSIDMS parameters.

CA IDMS/DB Reorg Processing--z/OS or OS/390 Environment

The following example illustrates a simple way to execute CA IDMS/DB Reorg in a z/OS or 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 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 Summary of Procedures for Operating CA IDMS/DB Reorg 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 USREEXEC1, USREEXEC2, USREEXEC3, and USREEXEC4 provide examples of JCL for executing CA IDMS/DB Reorg.

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.

Process 2--Gathering Information

This process gathers the information needed to determine the best strategy for a database reorganization. It initiates CAIDMS/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=
//SYSIPT      DD      *
      PROCESS,      OLDSUB=subschema-name,
                  NEWSUB=subschema-name,
                  OLDDMCL=dmcl-name,
                  NEWDMCL=dmcl-name,
                  STOPAFTER=ANALYSIS
      COPY,      AREA=area-name
      &inve...ip.

/*
```

Key for Sample JCL to Obtain AUDIT/ANALYSIS Report:

Parameter	Description
your.loadlib	The dataset name of the z/OS or OS/390 PDS library into which CA IDMS/DB Reorg was installed.
idms.loadlib	The dataset name of the z/OS or OS/390 PDS library in which your CA IDMS modules reside. You <i>must</i> supply a separate statement for every OS/390 PDS library needed.
dbmsdds	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.
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.

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 <i>must</i> include a COPY parameter statement for each area that is to be reorganized.)

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) -
)

```

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@Q  DD   DSN=*.PAGUTIL,VOL=REF=*.PAGUTIL,
                 DISP=SHR,DCB=DSORG=DA

```

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)
```

Key for Allocating Work Files JCL:

Parameter	Description
a	The appropriate SYSOUT class for your installation.
vsam.work.space	The dataset name of the <i>one</i> 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.
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.

Process 4--Reorganizing the Database

This process reorganizes the database by executing the reorganization process of CA IDMS/DB Reorg as one z/OS or 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
//dbfil01  DD  DSN=user.dbfil01,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
//SYSLST          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=newdmcl
//SYSIPT          DD  *
PROCESS,          OLDSUB=subschema-name,
                           NEWSUB=subschema-name,
                           OLDDMCL=dmcl-name,
                           NEWDMCL=dmcl-name,
                           RESTART=NEXT /*
```

Key for Sample JCL to Reorganize a Database:

Parameter	Description
your.loadlib	The dataset name of the z/OS or OS/390 PDS library into which CA IDMS/DB Reorg was installed.

Parameter	Description
idms.loadlib	The dataset name of the z/OS or OS/390 PDS library in which your subschema(s) and DMCL(s) reside. You <i>must</i> supply a separate statement for every OS/390 PDS library needed.
dbfil01/dbfilnn	The DDnames of the database files to be accessed.
user.dbfil01 /user.dbfilnn	The dataset name of the file accessed by either the old subschema or the new subschema. You <i>must</i> 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 z/OS or 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 <i>must</i> 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 <i>one</i> VSAM file to be used by CA IDMS/DB Reorg. (Refer to the appropriate JCL in Process 3--Allocating Work Files 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.

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.

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 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 Summary of Procedures for Operating CA IDMS/DB Reorg 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.

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.

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=dmcl-name,
  NEWDMCL=dmcl-name,
  STOPAFTER=ANALYSIS,
  COPY, AREA=area-name

  &inverlip.

/*
  TRACE OFF; SIGNAL ON ERROR
  */
CA_LOADLIB_FN      = 'yourlib'
IDMS_LOADLIB_FN    = 'idmslib'
SORTLIB_FN         = 'sortlib'
/*
  /*
  /* Link and access the Minidisks containing the required librarie(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 SYSLST  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='newdmcl' 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'
```

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 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.

Parameter	Description
fn ft m	The filename, filetype, 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 filename, filetype, and file mode (respectively) of the BDAM PAGUTIL file. The file mode number must be 6.
newdcml	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.
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 <i>must</i> include a COPY parameter statement for each area that is to be reorganized.)

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

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 z/OS, 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) -
  )

```

Following is a sample input file for the AMSERV command shown in Process 3--Allocating Work Files. Name this file **mastcat** AMSERV fm.

```

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

```

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

AMSERV vsamwork

Following is 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))

```

If the PAGUTIL file is to be relative BDAM, use the following JCL.

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

Key for VM/ESA JCL:

Parameter	Description
<i>vsam.work.space</i>	The file name of the VSAM PAGUTIL file to be used by CA IDMS/DB Reorg.
<i>record.numbers</i>	The number of records to be allocated in the VSAM work space.
<i>usetype</i>	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.
<i>mastcat</i>	The file name of the VSAM master catalog.
<i>dskxxx</i>	The name of the minidisk on which to allocate a the VSAM files.
<i>n</i>	The number of cylinders to be allocated for the master catalog.
<i>cuu</i>	The appropriate minidisk device address for the BDAM PAGUTIL file.
<i>nn</i>	The number of cylinders required to hold the BDAM PAGUTIL file, based upon a block size of 8168. See :tref refid=T5003. for the number of PAGUTIL records.
<i>fm</i>	The file mode for the relevant file.
<i>fm ft fm6</i>	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.*

Process 4--Reorganizing the Database

This process reorganizes the database. Sample parameters for the 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,
NEWDMCL=dmc1-name,
RESTART=NEXT

```

This process uses the same sample EXEC to run CA IDMS/DB Reorg as shown in Process 2--Information Gathering.

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 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 Summary of Procedures for Operating CA IDMS/DB Reorg 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.

SAMPJCL 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.

VSE/ESA File Assignments

Even if you use a storage management tool such as BrightStor™ CA-DYNAM, CA IDMS/DB Reorg requires an ASSGN statement for every file except SORTWKnn. 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 Work Files and Corresponding Logical Units table in the [Allocating Work Files](#) (see page 110) section. Unless the ASSIGN specifies VSAM or BDAM, the file may be defined with either DLBL or TLBL.

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.

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.

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,
OLDBN=db-name,
NEWDBN=db-name,
STOPAFTER=ANALYSIS
COPY, AREA=area-name
&invellip.

/* * $$ EOJ

```

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.
subschema-name	Two subschema names are required. The firstsubschema 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 (OLDBN=) 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 <i>must</i> include a COPY parameter statement for each area that is to be reorganized.)

Process 3--Allocating Work Files

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. 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) -
                      ) -
                      ( -
                      NAME('vsam.work.space.DATA') -
                      )
/*
*

```

Work Files and Corresponding Logical Units:

dbmsnms	SYSnns
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
SECIIX2	SYS029
DBKEYS	SYS032

Key for Sample JCL to Allocate VSAM Work File PAGUTIL:

Parameter	Description
vsam.work.space	The file ID of the <i>one</i> 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.

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 ACCOMMODATE 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  
  
          &inverellip.  
// 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 PARAMETERS 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=dmcl-name,
NEWDMCL=dmcl-name,
OLDDBN=db-name,
NEWDBN=db-name,           RESTART=NEXT
/*
/*
* $$ E0J

```

Key for Sample JCL to Reorganize a Database:

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

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 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 <i>must</i> include a COPY parameter statement for each area that is to be reorganized.)

Chapter 7: Messages

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 *xxxxnnns* 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 is 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 finishes executing the step in which the warning was generated, but it will not execute any subsequent reorganization steps. If the warning refers to an invalid value in a parameter statement, CA IDMS/DB Reorg executes the Analysis step so that you receive an Audit/Analysis report.

(E) Error--A message with a severity code of E is an error message. An error message reports an unrecoverable 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 continues processing until it reaches the end of the current step.

This section contains the following topics:

[Condition Codes](#) (see page 117)

[Audit Report](#) (see page 118)

[VSE/ESA File Processing Alternate Method](#) (see page 118)

[Message Text](#) (see page 118)

[Return Codes](#) (see page 180)

Condition Codes

In a z/OS or 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.

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 System Output.

VSE/ESA File Processing Alternate Method

Occasionally, you 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.

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, DURhh.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=nnnnnnnnnnnnnnnnnn

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
PROGRAMmodule-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 PROGRAMUSRAUDT

Reason:

USRAUDT received an invalid communication switch.

Action:

Contact Computer Associates Product Support.

AUDT012I DB/REORG IS BETWEEN STEPS PERFORMING A GENERAL SERVICEFUNCTION

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-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.

CALL903E NOT ENOUGH STORAGE - PGM=module-name

Reason:

Module cannot be loaded due to insufficient GETVIS area (VSE/ESAonly).

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 Messages and Codes Guide* or the supplemental list in the following figure. 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 Messages and Codes Guide*.

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 Error Codes:

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

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
BECAUSEEXTRACT 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.

DRVRO01E 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.

DRVRO02E 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.

DRVRO03E ERROR CALLING module-name, NOT ENOUGH STORAGE TO LOADPROGRAM**Reason:**

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

Action:

Increase the region size and resubmit the job.

DRVRO04E 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 - GSSFILERETURN CODE IS n1,n2,n3,n4**Reason:**

The file handler is unable to open the file.

Action:

See the tables in Return Codes for an explanation and appropriate action for the return codes indicated.

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

Reason:

The file handler is unable to read the file.

Action:

See the tables in Return Codes for an explanation and appropriate action for the return codes indicated.

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

Reason:

The file handler is unable to write to the file.

Action:

See the tables in Return Codes for an explanation and appropriate action for the return codes indicated.

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

Reason:

The file handler is unable to update the file.

Action:

See the tables in Return Codes for an explanation and appropriate action for the return codes indicated.

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

Reason:

The file handler is unable to close the file.

Action:

See the tables in Return Codes 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.....nnnnnn

Reason:

An internal processing error has been detected by the message tableprocessor.

Action:

Contact Computer Associates Product Support.

MTCH001E ERROR FOUND BY STEP INITIALIZATION ROUTINE - USRSUBS

Reason:

An unrecoverable error condition has been detected by USRMTC.

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 WRITTENnumber-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 LDELRECORD**Reason:**

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

Action:

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

**MTCH009E COULD NOT CREATE FORMAT2 ALLOCATION EXTRACT RECORD
BECAUSEFORMAT1 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 THEINPUT 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 AuditReport 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 AuditReport 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 xxxxxxxxxxxxxxxxxxxxxxx**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 a 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 xxxxxxxxxxxxxxx

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 – GSS CALL 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 OCCURRED DURING LOAD OF xxxxxxxx – GSS LOAD RETURN CODE IS nn

Reason:

Module load processor error occurred.

Action:

Contact Computer Associates Product Support.

NKWP053E ERROR OCCURRED DURING WRITE OF xxxxxxxx – GSS PRINT 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 ASSGN 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 – GSS FILE RETURN CODE IS n1,n2,n3,n4

Reason:

The file handler is unable to open the file.

Action:

See the tables in Return Codes for an explanation and appropriate action for the indicated return codes.

NKWP065E ERROR OCCURRED DURING READ OF xxxxxxxx FILE – GSS FILE RETURN CODE IS n1,n2,n3,n4

Reason:

The file handler is unable to read the file.

Action:

See the tables in Return Codes for an explanation and appropriate action for the indicated return codes.

NKWP066E ERROR OCCURRED DURING CLOSE OF xxxxxxxx FILE – GSS FILE RETURN CODE IS n1,n2,n3,n4

Reason:

The file handler is unable to close the file.

Action:

See the tables in Return Codes 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:**

An 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 xxxxxxxxxxx

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 xxxxxxxxxxx

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 unrecoverable 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-datecompile-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
NOTCORRESPONDING 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 full words 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/REORGrelease-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 unrecoverable 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) IIXEXOL2) IIXSRKY
3) IIXEXTR MUST 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 Overview.

**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 ISerror-code n**Reason:**

An unrecoverable 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 z/OS or 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 inputfile 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 is not a valid CAIDMS/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:

- The specified set exists within only one of the subschemas.
- 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 STOP AFTER STEP CANNOT PRECEDE RESTART STEP**Reason:**

The step specified in the STOP AFTER parameter is processed before the step specified in the RESTART parameter.

Action:

Change the incorrect STOP AFTER 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 PAGE RESERVE parameter specifies a value that is too large.

Action:

Correct the PAGE RESERVE 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 unrecoverable 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 INFOROUTE or REPLYROUTE parameter were specified.

Action:

Specify another option for DSPLYINT or delete the INFOROUTE 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 BUTNOT 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 DIFFERSFROM 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 AVM/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 - NOTSUPPORTED**Reason:**

CA IDMS/DB Reorg does not currently support SQL databases.

Action:

Do not run an CA IDMS/DB Reorg against an SQL database.

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 BLOCKNUMyyy 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 unrecoverable 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 unrecoverable 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 GSSPRINT.

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-namedisposition
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 unrecoverable 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 z/OS or 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 unrecoverable 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 unrecoverable error condition has been detected.

Action:

Contact Computer Associates Product Support.

RORG006E ERROR IN SORT PROGRAM module-name INVALID SORT RECORD PARAMETER

Reason:

An unrecoverable 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 unrecoverable error condition has been detected.

Action:

Contact Computer Associates Product Support.

SECA001E ERROR FOUND BY STEP INITIALIZATION ROUTINE - USRSUBS

Reason:

An unrecoverable 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. The number of bytes targeted to the area is greater than the allocated space.

Action:

Correct this condition by creating adequate space in the new subschema for the area indicated by allocating more space to the area, by redistributing records to other areas, or by using a combination of these methods. The situation arises when attempting to allocate space for the named record above. This record cannot be fragmented.

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 ONINPUT RECORD. SAVE SNAP AND CALL SUPPORT

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 SNAP AND CALL SUPPORT

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 unrecoverable 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-records

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 unrecoverable 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 unrecoverable 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-1module-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 EXTRANEous 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 FILE COUNT 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.

WTOR001 =====> 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 WTOR001 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 WTOR001 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 ON REQUEST 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 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 ATEND OF STEP

Reason:

This message is displayed in response to a SET INTERVAL operator command.

Action:

None.

Return Codes

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.

n4 Return Codes:

n4	Reason	Action
4	End-of-file	Call Product Support.

n4	Reason	Action
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.
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.

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 overrode parm list	Remove DCB information from JCL and ensure that the correct files are referenced.
2	Parm list overrode JCL/label	Remove DCB information from JCL and ensure that the correct files are referenced.
3	Unrecognized request	Call Product Support.
4	z/OS or OS/390 x13 ABEND trapped at open	Fix cause for x13 ABEND.

n2	Reason	Action
5	Tried to update seq. file	Call Product Support.
6	VSAM write at other than load	Call Product Support.
7	SOS table could not expand	Call Product Support.
8	z/OS or 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	z/OS or 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 • 64, 118

C

CA IDMS/DB Reorg Output • 63
CA IDMS/DB Reorg Processing--VM/ESA
 Environment • 99
CA IDMS/DB Reorg Processing--VSE/ESA
 Environment • 107
CA IDMS/DB Reorg Processing--z/OS or OS/390
 Environment • 92
CA IDMS/DB Reorg Reorganization Capabilities • 51
CA IDMS/DB Reorg's Inputs • 44
CA IDMS/DB Reorg's Output • 19
CA IDMS/DB Reorg's Record Allocation Techniques •
 45
Condition Codes • 117
Console Communications Facility • 41

D

Database Access • 75
DBKEYS Interface File • 73

H

How CA IDMS/DB Reorg Performs Reorganization •
 15

I

Improve System and CA IDMS/DB Performance • 15

M

Message Text • 118

N

NOSWEEP Statement • 40

O

Operational Flexibility During Reorganization • 18
Operational Flow and Considerations • 75

P

Parameters and Their Functions • 21

PROCESS Statement • 23

R

Reorganization Steps • 57

S

Select Any Area or Entire Database • 13
Status Display • 73
syntax diagram • 9

U

Understanding CA IDMS/DB Reorg Processing • 43
Understanding Syntax Diagrams • 9

V

VSE/ESA File Processing Alternate Method • 108

W

Why Reorganize a Database? • 13