CA IDMS - 19.0
Using IDMS/DB Reorg

Date: 15-Jan-2018
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 © 2017 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.
# Table of Contents

## Reorganize a Database

- Select Area or Entire Database ............................................................................. 8
  - Updates Set Linkage Automatically .................................................................... 9
  - Leaves Relationships and Records Unaltered ...................................................... 9
  - Recognizes SR8 Index Areas ............................................................................... 9
- Improve System and CA IDMS DB Performance .................................................. 10
- How CA IDMS/DB Reorg Performs Reorg .............................................................. 10
  - Reorganization Controlled by Parameter ............................................................ 11
  - Reorganization Controlled by Subschema and DMCL ......................................... 12
  - Reorganization Controlled by Internal Logic ....................................................... 13
- Operational Flexibility During Reorg .................................................................... 13
  - STOPAFTER and RESTART Parameters ............................................................. 13
  - Console Communication Facility ........................................................................ 14
- CA IDMS/DB Reorg Output .................................................................................... 14
  - Audit Report Designed for Operations Flexibility .............................................. 14
  - CA IDMS/DB Reorg Interface Capabilities ......................................................... 15

## Reorg Parameters and Commands

- Parameters Statements Control .......................................................................... 15
  - PROCESS Statement Overview ........................................................................... 16
  - COPY Statement Overview ................................................................................ 16
  - NOSWEEP Statement Overview ........................................................................ 16
- The PROCESS Statement ...................................................................................... 17
  - PROCESS Parameter ........................................................................................... 19
  - OLDSUB Parameter ............................................................................................ 19
  - NEWSUB Parameter ........................................................................................... 20
  - OLDDMCL Parameter .......................................................................................... 20
  - NEWDMCL Parameter ........................................................................................ 20
  - OLDDBN Parameter ............................................................................................. 20
  - NEWDBN Parameter ............................................................................................ 21
  - RPT Parameter ..................................................................................................... 21
  - RESTART Parameter ............................................................................................ 22
  - STOPAFTER Parameter ....................................................................................... 23
    - Using RESTART with STOPAFTER ................................................................. 23
  - DSPLYINT Parameter .......................................................................................... 24
VSE/ESA File Processing Alternate Method ................................................................. 90
Process 1--Defining the New Areas ........................................................................... 90
Process 2--Information Gathering ............................................................................. 90
Process 3--Allocating Work Files ............................................................................ 92
Process 4--Reorganizing the Database ..................................................................... 93
Using IDMS/DB Reorg

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.

For more information, see the following topics:

- Reorganize a Database (see page 8)
- Reorg Parameters and Commands (see page 15)
- Understand CA IDMS/DB Reorg Processing (see page 33)
- CA IDMS/DB Reorg System Output (see page 52)
- IDMS/DB Reorg Operations (see page 62)

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. For more information, see the following topics:

- Select Area or Entire Database (see page 8)
- Improve System and CA IDMS DB Performance (see page 10)
- How CA IDMS/DB Reorg Performs Reorg (see page 10)
- Operational Flexibility During Reorg (see page 13)
- CA IDMS/DB Reorg Output (see page 14)

Select 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 8) section.

- Updates Set Linkage Automatically (see page 9)
- Leaves Relationships and Records Unaltered (see page 9)
- Recognizes SR8 Index Areas (see page 9)
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 Reorg

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.

- Reorganization Controlled by Parameter (see page 11)
- Reorganization Controlled by Subschema and DMCL (see page 12)
- Reorganization Controlled by Internal Logic (see page 13)

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 Administering section and Administrating section.

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 GSDTPARM 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 Parameter

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 page size
- 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.
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.

See the following topics for more information:

Operational Flexibility During Reorg

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 (see page 13)
- Console Communication Facility (see page 14)

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 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 (see page 14)
- CA IDMS/DB Reorg Interface Capabilities (see page 15)

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 on 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.

Reorg Parameters and Commands

This section 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.

See the following topics for more information:
- Parameters Statements Control (see page 15)
- The PROCESS Statement (see page 17)
- The COPY Statement (see page 27)
- NOSWEEP (see page 30)
- Console Communications Facility (see page 31)

Parameters Statements Control

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 (see page 16)
- COPY Statement Overview (see page 16)
- NOSWEEP Statement Overview (see page 16)
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:
The PROCESS Statement

OLDDMCL = dmcl-name, NEWDMCL = dmcl-name

,OLDDBN = dbname ] [ ,NEWDBN = dbname ]

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

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

,CURRENT
ANALYSIS
OPTALLOC
PRIALLOC
SECALLOC
MATCH
IIXSDBK
UPDLINK
STORE
,EOR

,STOPAFTER = [ ,DSPLYINT = [interval ]
NO
REQUEST
STEP

,[INFOROUTE = route-code ]
,REPLYROUTE ]

[,TSOUSERS = {user-id1, user-id2, ... user-id10} ]

,[BLOCKNUM = block-size ]

,SORTFLAG = {ALL, CRITICAL, NONE }

,SORTSIZE = number - of - k

,COPY,AREA = area-name

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

,[TARGETROOT ]

,[PAGERESERVE = number - of - bytes ]

,[NOSWEEP, AREA = area-name ]

NOSWEEP Statement Overview
The PROCESS statement is required; it must be the first statement entered.

- PROCESS Parameter (see page 19)
- OLDSUB Parameter (see page 19)
- NEWSUB Parameter (see page 20)
- OLDDMCL Parameter (see page 20)
- NEWDMCL Parameter (see page 20)
- OLDDBN Parameter (see page 20)
- NEWDBN Parameter (see page 21)
- RPT Parameter (see page 21)
- RESTART Parameter (see page 22)
- STOPAFTER Parameter (see page 23)
  - Using RESTART with STOPAFTER (see page 23)
- DSPLYINT Parameter (see page 24)
- TSOUSERS Parameter (see page 26)
- SORTFLAG Parameter (see page 27)
- SORTSIZE Parameter (see page 27)
- BLOCKNUM Parameter (see page 27)

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 Statement

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

RPT = {NEXT
       OPTALLOC
       PRIALLOC
       SECALLOC
       MATCH
       IIIXSDKB
       UPDLINK
       STORE
       }

RESTART = {NEXT
           OPTALLOC
           PRIALLOC
           SECALLOC
           MATCH
           IIIXSDKB
           UPDLINK
           STORE
           }

STOPAFTER = {CURRENT
              ANALYSIS
              OPTALLOC
              PRIALLOC
              SECALLOC
              MATCH
              IIIXSDKB
              UPDLINK
              STORE
              EOR

              }

DSPLYINT = {interval
             NO
             REQUEST
             STEP

              }

INFOROUTE = {route-code
              [REPLYROUTE]

              }

TSOUSERS = {user-id1, user-id2, ..., user-id10}

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
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**

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 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**

**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.

- **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.

**OLDDBN Parameter**
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, iixsd ]

[ ,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.

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 on 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 (IXSDBK 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.
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, SECALLOC, OPTALLOC, PRIALLO, 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.

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

Using RESTART with STOPAFTER

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

\[
\text{[D}S\text{PSLYINT} = \{\text{interval} \quad \text{NO} \quad \text{REQUEST} \quad \text{STEP}\}\]

- \[
\text{[D}S\text{PSLYINT} = \{\text{display interval}\}\]

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:

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

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.

**The 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).

- Copy Parameter (see page 28)
- AREA Parameter (see page 28)
- PRIORITY Parameter (see page 29)
- PAGERESERVE Parameter (see page 30)
- TARGETROOT Parameter (see page 30)
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:**

\[
\text{COPY AREA=area-name} \quad \text{PRIORITY=}\{\text{DIRECT CALC record-name sublist-of-record-names}\} \quad \text{TARGETROOT} \quad \text{PAGERESERVE=number-of-bytes}
\]

**COPY Statement**

**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 (sublist-of-record names)]) CALC record-name]
```

- **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.

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

- \[\text{PAGERESERVE=number-of-bytes}\]
  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

- \[\text{TARGETROOT}\]
  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.

NOSWEEP

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

The NOSWEEP parameter decreases CA IDMS/DB Reorg execution time when it is used for non-copy areas that contain few set occurrences with cross-area linkage. When NOSWEEP is used, CA IDMS/DB Reorg 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 NOSWEEP

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

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

NOSWEEP Syntax and Options

\[ NOSWEEP, \text{AREA=}area-name \]

- **NOSWEEP**
  Initiates NOSWEEP 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 NOSWEEP statement can only be specified once for a given area.
- The *area-name* specified must not contain records that are linked (as members) to an SR8 set in a COPY area.

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 Command (see page 32)**
- **SET INTERVAL Command (see page 32)**
- **STOP Command (see page 32)**
- **CANCEL Command (see page 33)**

### 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.

Understand CA IDMS/DB Reorg Processing

This section 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 section discusses CA IDMS/DB Reorg’s inputs and record allocation techniques that improve system performance.

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 on CA IDMS/DB Reorg:
- CA IDMS/DB Reorg Inputs (see page 34)
- CA IDMS/DB Reorg Record Allocation Techniques (see page 35)
- CA IDMS/DB Reorg Reorg Capabilities (see page 41)
- Reorg Steps (see page 47)

CA IDMS/DB Reorg 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 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 (see page 36)
- Reduce Fragmented Records (see page 36)
- Allocating DIRECT Records (see page 36)
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 variable length CALC record does not fit on the target page, but the target page can accommodate the root portion and TARGETROOT parameter was specified, CA IDMS/DB Reorg 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 first available 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:
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.

![Diagram](image1)

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

![Diagram](image2)

Sorted VIA Records
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:
CA IDMS/DB Reorg Reorg Capabilities

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

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))

Allocation and Page Reserve
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 stored. 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:
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:

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

\[ \text{PRIORITY=(A,C,B)} \]

Advantage CA-IDMS/DB Reorg allocates the records this way:
• **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:

![Diagram showing the change in page ranges of two areas](image)

**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 CA IDMS/DB Reorg has the capabilities to:

Minimize fragmented records. CA IDMS/DB Reorg does not fragment records unless there is not a page available with enough free space to accommodate the entire record.

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:
Reorg 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.

- Functions of the Reorganization Steps (see page 50)

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:**
Reorganization Steps
Functions of the Reorganization Steps

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

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

- **OPTALLOC (Optimum Allocation/Pointer Extract) Step**—Determines the optimum target page allocation for every record in the areas to be reorganized. Processing in OPTALLOC allocates records to pages as if a page could hold every record that targeted to that page. CA IDMS/DB Reorg then creates pointer extracts for set linkages with records in the areas selected by using the COPY statement. If a record is indexed and the index is maintained in db-key sequence, the extract is flagged for later special processing by the MATCH step and the IIXSDBK step.

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

- **SECALLOC (Secondary Allocation) Step**—Allocates real db-keys for VIA records that do not target to the owner page, and for records that were written to the UNALLOC file. For variable length records that cannot fit as a whole on any page in that record's page range, the SECALLOC Step assigns db-keys to record fragments. If an integrated index (SR8) resides in an area being copied, the program:
  - 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:*

<table>
<thead>
<tr>
<th>Step/Function</th>
<th>Allocates Records to Area Specified in COPY Statement</th>
<th>Updates Set Linkage</th>
<th>Accesses Old Database</th>
<th>Access New Database</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step/ Function</td>
<td>Allocates Records to Area Specified in COPY Statement</td>
<td>Updates Set Linkage</td>
<td>Accesses Old Database</td>
<td>Access New Database</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------------------</td>
<td>---------------------</td>
<td>-----------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>OPTALLO C Step</td>
<td>Computes optimum target page.</td>
<td>Extracts pointers which will change. If copying area with IIX set, writes individual bottom level entries to SECOPTA.</td>
<td>Reads records being copied and those with set linkage.</td>
<td></td>
</tr>
<tr>
<td>PRIALLOC Step</td>
<td>Allocates DIRECT, CALC, and VIAs stored on owner's page.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SECALLOC C Step</td>
<td>Allocates CALC, DIRECT, VIA overflows, fragments, VIAs not stored on owner's page and SR8 records. Rebuilds IIX tables to create new SR8 structures.</td>
<td>Extracts pointers for all SR8 set linkage.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATCH Step</td>
<td>Replaces old db-key in pointer extracts with new db-keys from allocations.</td>
<td>Replaces old db-key in pointer extracts with new db-keys from allocations.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IIXSDBK Step</td>
<td>Reorders IIX set sorted by db-key into proper sequence of db-key.</td>
<td>Reorders IIX set sorted by db-key into proper sequence of db-key.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPDLINK Step</td>
<td>Updates pointer prefix or IXDET table entry with new db-key.</td>
<td></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>STORE Step</td>
<td>Populate areas being copied.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **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.

## CA IDMS/DB Reorg System Output

This section is a section 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.

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 CA IDMS/DB Reorg steps and activities by providing three types of messages:

- Parameter Messages
- Processing Messages
- Restart Information Messages

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

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

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

### 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 on 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 section 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.

**CA IDMS/DB Reorg 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.

- Audit Subreport--Area Recap for Old Subschema (see page 55)
- Audit Subreport--Set Linkage Recap for Area (see page 56)
- Restart Information Messages (see page 58)

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

<table>
<thead>
<tr>
<th>Report Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Title</td>
<td>The title of the report.</td>
</tr>
<tr>
<td>UR</td>
<td>The tape from which CA IDMS/DB Reorg was installed.</td>
</tr>
<tr>
<td>RELEASE</td>
<td>The version number of CA IDMS/DB Reorg that has been installed.</td>
</tr>
<tr>
<td>Parameter Messages</td>
<td>Lists all input parameters and convey information on 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).</td>
</tr>
<tr>
<td>Processing Messages</td>
<td>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. 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.</td>
</tr>
</tbody>
</table>
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 on specific messages.

Audit Report - Processing Parameters Step:

<table>
<thead>
<tr>
<th>ID</th>
<th>RELEASE</th>
<th>DATE</th>
<th>CA-IDS</th>
<th>TIME</th>
<th>PAGE</th>
<th>mm/dd</th>
</tr>
</thead>
<tbody>
<tr>
<td>/DB REORG</td>
<td>Rnn.nn</td>
<td>DATE</td>
<td>AUDIT REPORT</td>
<td>mm/dd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/yy</td>
<td>hh:mm:ss</td>
<td>nnnn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AUDT001I STEP: PARM, STARTED

3-----4-----5-----6-----7-V

V-----1-----2-----

NKWP091I INPUT PARAMETER STATEMENT
NEW=DBR18ALL,ST=UPDLINK
PARM043I NO PARAMETER ERRORS DETECTED

AUDT002I STEP: PARM, ENDED, CC= 0000

Audit Report - OPTALLOC Step:

<table>
<thead>
<tr>
<th>ID</th>
<th>RELEASE</th>
<th>DATE</th>
<th>CA-IDS</th>
<th>TIME</th>
<th>PAGE</th>
<th>mm/dd</th>
</tr>
</thead>
<tbody>
<tr>
<td>/DB REORG</td>
<td>Rnn.nn</td>
<td>DATE</td>
<td>AUDIT REPORT</td>
<td>mm/dd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/yy</td>
<td>hh:mm:ss</td>
<td>nnnn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AUDT001I STEP: OPTALLOC, STARTED

3-----4-----5-----6-----7-V

V-----1-----2-----

OPTX001I OBTAINS FOR AREA SWEEP................................................245
OPTX002I OBTAINS FOR SET WALK..................................................142
OPTX087I EXTRACT RECORDS WRITTEN...............................................817
OPTX088I EXTRACT RECORDS WRITTEN................................................39
OPTX090I SECOPTA RECORDS WRITTEN................................................34
OPTX013I NUMBER OF IIX SORTED DBKEYS AT BOTTOM LEVEL ........................0
OPTX014I NUMBER OF IIXSRKY RECORDS THAT WILL BE CREATED....................0
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:

<table>
<thead>
<tr>
<th>Report Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subreport Heading</td>
<td>The title of the subreport.</td>
</tr>
<tr>
<td>AREA</td>
<td>A list of the names of all the areas in the old subschema.</td>
</tr>
<tr>
<td>NBR-PAGES</td>
<td>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.</td>
</tr>
<tr>
<td>PROCESSING-TYPE</td>
<td>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: 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.</td>
</tr>
</tbody>
</table>

Audit Subreport--Area Recap for Old Subschema:

<table>
<thead>
<tr>
<th>ID /DB REORG</th>
<th>RELEASE</th>
<th>DATE</th>
<th>CA-IDMS</th>
<th>TIME</th>
<th>PAGE</th>
<th>mm/dd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rnn.nn</td>
<td>nnnn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>hh:mm:ss</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AUDT001I STEP: ANALYSIS, STARTED DATE: mm/dd/yy TIME: hh:mm:ss

AREA Recap for Old Subschema

- - - - - - - - - - - - AREA - - - - - - - - - - - -

REGION 30 NOT ACCESSED
REGION 30 NOT ACCESSED

B0GUS-1
B0GUS-2

15-Jan-2018 55/95
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 section for instructions on how to calculate file sizes.

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

<table>
<thead>
<tr>
<th>Report Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subreport Heading</td>
<td>The title of the subreport.</td>
</tr>
<tr>
<td>area-name</td>
<td>The name of the area for which set linkage is reported.</td>
</tr>
<tr>
<td>record-type</td>
<td>The name of a record in the area. Each record is separated by a line of dashes.</td>
</tr>
<tr>
<td>nn</td>
<td>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.</td>
</tr>
<tr>
<td>location</td>
<td>The location mode of the record in the old database and in the new database.</td>
</tr>
<tr>
<td>page range</td>
<td>The page range for the record in the old database and in the new database.</td>
</tr>
<tr>
<td>OWN/MEM</td>
<td>Indicates whether the record is an owner or a member of the set in the SET NAME field.</td>
</tr>
<tr>
<td>SET-NAME</td>
<td>The name of the set.</td>
</tr>
</tbody>
</table>

The following table provides a summary of the details from the Set Linkage Recap for Area:

<table>
<thead>
<tr>
<th>REGION</th>
<th>30</th>
<th>NOT ACCESSED</th>
<th>BOGUS-3-</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGION</td>
<td>30</td>
<td>COPY</td>
<td>CLASS-</td>
</tr>
<tr>
<td>REGION</td>
<td>15</td>
<td>COPY</td>
<td>DEPT-</td>
</tr>
<tr>
<td>REGION</td>
<td>15</td>
<td>COPY</td>
<td>LOC-</td>
</tr>
<tr>
<td>REGION</td>
<td>20</td>
<td>SWEEP</td>
<td>MISC1-</td>
</tr>
<tr>
<td>REGION</td>
<td>20</td>
<td>COPY</td>
<td>MISC2-</td>
</tr>
<tr>
<td>REGION</td>
<td>80</td>
<td>COPY</td>
<td>STUDENT-</td>
</tr>
</tbody>
</table>
### Report Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET-POINTERS</td>
<td>Indicates types of pointers in the set: NEXT, PRIOR, or OWNER.</td>
</tr>
<tr>
<td>POINTE</td>
<td>Indicates one of three methods CA IDMS/DB Reorg uses to create extracts: AREA SWEEP, SET WALKING, or USING NEXT AND PRIOR.</td>
</tr>
<tr>
<td>NEW VIA-OWNER</td>
<td>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).</td>
</tr>
<tr>
<td>LOC-MODE</td>
<td>The location mode of the record type listed under NEW VIA-OWNER.</td>
</tr>
</tbody>
</table>

### 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:

<table>
<thead>
<tr>
<th>ID/DB REORG/yy</th>
<th>RELEASE/nn.nn</th>
<th>DATE/hh:mm:ss</th>
<th>CA-IDMS/DATE</th>
<th>TIME/MM/SS</th>
<th>PAGE/nnnn</th>
<th>mm/dd</th>
</tr>
</thead>
</table>

**SET LINKAGE RECAP FOR AREA CLASS-REGION**

```
08106 CLASS...... RECORD  14 EXTRACTS
/ RECORD (AVG) OLD LOCATION MODE OF VIA PAGE RANGE OF 89,001 89,030
NEW LOCATION MODE OF VIA PAGE RANGE OF 89,001 89,030
OWN/MEM ---SET-NAME---- ---SET-POINTER-- 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 PAGE RANGE OF 89,001 89,030
NEW LOCATION MODE OF VIA PAGE RANGE OF 89,001 89,030
OWN/MEM ---SET-NAME--- ---SET-POINTER-- POINTER-EXTRACTS-CREATED-
```
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.
CA IDMS - 19.0

- 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:**

<table>
<thead>
<tr>
<th>ID</th>
<th>RELEASE</th>
<th>DATE</th>
<th>CA-IDMS</th>
<th>TIME</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>/DB REORG</td>
<td>Rnn.nn</td>
<td>hh:mm:ss</td>
<td>nnnn</td>
<td>mm/dd</td>
<td></td>
</tr>
</tbody>
</table>

**RESTART STATUS INFORMATION**

PRST001I RESTART DATA....... OLD SUBSCHEMA ..... DBR18ALL
NEW SUBSCHEMA .... DBR18ALL

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

OPTALLOC............ STARTED mm/dd/yy AT hh:mm:ss

PRIALLOC............ STARTED mm/dd/yy AT hh:mm:ss

SECALLOC............ STARTED mm/dd/yy AT hh:mm:ss

MATCH.............. STARTED mm/dd/yy AT hh:mm:ss

ENDED mm/dd/yy AT hh:mm:ss

15-Jan-2018 59/95
PRST002I RESTART WITH ...... STORE

PRST003I DATASET DISPOSITION FOR RESTART.....

<table>
<thead>
<tr>
<th>Dataset</th>
<th>OLD KEEP</th>
<th>Created/Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTRL1</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>CNTRL2</td>
<td>OLD KEEP</td>
<td>UPDATED</td>
</tr>
<tr>
<td>EXTRACT</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>PRIOPTA</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>SECOPTA</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>PAGUTIL</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>UNALLOC</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>PRIREAL</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>SECREAL</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>ALLOCX</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>IIXSRKY</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>IIXEXOL</td>
<td>OLD KEEP</td>
<td>CREATE</td>
</tr>
<tr>
<td>IIXEXTR</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>UPDLINK</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
</tbody>
</table>

PRST003I DATASET DISPOSITION FOR RESTART.....

<table>
<thead>
<tr>
<th>Dataset</th>
<th>OLD KEEP</th>
<th>Created/Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTRL1</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>CNTRL2</td>
<td>OLD KEEP</td>
<td>UPDATED</td>
</tr>
<tr>
<td>EXTRACT</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>PRIOPTA</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>SECOPTA</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>PAGUTIL</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>UNALLOC</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>PRIREAL</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>SECREAL</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>ALLOCX</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>IIXSRKY</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>IIXEXOL</td>
<td>OLD KEEP</td>
<td>CREATE</td>
</tr>
<tr>
<td>IIXEXTR</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
<tr>
<td>UPDLINK</td>
<td>OLD KEEP</td>
<td>CREATED</td>
</tr>
</tbody>
</table>
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
```

Sample Status Display - PRIALLOC Step.

DBKEYS Interface File

The DBKEYS file is created when the DDname DBKEYS is specified in 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.

```
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Element Description</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLDDBKEY</td>
<td>PIC 9(8) COMP SYNC</td>
<td>Old db-key</td>
</tr>
<tr>
<td>NEWDBKEY</td>
<td>PIC 9(8) COMP SYNC</td>
<td>New db-key</td>
</tr>
<tr>
<td>SR3DBKEY</td>
<td>PIC 9(8) COMP SYNC</td>
<td>SR3 db-key. See note.</td>
</tr>
<tr>
<td>RECORDID</td>
<td>PIC 9(4) COMP</td>
<td>Record id</td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Element Description</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR3RECID</td>
<td>PIC 9(4) COMP</td>
<td>Original record id. True, only if the record was originally a SR2</td>
</tr>
</tbody>
</table>

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

**IDMS/DB Reorg Operations**

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

**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.

For more information, see the following topics:
- CA IDMS/DB Reorg Database Access (see page 62)
- CA IDMS/DB Reorg Operational Flow and Considerations (see page 62)
- CA IDMS/DB Reorg Processing z/OS or OS/390 Environment (see page 79)
- CA IDMS/DB Reorg Processing VM/ESA Environment (see page 84)
- CA IDMS/DB Reorg Processing VSE/ESA Environment (see page 89)

**CA IDMS/DB Reorg 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.

**CA IDMS/DB Reorg 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.

Contents

- Integrated Indexes (see page 63)
- File Space Not Needed (see page 64)
- Allocating Space for Work Files and Sort Work Areas (see page 64)
- Work Files (see page 64)
  - ANALYSIS Step (see page 64)
  - OPTALLOCP Step (see page 66)
  - PRIALLOC Step (see page 67)
  - SECALLOC Step (see page 69)
  - MATCH Step (see page 72)
  - IIXSDBK Step (see page 74)
  - UPDLINK Step (see page 75)
  - STORE Step (see page 76)
- Sort Work Areas (see page 77)
- Summary of Procedures for Operating CA IDMS/DB Reorg (see page 77)

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 SECIX2.

- If you are updating areas containing integrated index sets sorted by db-key, after the OPTALLOCP 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:**
ANALYSIS Step

Estimating Space for Work Files (CNTRL1 CNTRL2):

<table>
<thead>
<tr>
<th>Work File(s)</th>
<th>REC</th>
<th>LRECL</th>
<th>BLKSIZE</th>
<th>Space Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB</td>
<td>6148</td>
<td>6152</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.

**OPTALLOC Step**

*Work File Usage For OPTALLOC Step:*

- Optimal allocation step
- Work file usage
- Estimating space for work files (EXTRACT, PRIOPTA, SECOPTA)

![Diagram showing work file usage for OPTALLOC step]
### Work File Usage For PRIALLOC Step:

<table>
<thead>
<tr>
<th>Work File(s) FM</th>
<th>BLKSIZE</th>
<th>Space Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EXTR FB 22</strong></td>
<td>6138</td>
<td># of EXTRACT records = ((# \text{ of EXTRACT records/record type}) \times \text{(# of occurrences of that record type)}). 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.</td>
</tr>
<tr>
<td><strong>PRIOP FB 23 + length of longest CALC key in copy area</strong></td>
<td>(CALCs + DIRECTs + VIAs) whose page range is the same as their owner’s page range (see note). 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). Note: Refer to Audit Subreport--Set Linkage Recap for Area for additional information.</td>
<td></td>
</tr>
<tr>
<td><strong>SECO FB</strong></td>
<td>The greater of 34 or (34 + length of longest integrated index symbolic key) (See message NLYZ012I in ANALYSIS Step).</td>
<td># of SECOPTA records = ((# \text{ of VIAs not in the same page range as the owner}) + \text{(# of all BL SR8 records)}).</td>
</tr>
</tbody>
</table>

**PRIALLOC Step**

**Work File Usage For PRIALLOC Step:**
### PRIALLOC Step

#### Estimating Space for Work Files (PRIREAL, UNALLOC, PAGUTIL):

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

### Work File(s) REC LRECL BLKSIZE Space Allocation

<table>
<thead>
<tr>
<th>Work File(s)</th>
<th>REC</th>
<th>LRECL</th>
<th>BLKSIZE</th>
<th>Space Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+ (pages-in-copy-area-nn/1020)) Note: 'pages-in-copy-area' refers to the new subschema.</td>
</tr>
</tbody>
</table>

**SECALLOC Step**

**Work File Usage for SECALLOC Step:**
SECALLOC Step

Estimating Space for Work Files (SECREAL):

<table>
<thead>
<tr>
<th>Work File(s)</th>
<th>REC FM</th>
<th>LRECL</th>
<th>BLFSIZE</th>
<th>Space Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECREAL</td>
<td>FB</td>
<td>22</td>
<td>6138</td>
<td># of SECREAL records = (# of SECOPTA records) + (# of UNALLOC records) + (# of SR8 records created) + (# of entries in the BL SR8) + (# of entries for UL SR8's - sets sorted by DBKEY only)</td>
</tr>
</tbody>
</table>

Note: This calculation must be done for each indexed set in the area.
BL - Bottom Level
UL - Upper Level

Recap
Number of records for each indexed set in the area =
number of SECOPTA records
+ number of UNALLOC records
+ number of 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, SECII2):

<table>
<thead>
<tr>
<th>Work File(s)</th>
<th>REC FM</th>
<th>LRECL</th>
<th>BLKSIZE</th>
<th>Space Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECII X2</td>
<td>VB</td>
<td>12+length of largest possible SR8 record.</td>
<td>Either LRECL+4 or 6K, whichever is larger.</td>
<td>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.) The amount of space is listed as &quot;Total Bytes to Store Index&quot; on the CA IDMS/DB Analyzer SR8 Index Report, SR8 Index Statistics Section.</td>
</tr>
</tbody>
</table>
| SECAL X2     | VB     | 18    | 6138    | The space required is no greater than the space for all existing upper and bottom level SR8 records. A more detailed calculation follows:
# of SECALX2 records = (3 * the # of BL SR8 records) + (3 * UL SR8 record) + (1 * # of UL SR8 entries).
where:
UL SR8 entries = (# of SR8 records - 1)
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. |

BL - Bottom Level SR8
UL - Upper Level SR8

For SECREAL and SECALX2 (optional detailed calculation)

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

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:

<table>
<thead>
<tr>
<th>Bottom level SR8 records</th>
<th>2100</th>
<th>10</th>
<th>= 210</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper level SR8 records</td>
<td>210</td>
<td>10</td>
<td>= 21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>= 3   (rounded up)</td>
</tr>
<tr>
<td>(Top level SR8 record)</td>
<td>3</td>
<td>10</td>
<td>= 1   (rounded up)</td>
</tr>
</tbody>
</table>

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:**
### MATCH Step

#### Estimating Space for Work Files (ALLOCX, IIXEXOL):

<table>
<thead>
<tr>
<th>Work File REC (s)</th>
<th>REC FM</th>
<th>LRECL</th>
<th>BLKSIZE</th>
<th>Space Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLOCX</td>
<td>FB</td>
<td>18</td>
<td>6138</td>
<td># of ALLOCX records = (# of EXTRACT records) + (# of PRIREAL records) + (# of SECREAL records).</td>
</tr>
<tr>
<td>IIXEXOL</td>
<td>FB</td>
<td>20</td>
<td>6140</td>
<td>See note. # of IIXEXOL records = (3) + (2 * BL SR8 records) + (1 * UL SR8 records) Note: This file is required only if the area being copied contains an indexed set sorted by DB-Key.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BL - Bottom Level SR8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UL - Upper Level SR8</td>
</tr>
</tbody>
</table>

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

Work File Usage for IIXSDBK Step:

- **Control Files**
  - IIXRKY
  - IIXEXTR

- **IIXSDBK**

Estimating Space for Work Files (IIXRKY IIXEXTR):

<table>
<thead>
<tr>
<th>Work File</th>
<th>REC FM</th>
<th>LRECL</th>
<th>BLKSIZE</th>
<th>Space Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIXRKY</td>
<td>F</td>
<td>20</td>
<td>6140</td>
<td>The number of records is listed in message OPTX014I.</td>
</tr>
<tr>
<td>IIXEXTR</td>
<td>FB</td>
<td>18</td>
<td>6138</td>
<td># of IIXEXTR records = (3 + (2 * BL SR8 record) + (1 * UL SR8 records)</td>
</tr>
</tbody>
</table>

**Diagram:**

- IIXSDBK Step
- Work File Usage for IIXSDBK Step:
  - IIXRKY
  - IIXEXTR
  - Control Files
  - IIXEXOL
**UPDLINK Step**

### Work File Usage for UPDLINK Step:

![Diagram of UPDLINK Step]

**Estimating Space for Work File (DBREC, DBKEY):**

<table>
<thead>
<tr>
<th>Work File(s)</th>
<th>REC</th>
<th>LRECL</th>
<th>BLKSIZE</th>
<th>Space Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBREC</td>
<td>VB</td>
<td>24+length of the longest record that fits on a single page in a copy area.</td>
<td>Either LRECL+4 or 6K, whichever is larger.</td>
<td># of DBREC record = (((24 * # of occurrences) + prefix + data) record-01) + &amp;vellip.</td>
</tr>
</tbody>
</table>
### Store Step

**Work File Usage For Store Step:**

The table below shows the space allocation calculation for the Store step:

<table>
<thead>
<tr>
<th>Work File(s)</th>
<th>REC</th>
<th>LRECL</th>
<th>BLKSIZE</th>
<th>Space Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$(((24 \times # \text{ of occurrences})$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$+ \text{ prefix + data record-nn})$</td>
</tr>
</tbody>
</table>

For the DBKEY FB 16 block, the BLOCK- NUM parameter is calculated as follows:

- Number of DBKEY records $= \left( \# \text{ of records in reorganized areas} - \right.$
- $(\text{SR8 records}) \times 16$

This calculation helps in determining the required space for the 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 on estimating the space allocation for each file in the following figures.

**Sort Work Areas:**

<table>
<thead>
<tr>
<th>Step</th>
<th>Sort Input File(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIALLOC</td>
<td>PRIOPTA</td>
</tr>
<tr>
<td>SECALLOC</td>
<td>SECOPTA PLUS UNALLOC</td>
</tr>
<tr>
<td>MATCH</td>
<td>EXTRACT PLUS PRIREAL PLUS SECREAL</td>
</tr>
<tr>
<td>IIIXDBK</td>
<td>IIIXEXOL</td>
</tr>
<tr>
<td>UPDLINK</td>
<td>ALLOCX PLUS IIIXEXTR PLUS SECALX2 PLUS SECIIX2</td>
</tr>
<tr>
<td>STORE</td>
<td>DBREC</td>
</tr>
</tbody>
</table>

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.


   PROCESS,
   OLDSUB=subschema-name,
   NEWSUB=subschema-name,
5. First PROCESS stops here.

6. Allocate space for work files. OPTALLOC Step.
   If sufficient disk space is not available for any file, it may be assigned to tape (except
   PAGUTIL). Use information from CA IDMS/DB Reorg Audit Report, from reports by CA IDMS
   /DB Analyzer. 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=ANALYSIS

   COPY,
       AREA= area-name

7. Second Process stops here.

8. 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


10. 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:]

11. Run CA IDMS/DB Reorg in Step mode.

12. All steps prior to the STORE step must specify the 'olddmcl' in the SYSIDMS parameters.

13. 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.

- Process 1--Defining the New Areas (see page 79)
- Process 2--Gathering Information (see page 80)
- Process 3--Allocating Work Files (see page 81)
- Process 4--Reorganizing the Database (see page 82)

Target or Distribution source library members USREXEC1, USREXEC2, USREXEC3, and USREXEC4 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 CA IDMS/DB Reorg, examines user parameters and subschemas, stops CA IDMS/DB Reorg after the ANALYSIS step, and produces the Audit/Analysis Report. In the control parameters which follow the JCL, it is necessary to include a COPY statement for each area that is to be reorganized.

```plaintext
//* 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 *
// SYSIPT DD PROCESS, OLDSUB=subschema-name,
// NEWSUB=subschema-name,
// OLDDMCL=dmcl-name,
// NEWDMCL=dmcl-name,
// STOPAFTER=ANALYSIS
// COPY, AREA=area-name &invellip.
/*

Key for Sample JCL to Obtain AUDIT/ANALYSIS Report:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>your.loadlib</td>
<td>The dataset name of the z/OS or OS/390 PDS library into which CA IDMS/DB Reorg was installed.</td>
</tr>
<tr>
<td>idms.loadlib</td>
<td>The dataset name of the z/OS or OS/390 PDS library in which your CA IDMS modules reside. You must supply a separate statement for every OS/390 PDS library needed.</td>
</tr>
<tr>
<td>dbmsdds</td>
<td>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.</td>
</tr>
<tr>
<td>work.files</td>
<td>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 DNAME.)</td>
</tr>
<tr>
<td>volser</td>
<td>The volume serial number of the disk or tape to be used for the control work files.</td>
</tr>
<tr>
<td>disk</td>
<td>An appropriate unit designation for the work file.</td>
</tr>
<tr>
<td>number.tracks</td>
<td>The number of tracks needed for the control work files if they reside on disk (3 cylinders on a 3350 are usually sufficient).</td>
</tr>
<tr>
<td>a</td>
<td>The appropriate SYSOUT class for your installation.</td>
</tr>
<tr>
<td>olldmcl</td>
<td>DMCL of the database prior to reorganization.</td>
</tr>
<tr>
<td>subschem</td>
<td>The subschema name.</td>
</tr>
<tr>
<td>a-name</td>
<td>The area name.</td>
</tr>
</tbody>
</table>
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.

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

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

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

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.

```jcl
/*
//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)
```
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>The appropriate SYSOUT class for your installation.</td>
</tr>
<tr>
<td>vsam. work. space</td>
<td>The dataset name of the one VSAM file to be used by CA IDMS/DB Reorg.</td>
</tr>
<tr>
<td>record. numbers</td>
<td>The number of records to be allocated in the VSAM or BDAM work space.</td>
</tr>
<tr>
<td>usetype</td>
<td>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.</td>
</tr>
<tr>
<td>volume. name</td>
<td>Volume to contain the cluster or component.</td>
</tr>
<tr>
<td>dbmsdds</td>
<td>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.</td>
</tr>
<tr>
<td>work.files</td>
<td>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.)</td>
</tr>
<tr>
<td>disk</td>
<td>An appropriate unit designation for the work file.</td>
</tr>
<tr>
<td>volser</td>
<td>The volume serial number of the disk or tape to be used for the work file.</td>
</tr>
<tr>
<td>tracks</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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.

```plaintext
/*
  * 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
*/
```
Key for Sample JCL to Reorganize a Database:

<table>
<thead>
<tr>
<th>Parameter Description</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>your. loadlib</strong></td>
<td>The dataset name of the z/OS or OS/390 PDS library into which CA IDMS/DB Reorg was installed.</td>
</tr>
<tr>
<td><strong>idms. loadlib</strong></td>
<td>The dataset name of the z/OS or OS/390 PDS library in which your subschema(s) and DMCL(s) reside. You must supply a separate statement for every OS/390 PDS library needed.</td>
</tr>
<tr>
<td><strong>dbfil01/dbfilnn</strong></td>
<td>The DDnames of the database files to be accessed.</td>
</tr>
<tr>
<td><strong>user. dbfil01/user. dbfilnn</strong></td>
<td>The dataset name of the file accessed by either the old subschema or the new subschema. You must supply a separate statement for every file that contains, or has set linkage with, an area named in a COPY statement.</td>
</tr>
<tr>
<td><strong>your. sortlib</strong></td>
<td>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.)</td>
</tr>
<tr>
<td><strong>sortwknn</strong></td>
<td>The DDnames of the sort files. You must supply a separate statement for every sort file needed.</td>
</tr>
<tr>
<td><strong>disk</strong></td>
<td>The appropriate unit designation for your sort files.</td>
</tr>
<tr>
<td><strong>sort. space</strong></td>
<td>The space calculated on sort input needs if sort files are on disk.</td>
</tr>
<tr>
<td><strong>vsam. work.file</strong></td>
<td>The dataset name of the one 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.)</td>
</tr>
<tr>
<td><strong>vsam-buffer</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>dbmsdds</strong></td>
<td>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, SECIX2, 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.</td>
</tr>
<tr>
<td><strong>work.files</strong></td>
<td>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.)</td>
</tr>
<tr>
<td><strong>a</strong></td>
<td>The appropriate SYSOUT class for your installation.</td>
</tr>
<tr>
<td><strong>newdmcl</strong></td>
<td>DMCL describing the database after the reorganization.</td>
</tr>
</tbody>
</table>
**Parameter Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subschema</td>
<td>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.</td>
</tr>
<tr>
<td>dmcl-name</td>
<td>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.</td>
</tr>
</tbody>
</table>

---

**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:

- **Process 1--Defining the New Areas** (see page 84)
- **Process 2--Information Gathering** (see page 84)
- **Process 3--Allocating Work Files** (see page 87)
- **Process 4--Reorganizing the Database** (see page 89)

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.

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

TRACING 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 PIREAL DISK USREXEC PIREAL 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 */
/* parameters 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'
"FORMAT cuu fm ( BLK 4K'
"RESERVE USREXEC PAGUTIL fm6'
"FILEDEF PAGUTIL DISK USREXEC PAGUTIL fm6',
"( DSORG DA'
"FILEDEF PAGUTL DISK USREXEC PAGUTIL fm6',
"( DSORG DA' */
SAY 'STARTING DATABASE REORGANIZATION'
SIGNAL OFF ERROR
'EXECOS OSRUN USRDVR'
USREORG RC = RC
IF USREORG RC > 4
THEN DO
  CALL ERROR
END
"CP SPOOL PRINTER NOCONT'
"CP CLOSE PRINTER NAME DBREORG LISTING'
"CP SPOOL PRINTER OFF'
SAY 'USREXEC FINISHED WITH A RETURN CODE OF' USREORG RC
"GLOBAL LOADLIB'
"GLOBAL TXTLIB'
"FILEDEF * CLEAR'
EXIT USREORG RC
/++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++/
ERROR:
/++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++/
ERROR RC = RC
TRACE OFF; SIGNAL OFF ERROR
SAY 'NON-ZERO RETURN CODE ENCOUNTERED IN EXEC AT LINE' SIGL
"CP SPOOL PRINTER NOCONT'
"CP CLOSE PRINTER NAME DBREORG LISTING'
"CP SPOOL PRINTER OFF'
"GLOBAL LOADLIB'
"GLOBAL TXTLIB'
"FILEDEF * CLEAR'
EXIT ERROR_RC
/*

Key for Sample EXEC to Run CA-IDMS/DB Reorg:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>yourlib</td>
<td>The file name of the load library into which you downloaded CA IDMS/DB Reorg.</td>
</tr>
<tr>
<td>idmslib</td>
<td>The file name of the load library containing your subschema and DMCL modules.</td>
</tr>
<tr>
<td>sortlib</td>
<td>The file name of the text library containing your sort modules.</td>
</tr>
</tbody>
</table>
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 USERTYPE 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**.

```plaintext
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.

```plaintext
DEL (vsam.work.space)CLUSTER
DEFINE CLUSTER (NAME(vsam.work.space)--
NUMBERED--
RECORDSIZE (8168 8168)--
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 'PAGUTIL'
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:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vsam. work. space</td>
<td>The file name of the VSAM PAGUTIL file to be used by CA IDMS/DB Reorg.</td>
</tr>
<tr>
<td>record. numbers</td>
<td>The number of records to be allocated in the VSAM work space.</td>
</tr>
<tr>
<td>usetype</td>
<td>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.</td>
</tr>
<tr>
<td>mastcat</td>
<td>The file name of the VSAM master catalog.</td>
</tr>
<tr>
<td>dskxxx</td>
<td>The name of the minidisk on which to allocate a the VSAM files.</td>
</tr>
<tr>
<td>n</td>
<td>The number of cylinders to be allocated for the master catalog.</td>
</tr>
<tr>
<td>cuu</td>
<td>The appropriate minidisk device address for the BDAM PAGUTIL file.</td>
</tr>
<tr>
<td>nn</td>
<td>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.</td>
</tr>
<tr>
<td>fm</td>
<td>The file mode for the relevant file.</td>
</tr>
<tr>
<td>fm ft fm6</td>
<td>The file name, file type, and file mode (respectively) of the BDAM PAGUTIL file. The file mode number must be 6.</td>
</tr>
</tbody>
</table>
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=dmcl-name, NEWDMCL=dmcl-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:

- VSE/ESA File Assignments (see page 90)
- VSE/ESA File Processing Alternate Method (see page 90)
- Process 1--Defining the New Areas (see page 90)
- Process 2--Information Gathering (see page 90)
- Process 3--Allocating Work Files (see page 92)
- Process 4--Reorganizing the Database (see page 93)

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.

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 89) 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=d
mcl-name, NEWDMCL=mcl-name, OLDDBN=db-name, NEWDBN=db-
name, COPY, STOPAFTER=ANALYSIS
COPY, AREA=area-name
/
/* * $$ EOJ

Key for Sample JCL to Obtain AUDIT/ANALYSIS Report:

<table>
<thead>
<tr>
<th>Parameter Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>your. loadlib</td>
</tr>
<tr>
<td>The data set name of the core image library into which you download CA IDMS/DB Reorg.</td>
</tr>
<tr>
<td>volser</td>
</tr>
<tr>
<td>The volume serial number or the generic assignment of the disk volume on which the file, specified in the preceding DLBL statement, resides.</td>
</tr>
<tr>
<td>SYSnnn</td>
</tr>
<tr>
<td>The programmer logical units of the database files to be accessed.</td>
</tr>
<tr>
<td>idms. loadlib</td>
</tr>
<tr>
<td>The data set name of the core image library in which your DMCL and subschema reside.</td>
</tr>
<tr>
<td>dbms. sublib /idms. sublib</td>
</tr>
<tr>
<td>The sublibrary name of the VSE/ESA library specified in the previous file name.</td>
</tr>
<tr>
<td>olldmcl</td>
</tr>
<tr>
<td>The DMCL of the old database, before it is reorganized.</td>
</tr>
<tr>
<td>subschema-name</td>
</tr>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>
## Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dmcl-name</td>
<td>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.</td>
</tr>
<tr>
<td>db-name</td>
<td>Two db-names are optional. The first db-name (OLDDBN=) describes the database before reorganization. The second db-name name (NEWDBN=) describes the database after reorganization.</td>
</tr>
<tr>
<td>area-name</td>
<td>The name of an area to be reorganized by CA IDMS/DB Reorg. (You must include a COPY parameter statement for each area that is to be reorganized.)</td>
</tr>
</tbody>
</table>

## 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.

```jcl
/*
* 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) - usertype - VOLUMES(volume.name) - ) - DATA
(NAME('vsam.work.space.DATA') - )

/*
*
```

## Work Files and Corresponding Logical Units:

<table>
<thead>
<tr>
<th>dbmsnms</th>
<th>SYSnnn</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTRL1</td>
<td>SYS010</td>
</tr>
<tr>
<td>CNTRL2</td>
<td>SYS011</td>
</tr>
<tr>
<td>PAGUTIL</td>
<td>SYS014</td>
</tr>
<tr>
<td>EXTRACT</td>
<td>SYS015</td>
</tr>
<tr>
<td>PRIOPTA</td>
<td>SYS016</td>
</tr>
<tr>
<td>SECOPTA</td>
<td>SYS017</td>
</tr>
<tr>
<td>PRIREAL</td>
<td>SYS018</td>
</tr>
<tr>
<td>UNALLOC</td>
<td>SYS019</td>
</tr>
<tr>
<td>SECREAL</td>
<td>SYS020</td>
</tr>
<tr>
<td>ALLOCX</td>
<td>SYS021</td>
</tr>
<tr>
<td>DBREC</td>
<td>SYS022</td>
</tr>
</tbody>
</table>
### Table

<table>
<thead>
<tr>
<th>dbmsnms</th>
<th>SYSnnn</th>
</tr>
</thead>
<tbody>
<tr>
<td>IXEIXOL</td>
<td>SYS25</td>
</tr>
<tr>
<td>IXESRKY</td>
<td>SYS26</td>
</tr>
<tr>
<td>IXEXTR</td>
<td>SYS27</td>
</tr>
<tr>
<td>SECALX2</td>
<td>SYS28</td>
</tr>
<tr>
<td>SECIIX2</td>
<td>SYS29</td>
</tr>
<tr>
<td>DBKEYS</td>
<td>SYS32</td>
</tr>
</tbody>
</table>

### Key for Sample JCL to Allocate VSAM Work File PAGUTIL:

<table>
<thead>
<tr>
<th>Parameter Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>vsam. work. space</strong></td>
</tr>
<tr>
<td><strong>volser</strong></td>
</tr>
<tr>
<td><strong>retrk</strong></td>
</tr>
<tr>
<td><strong>amount</strong></td>
</tr>
<tr>
<td><strong>record. numbers</strong></td>
</tr>
<tr>
<td><strong>usetype</strong></td>
</tr>
<tr>
<td><strong>volume. name</strong></td>
</tr>
</tbody>
</table>

## 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.

```plaintext
* SAMPLE JCL TO REORGANIZE A DATABASE USING NEW AREAS
* LIBRARY DEFINITIONS
* DBMS,'your.loadlib'
* IDMS,'idms.loadlib'
* LIBDEF *,SEARCH=(DBMS.,IDMS.,)
* 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:
* DBMS,SORTW1,'SORT.WORK1',0,SD
* EXTENT SYS001,,retrk,amount // ASSGN SYS001,DISK,VOL=volser,SHR
* DBMS,SORTW2,'SORT.WORK2',0,SD
* EXTENT SYS002,,retrk,amount // ASSGN SYS002,DISK,VOL=volser,SHR
* DBMS,SORTW3,'SORT.WORK3',0,SD
* EXTENT SYS003,,retrk,amount // ASSGN SYS003,DISK,VOL=volser,SHR
* DBMS,SORTW4,'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
&inveilip.
// DLBL DBFILNN,'user.dbfile0nn',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
*   DBC PARAMETERS LIKE RECFM, LRECL AND BLSIZE 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 USRDVR,SIZE=(USRDVR,#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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>your. loadlib</td>
<td>The data set name of the core image library into which you downloaded CA IDMS/DB Reorg.</td>
</tr>
<tr>
<td>idms. loadlib</td>
<td>The data set name of the core image library in which your DMCL and subschema reside.</td>
</tr>
<tr>
<td>volser</td>
<td>The volume serial number or generic assignment of the disk volume on which the file, specified in the previous statement, resides.</td>
</tr>
<tr>
<td></td>
<td>The sublibrary name of the VSE/ESA library specified in the previous file name.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>dbms. sublib</td>
<td>The relative track or relative block number: the starting position on the DASD for storage of the work file specified in the previous statement.</td>
</tr>
<tr>
<td>/idms. sublib</td>
<td></td>
</tr>
<tr>
<td>reltrk</td>
<td>The space allocation you need to store the file specified in the previous statement.</td>
</tr>
<tr>
<td>amount</td>
<td>The file ID of each of the database files to be accessed.</td>
</tr>
<tr>
<td>user. dbfile01</td>
<td>The file ID assigned to the work files used by the CA IDMS/DB Reorg utility.</td>
</tr>
<tr>
<td>/user. dbfilen</td>
<td></td>
</tr>
<tr>
<td>dbmsnms</td>
<td>The file names required by the CA IDMS/DB Reorg utility described earlier in this section.</td>
</tr>
<tr>
<td>work.files.</td>
<td></td>
</tr>
<tr>
<td>dbmsnms</td>
<td>The file ID assigned to the work files used by the CA IDMS/DB Reorg utility.</td>
</tr>
<tr>
<td>SYSnnn</td>
<td>The programmer logical units of the database files to be accessed.</td>
</tr>
<tr>
<td>#k</td>
<td>The number of kilobytes of storage to be made available to the sort phase.</td>
</tr>
<tr>
<td>newdmcl</td>
<td>DMCL describing the database after the reorganization.</td>
</tr>
<tr>
<td>subschema-name</td>
<td>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.</td>
</tr>
<tr>
<td>dmcl-name</td>
<td>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.</td>
</tr>
<tr>
<td>area-name</td>
<td>The name of an area to be reorganized by CA IDMS/DB Reorg. (You <strong>must</strong> include a COPY parameter statement for each area that is to be reorganized.)</td>
</tr>
</tbody>
</table>