CA IDMS - 19.0
Using CA IDMS/DB Audit

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

Physical Integrity of the Database ................................................................. 11

When to Use CA IDMS/DB Audit ................................................................. 11

Periodic Maintenance .................................................................................. 11
Special Situations ......................................................................................... 11
Logical Integrity ........................................................................................... 12

CA IDMS/DB Audit Finds Physical Integrity Errors ................................. 12

CA IDMS/DB Audit Fixes Physical Integrity Errors .................................. 12

CA IDMS/DB Audit Reporting ................................................................. 14
Audit Report ............................................................................................... 14
Integrity Analysis Report .......................................................................... 14
Report Formats ......................................................................................... 15

Control of CA IDMS/DB Audit ................................................................. 16
PROCESS Statement .................................................................................. 17
AUDIT Statement ..................................................................................... 17
AREA, RECORD, and SET Statements .................................................. 17
FIX Statement ......................................................................................... 17
REPORTS Statement ............................................................................. 17

Unique Access to the Database ................................................................. 18

Operating Considerations ........................................................................ 18
Back Up the Database ............................................................................... 18
Plan Space Allocation .......................................................................................................................... 18
Select Readymode ................................................................................................................................. 18
Plan Run Time ..................................................................................................................................... 19
Review Logical Integrity ......................................................................................................................... 19

CA IDMS/DB Audit Concepts ............................................................................................................ 19
Data Structure Diagram 1 ...................................................................................................................... 20
Integrity Errors ....................................................................................................................................... 21
  Pointer Errors ..................................................................................................................................... 21
  Set Errors .......................................................................................................................................... 27
    Disconnected Record ...................................................................................................................... 28
  Sorted Sets ........................................................................................................................................ 29
  Multiple Owner Records ................................................................................................................... 30
  Invalid Db-keys ................................................................................................................................. 32
  CALC Sets ......................................................................................................................................... 34
Integrated Index Integrity Errors ......................................................................................................... 35
  Pointer Errors in Integrated Index SR7/SR8 Structure ...................................................................... 36
  Orphans Records in Integrated Index Sets ......................................................................................... 37
  Miscellaneous Integrated Index Errors ............................................................................................ 38
Orphan Record Errors in Non-Integrated Index Sets ......................................................................... 43
Fragmented Record Errors .................................................................................................................. 45
Compressed Record Errors .................................................................................................................. 46
Page Errors .......................................................................................................................................... 47
Auditing Methods ................................................................................................................................. 48
  QuickCheck vs. Standard Audit ...................................................................................................... 49
Audit Phase ............................................................................................................................................. 50
  Sweep Processing ............................................................................................................................ 50
  Set Walk Processing ......................................................................................................................... 52
    USERSETS Option ......................................................................................................................... 52
    CALCSETS Option ......................................................................................................................... 52
    SETLIMIT Option .......................................................................................................................... 53
    DBKEYTBL Option ......................................................................................................................... 54
    Prior Pointers ............................................................................................................................... 55
  Walknext Processing ......................................................................................................................... 57
    NOSWEEP Option ......................................................................................................................... 60
  Set Disconnection Processing ............................................................................................................ 60
    Verifying Pointers ........................................................................................................................ 61
    DISCONNECT with WALKNEXT ................................................................................................. 61
Integrated Index Processing .............................................................................................................. 62
Page Audit Processing .......................................................................................................................... 63
Specifying Database Auditing ................................................................. 63
  Area Sweeping ................................................................................. 64
  Set Auditing .................................................................................... 64
  Limiting the Audit ........................................................................... 64
Parameter Controlled Functions ............................................................... 65
Fix Phase ............................................................................................ 65
  Errors Fixed ................................................................................... 66
  Limits ............................................................................................ 66
  Errors Only Detected ..................................................................... 66
  Logical Integrity ............................................................................ 67
  Multiple Owner Fixing ................................................................. 68

CA IDMS/DB Audit Parameters .............................................................. 69
Notation Conventions and Syntax ........................................................ 70
Parameter Statements and Options ......................................................... 74
  PROCESS Statement ...................................................................... 74
  AUDIT Statement ......................................................................... 74
  AREA, RECORD, and SET Statements ............................................ 74
  FIX Statement .............................................................................. 75
  REPORTS Statement .................................................................... 75
CA IDMS/DB Audit PROCESS Statement ................................................ 75
  PROCESS Parameter ....................................................................... 76
  READYMODE Option ..................................................................... 76
  SORTMSGS Option ....................................................................... 77
  STARTWITH Option ...................................................................... 77
  AUDITTYPE Option ....................................................................... 77
  SCAN Option ................................................................................ 78
The AUDIT Statement .......................................................................... 78
  AUDIT Parameter .......................................................................... 79
  PAGES Option ............................................................................... 79
  INDEX Option ............................................................................... 80
  DISCONNECT Option .................................................................... 80
  CALCSETS Option ........................................................................ 81
  USERSETS Option ........................................................................ 81
  SETLIMIT Option ......................................................................... 81
  DBKEYTBL Option ....................................................................... 82
  CORE and DISK Options ............................................................... 82
    Storage for the Db-key Table ...................................................... 82
    Storage for the Db-key Save Table ............................................. 83
    When to Use Core Storage ......................................................... 83
    When to Use Disk Storage ......................................................... 83
Report of Allocated Storage .................................................................................................................. 83
WALKNEXT Option ................................................................................................................................. 83
STOPAFTER Option ................................................................................................................................. 84
AREA, RECORD, and SET ......................................................................................................................... 84
The AREA Statement ............................................................................................................................... 85
AREA Parameter ......................................................................................................................................... 85
BEGIN and END Options ......................................................................................................................... 86
NOACCESS Option .................................................................................................................................... 86
NOSWEEP Option ..................................................................................................................................... 86
SWEEP Option ......................................................................................................................................... 87
The RECORD Statement ............................................................................................................................ 87
RECORD Parameter .................................................................................................................................. 87
DISCONNECT Option ................................................................................................................................ 87
NOACCESS Option ................................................................................................................................... 88
NOSWEEP Option ................................................................................................................................... 88
SWEEP Option ......................................................................................................................................... 88
The SET Statement ..................................................................................................................................... 88
SET Parameter ......................................................................................................................................... 89
DISCONNECT Option ............................................................................................................................... 89
AUDIT and NOAUDIT Options .................................................................................................................... 89
The FIX Statement .................................................................................................................................... 89
FIX Parameter .......................................................................................................................................... 90
PAGES Option ......................................................................................................................................... 90
INDEXORPHANS Option ........................................................................................................................... 90
CALCSETS Option .................................................................................................................................... 91
USERSETS Option ..................................................................................................................................... 91
LDEL Option ............................................................................................................................................ 91
The REPORTS Statement ........................................................................................................................... 92
REPORTS Parameter ................................................................................................................................ 92
RECDATA Options .................................................................................................................................... 93
SETMBRS Options .................................................................................................................................... 93

CA IDMS/DB Audit System Output .......................................................................................................... 94
Audit Report ................................................................................................................................................ 94
Integrity Analysis Report ............................................................................................................................ 94
Pre-Audit Scan ......................................................................................................................................... 94
  Processing Without SCAN ...................................................................................................................... 95
  Processing With SCAN ........................................................................................................................... 95
The Integrity Analysis Report ..................................................................................................................... 97
  Diagramming Errors .............................................................................................................................. 98
CA IDMS/DB Storage Requirements ........................................................................................................ 106
Operating Preparations .............................................................................................................................. 107
Set Errors -- Before Image Report ........................................................................................................... 108
Set Errors -- After Image Report .............................................................................................................. 109
Set Disconnections ................................................................................................................................. 110
Integrated Index Pointer Errors ................................................................................................................ 111
Integrated Index Orphans .......................................................................................................................... 112
Integrated Index Orphans Fixed ............................................................................................................... 113
Additional Integrated Index Errors ....................................................................................................... 114
From/Thru Page Range Errors .................................................................................................................. 115
Orphans of Sets Other Than Integrated Index Sets .................................................................................... 116
Fragmented Record Errors ....................................................................................................................... 117
LDEL Records Physically Deleted ............................................................................................................. 118
Page Errors ............................................................................................................................................. 119

CA IDMS/DB Audit Operations ..................................................................................................................... 120
Database Access ....................................................................................................................................... 121
Modes of Operation ................................................................................................................................. 122
Using the Extract File ............................................................................................................................... 123
Inclusive and Exclusive Processing .......................................................................................................... 124
Processing Examples ............................................................................................................................... 125
Parameter Guidelines ............................................................................................................................... 126
Inclusive Processing Examples .................................................................................................................. 127
Exclusive Processing Examples ............................................................................................................... 128
Mixed Processing Examples .................................................................................................................... 129
Operating Preparations ............................................................................................................................. 130
Performance Factors ............................................................................................................................... 131
Estimating USERSETS I/Os with Standard Auditing ............................................................................. 132
Estimating USERSETS I/Os with QuickCheck Auditing ........................................................................ 133
Estimating CALCSETS I/Os with Standard or QuickCheck Auditing ..................................................... 134
Estimating WALKNEXT I/Os with Standard Auditing ........................................................................ 135
Estimating DISCONNECT I/Os with Standard Auditing ........................................................................ 136
Effect of Specifying Both WALKNEXT and DISCONNECT ................................................................... 137
Improving Performance ........................................................................................................................... 138
CA IDMS/DB Storage Requirements ......................................................................................................... 139
Main Storage ........................................................................................................................................... 140
External File Storage ............................................................................................................................... 141
EXTRACTS File ....................................................................................................................................... 142
WORKFILE File ..................................................................................................................................... 143
DBKYWORK File .................................................................................................................................... 144
INDXEXTR and INDEXWORK Files ...................................................................................................... 145
Using CA IDMS/DB Audit

CA IDMS/DB Audit is a software tool that allows you to examine the physical integrity of all or part of a CA IDMS/DB. In addition, you can direct CA IDMS/DB Audit to fix most of the errors found and to report the errors and the corrections. A report of simulated corrections can be examined before actually updating the database. Elimination of physical inconsistencies in your database sets the stage for smooth operations of application programs.

CA IDMS/DB Audit uses the following features of CA IDMS/DB I/O:

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

Additionally, features of the CA IDMS/DB engine and the SYSIDMS PREFETCH parameter for read-ahead processing replace the read-ahead processing previously provided by FASTSCAN and EXCP I/O level processing previously defined by the GSDTPARM installation defaults.

For more information, see the following topics:

- Physical Integrity of the Database (see page 11)
- When to Use CA IDMS/DB Audit (see page 11)
- CA IDMS/DB Audit Finds Physical Integrity Errors (see page 12)
- CA IDMS/DB Audit Fixes Physical Integrity Errors (see page 12)
- CA IDMS/DB Audit Reporting (see page 14)
- Control of CA IDMS/DB Audit (see page 16)
- Unique Access to the Database (see page 18)
- Operating Considerations (see page 18)
- CA IDMS/DB Audit Concepts (see page 19)
- CA IDMS/DB Audit Parameters (see page 69)
- CA IDMS/DB Audit System Output (see page 94)
- CA IDMS/DB Audit Operations (see page 126)
Physical Integrity of the Database

The physical integrity of a database is a primary concern of the database administrator. If there are physical inconsistencies in the data, the system programmers, internal auditors, applications programmers, and data processing managers cannot use programs effectively, because application programs will not accurately retrieve the records from the database.

When to Use CA IDMS/DB Audit

Contents

- Periodic Maintenance (see page 11)
- Special Situations (see page 11)
- Logical Integrity (see page 12)

CA IDMS/DB Audit is useful both for periodic maintenance and for detecting and fixing errors that become apparent in special circumstances.

Periodic Maintenance

A database, like a car, will function more smoothly if it has periodic tune-ups. You can use CA IDMS/DB Audit at regular intervals to examine the physical integrity of your database, to report the errors it finds, and to correct errors of physical integrity.

Special Situations

If your car is involved in an accident, or if it is not performing efficiently, it needs testing and repairs. Likewise, special situations indicate a need for examination of your database. Several conditions are likely to cause physical integrity errors in a database:

- Improper recovery from program, system, or hardware failures
- Malfunctions in the operation of a disk drive, disk controller, or other hardware
- Improper use of the CA IDMS utility IDMSBCF, FIX PAGE function

If you know one of these conditions has occurred, you can use CA IDMS/DB Audit to search for problems before running application programs.

Even if there has not been a known failure in the system, results of application programs sometimes indicate broken chains or other physical integrity errors. If an application program terminates abnormally or if the output from a program indicates broken chains, CA IDMS/DB Audit can find and correct most of the physical integrity errors.
Logical Integrity

Once the physical integrity errors have been found and corrected by CA IDMS/DB Audit, you can use other means to find and correct logical errors in your database. An example of a logical error is shown in the section on the Fix Phase in Concepts (see page 19).

CA IDMS/DB Audit Finds Physical Integrity Errors

CA IDMS/DB Audit finds physical integrity errors in your CA IDMS/DB. Several errors are illustrated in Figure 1.1. CA IDMS/DB Audit detects these problems:

- Inconsistent page headers and footers
- Record occurrences that fall outside the page range for a record type
- Set disconnections
- Invalid pointers
- Fragmented records that cannot be properly assembled
- Orphan records not properly connected to a set
- Integrated index structure errors
- Set integrity errors
- Logically deleted records (LDELS)

CA IDMS/DB Audit Fixes Physical Integrity Errors

CA IDMS/DB Audit can fix most of the errors it finds; it can also physically delete LDELS and adopt integrated index orphans. If you want to review the corrections before actually updating the database, CA IDMS/DB Audit produces reports of simulated changes.
Figure 1.1: Examples of Physical Integrity Errors

IDMSDB--CA IDMS/DB Audit Fixes Physical Integrity Errors
CA IDMS/DB Audit Reporting

CA IDMS/DB Audit produces two types of reports: an Audit Report and an Integrity Analysis Report. The Audit Report displays a summary of the CA IDMS/DB Audit run; the Integrity Analysis Report shows details of each physical integrity error, found and/or corrected.

- Audit Report (see page 14)
- Integrity Analysis Report (see page 14)
- Report Formats (see page 15)

Audit Report

Every run of CA IDMS/DB Audit produces an Audit Report. The report contains these sections:

- Parameter Validation
- Parameter Processing Option
- Areas, Records, and Sets
- Runtime Status and Statistics
- Audit Statistics
- Fix Statistics.

Integrity Analysis Report

Generation of the Integrity Analysis Report is controlled by the REPORTS parameter statement. You can generate reports for the current audit and fix operations; you can suppress the Integrity Analysis Report; or you can generate reports for a previous audit from which you saved the extract file.

The Integrity Analysis Report displays errors found in the audit phase and corrections (simulated or actual) made in the fix phase. The report is comprised of two sections:

**Before Image Report**--a report listing the details of all physical integrity errors found by CA IDMS/DB Audit in the audit phase.

**After Image Report**--a report showing the corrections made to the database during the fix phase. The After Image Report displays either simulated or actual corrections, depending on whether the fix phase was run in simulation or update mode.

If CA IDMS/DB Audit was run in the simulation mode, the report shows how the corrections would affect the database. After reviewing the simulated corrections, if you want to actually update the database, you can then make a separate fix run of CA IDMS/DB Audit, using the extract file saved...
from the previous run. Updates should not be performed on the database between the audit phase and the fix phase of CA IDMS/DB Audit, because such updates could alter the results or cause abnormal termination of the fix processing.

Report Formats

You can specify the type of notation to be used on the Integrity Analysis Report for display of data, and you can select a complete format or abbreviated format for display of sets in error.

Errors within sets can be displayed in only character notation (alphanumeric and special characters) or displayed in both character and hexadecimal notation.

Display of sets containing errors is available in either of two formats: complete format, which shows all of the members of each set in error; or abbreviated format, which shows the owner of the set in error, each member that is in error, and the next and prior members surrounding each member in error.
IDMSDB--Report Formats

*Figure 1.2: Reports*

**Control of CA IDMS/DB Audit**

**Contents**
- PROCESS Statement (see page 17)
- AUDIT Statement (see page 17)
- AREA, RECORD, and SET Statements (see page 17)
Operation of CA IDMS/DB Audit is controlled by parameter statements.

**PROCESS Statement**

The PROCESS parameter statement allows you to specify how the database is to be readied and which SORT messages to display on the Audit Report. Additional options initiate inclusive or exclusive processing, specify the audit type as either Standard or QuickCheck, and enable you to check your parameters before auditing begins.

**AUDIT Statement**

The AUDIT parameter statement allows you to control the audit phase of CA IDMS/DB Audit. You can specify whether you want to find physical integrity errors in CALC sets, errors in user-defined sets, page errors, disconnections in sets, or index orphans (that is, records whose integrated index "up" pointer does not accurately point to the correct SR8 record). You can also specify any combination of those choices.

**AREA, RECORD, and SET Statements**

Associated with the AUDIT statement are AREA, RECORD, and SET statements, which allow you to limit the examination to a part of the database defined by the subschema.

**FIX Statement**

The FIX statement directs CA IDMS/DB Audit to simulate corrections or to actually update the records in the database. The errors to be corrected must be recorded in an extract file, from either a current run or a previous run. If the audit was made in a previous run, you must be sure that the database was not updated between the audit run and the fix run.

Corrections can be made to page headers and footers; next, prior, and owner pointers in CALC sets, user-defined sets, and integrated index sets. Integrated index orphan records can be "adopted" and the orphan counts decreased. Logically deleted records (LDELS) can also be physically deleted from the database.

**REPORTS Statement**

The REPORTS statement controls production of the Integrity Analysis Report.
Unique Access to the Database

CA IDMS/DB Audit uses a unique method to access the CA IDMS/DB database; it provides these efficient features:

- CA IDMS/DB Audit can access the SR1 records and walk or fix the CALC set as it does any user defined record or set
- CA IDMS/DB Audit can examine an entire physical page as well as an individual data record
- When CA IDMS/DB Audit performs an area sweep it uses several buffers to reduce the amount of I/O wait time

Operating Considerations

Contents

- Back Up the Database (see page 18)
- Plan Space Allocation (see page 18)
- Select Readymode (see page 18)
- Plan Run Time (see page 19)
- Review Logical Integrity (see page 19)

In order to use CA IDMS/DB Audit to best advantage, you should consider several operating procedures. Among these are backing up the database, selecting the readymode, selecting parameters that do not unduly lengthen the run time, and reviewing the logical integrity.

Back Up the Database

CA IDMS/DB Audit does not write before images and after images of changed data to a journal file. Therefore, you should back up the database before attempting fix processing. You should also back up the database after fix processing has been performed.

Plan Space Allocation

If you are auditing large sets, you can allocate storage for the processing tables to disk instead of core.

Select Readymode

If you are using CA IDMS/DB Audit as a periodic maintenance tool, you can select RETRIEVAL mode. The database is readied in SHARED RETRIEVAL mode, and CA IDMS/DB Audit can share the database with other programs. In this mode, however, CA IDMS/DB Audit can only simulate corrections; it
cannot actually update the database in RETRIEVAL mode. Also, if you use RETRIEVAL mode, you should ensure that no other program updates the database while CA IDMS/DB Audit is auditing or fixing the database.

**Plan Run Time**

The total amount of time to run CA IDMS/DB Audit depends on which parameters you specify. If the utility is required to access record occurrences and set occurrences over and over, the operating time is significantly increased. The most time consuming features are DISCONNECT and WALKNEXT. You can run CA IDMS/DB Audit without these parameters, and then use them in another run if there are indications on the Audit Report that there are disconnected or orphan records. The Standard audit type is generally slower than the QuickCheck type and QuickCheck provides the added bonus of orphan detection without WALKNEXT.

**Review Logical Integrity**

Review the reports to understand the errors that were detected and the way the errors have been fixed. It may be necessary to validate the *logical* integrity of the corrected set occurrences. CA IDMS/DB Audit finds and corrects only *physical* integrity errors.

**CA IDMS/DB Audit Concepts**

CA IDMS/DB Audit detects a number of different physical integrity errors. The first section of this section describes each type of error and provides an illustrated example. The conventions and the data structure diagram used for these examples are presented on the following pages.

QuickCheck and Standard are the two auditing methods CA IDMS/DB Audit provides. The method you select determines the types of errors CA IDMS/DB Audit can locate. The auditing method also greatly affects processing time.

In the audit phase, CA IDMS/DB Audit uses a variety of processes to find errors. This section of the section describes each process, the parameter used to initiate each process, and the specific error type it searches for. Set diagrams illustrate these processes.

In the fix phase, CA IDMS/DB Audit fixes most error types located by the audit phase. This section describes which errors can be fixed and also presents diagrammed examples.

For more information, see the following topics:
- Data Structure Diagram 1 (see page 20)
- Integrity Errors (see page 21)
- Auditing Methods (see page 48)
- Audit Phase (see page 50)
- Specifying Database Auditing (see page 63)
- Parameter Controlled Functions (see page 65)
- Fix Phase (see page 65)
Data Structure Diagram 1

Figure 2.2 presents the data structure diagram TESTSUB, the sample database used in the examples in this section.

NOTE: All IX sets are system-owned integrated index sets residing in INDBX-REGION.
Integrity Errors

CA IDMS/DB Audit can detect the following categories of physical integrity errors when auditing a database:

- Pointer Errors
- Set Errors
- Integrated Index Errors
- Userset Orphan Record Occurrences
- Fragmented (variable length) Record Errors
- Compressed Record Errors
- Page Errors

All physical integrity errors are reported on the Integrity Analysis Report. For each error encountered, the report lists the members of the set, plus an error message just below the error. If SETMBRS=ALL (default) is specified, every member of the set is listed. If SETMBRS=ERRSONLY is specified, the only records displayed are the owner, the member in error, and the prior and next records for each member in error.

- Pointer Errors (see page 21)
- Set Errors (see page 27)
  - Disconnected Record (see page 28)
  - Sorted Sets (see page 29)
  - Multiple Owner Records (see page 30)
  - Invalid Db-keys (see page 32)
  - CALC Sets (see page 34)
- Integrated Index Integrity Errors (see page 35)
  - Pointer Errors in Integrated Index SR7/SR8 Structure (see page 36)
  - Orphans Records in Integrated Index Sets (see page 37)
  - Miscellaneous Integrated Index Errors (see page 38)
- Orphan Record Errors in Non-integrated Index Sets (see page 43)
- Fragmented Record Errors (see page 45)
- Compressed Record Errors (see page 46)
- Page Errors (see page 47)

Pointer Errors

Pointer errors are related to an individual record occurrence in a specific set. CA IDMS/DB Audit can correct pointer errors in user-defined sets, integrated index sets, and CALC sets. There are six types of pointer errors.
1. Next, prior, or owner db-key points outside area page ranges in the subschema

CA IDMS/DB Audit verifies that all member records of a multi-member set to be audited, and the areas where the records reside, are copied into the subschema used by CA IDMS/DB Audit. (Except CA IDMS/DB Audit cannot verify that all CALC records are copied into the subschema when CALCSET audit is requested.) Therefore, the page number for a next, prior, or owner pointer must be for a page within an area copied into the subschema.

For example, in Figure 2.3, assume that the only area copied in the subschema is DEPT-REGION. As a result, only records within the DEPT-REGION area can be copied into the subschema. In this case, John Allen's next db-key is not within the DEPT-REGION page ranges and is in error. CA IDMS/DB Audit will fix the error by correcting the pointer so that it points to HISTORY.
Figure 2.3: Area Points to Area Outside Page Range

2. Next, prior or owner db-key points to a non-readied area

CA IDMS/DB Audit verifies whether all member records of a set to be audited, and the areas where the records reside, are copied into the subschema used by CA IDMS/DB Audit. (Except CA IDMS/DB Audit cannot verify that all CALC records are copied into the subschema when CALCSET audit is requested.) Therefore, the page number in a next, prior, or owner pointer must be for a page within an area readied by CA IDMS/DB Audit.
For example, in Figure 2.4, assume that the DEPT-REGION area was readied and the STUDENT-REGION area was not readied, because NOACCESS was specified by the user. As a result, Joan Brown's next db-key is invalid, because it points to a record that is in the non-readied area. CA IDMS/DB Audit corrects the next db-key to point to ENGLISH.

### Figure 2.4: Record Points to Non-Readied Area

- **DEPT**
  - **DEPT-TEACHER**
  - **TEACHER**

<table>
<thead>
<tr>
<th>AREA</th>
<th>LO PAGE NUMBER</th>
<th>HI PAGE NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEPT-REGION</td>
<td>2500</td>
<td>2600</td>
</tr>
<tr>
<td>STUDENT-REGION</td>
<td>6000</td>
<td>6100</td>
</tr>
</tbody>
</table>

**LINKAGE=NEXT, PRIOR, OWNER**

<table>
<thead>
<tr>
<th>PAGE 2501</th>
</tr>
</thead>
<tbody>
<tr>
<td>2501 05</td>
</tr>
<tr>
<td>2501 06</td>
</tr>
<tr>
<td>6051 03</td>
</tr>
</tbody>
</table>
3. Next, prior, or owner db-key points to a non-existent record

In Figure 2.5, Carol Smith's next db-key points within the DEPT-REGION area, but the record for the specified db-key does not exist. Because the set is a user set, CA IDMS/DB Audit corrects the next pointer so that it points to the Joan Brown record.

Figure 2.5: Record Points to Non-Existent Record

4. Next, prior or owner db-key points to an invalid record type for the set
CA IDMS/DB Audit verifies whether all member records of a set are copied into the subschema used by CA IDMS/DB Audit. (Except CA IDMS/DB Audit cannot verify that all CALC records are copied into the subschema when CALCSET audit is requested.) Therefore, a valid next, prior, or owner db-key must point to a valid record type for that set.

For example, in Figure 2.6, the only valid record types for the DEPT-TEACHER set are a DEPT record and a TEACHER record. Therefore, Jane Doe's next db-key is in error because it is pointing to SUBJECT record Creative Writing. CA IDMS/DB Audit corrects the error by changing the next pointer so that it points to the MATH record.

![Diagram of record relationships]

**Figure 2.6: Record Points to Invalid Record Type**

5. Next, prior or owner db-key is null

The next, prior, or owner db-key for this record is equal to HIGH-VALUES. If the set is user-defined, CA IDMS/DB Audit corrects the error by entering a valid pointer.

6. Db-key of record is outside the from/thru page range for record type

This type of error probably occurs because a schema was compiled when the database was originally developed, and a FROM/THRU page range was not specified for the record type. As a result, the FROM/THRU page range defaulted to the low-area-page/high-area-page number of the area. The schema was recompiled with a different FROM/THRU page range for the record type. Some records stored before the recompilation now fall outside the new FROM/THRU page range.

This error does not affect the walking of a set, but it can cause CALC records stored before the recompilation to be lost. This loss happens because the CA-IDMS CALC routine uses the current FROM/THRU page range in its calculation of the randomized CALC page number.

In Figure 2.7, the FROM/THRU page range for the TEACHER record is pages 2500 thru 2550. The db-key of Charles Spencer falls outside this range. CA IDMS/DB Audit does not fix this type of error.
Set Errors

Set errors are related to a set occurrence or to a group of record occurrences for a specific set. There are five major groups of set errors:

- Disconnected Record
Disconnected Record

There are two types of disconnected records:

1. A fully disconnected record

A fully disconnected record is one that is not connected to any set occurrence. All of the disconnected record's db-key values pointing to the specified set are HIGH-VALUES. CA IDMS/DB Audit finds disconnected records when the AUDIT statement indicates DISCONNECT=EXPLICIT (plus the DISCONNECT option in RECORD or SET statements), or DISCONNECT=ALL.

Mandatory-automatic sets are searched for disconnections when you specify the ALL or the EXPLICIT option. Because CA IDMS automatically connects mandatory-automatic sets at store time, and allows disconnect only at ERASE, a record belonging to a mandatory automatic set should never appear disconnected. In Figure 2.8, David Herman is an example of a fully disconnected record because it is not connected to any set.

2. A partially disconnected record

A partially disconnected record is a record that points into a set, but no record in the set points to it. Figure 2.9 presents a partially disconnected record.

Partially disconnected records are also referred to as orphan records. Orphans are described in greater detail later in this section.
Sorted Sets

There are two types of errors in sorted sets:

1. Sorted set is out of sequence

In a set that is sorted in descending order, the sort key of the previous record in the set is greater than the sort key of the current record. In a set that is sorted in ascending order, the sort key of the previous record is less than the sort key of the current record. For example, in Figure 2.10, the DEPT-TEACHER set should be sorted in ascending order by last-name. Jane Doe is out of sequence, however; she should precede Harold Jones in the set. CA IDMS/DB Audit corrects this set so that ENGLISH points to Jane Doe, Jane Doe points to Harold Jones, and Harold Jones points to Carol Smith.
2. Duplicates in a sorted set when duplicates are not allowed

The sort key of the current record equals the sort key of the previous record in the set occurrence. For example, in Figure 2.11, duplicates are not allowed within the DEPT-TEACHER set. David Herman, however, is found to be in the set twice. CA IDMS/DB Audit does not fix this type of error.

Multiple Owner Records

There are two error situations that cause CA IDMS/DB Audit to report a second owner in a set. Only Standard auditing detects multiple owners in a set. The way CA IDMS/DB Audit corrects multiple owner record errors is described in detail later in this section.

1. A single set contains two owners

In Figure 2.12, the set occurrence owned by the ENGLISH record is a complete set chain. As the set is walked from the ENGLISH record, however, another owner record (MATH) is encountered.
2. A broken set points into a complete set

The set owned by the HISTORY record in Figure 2.13 is broken. When this set is walked in the next direction, a second owner record (CHEMISTRY) is retrieved. Carol Smith's next db-key points to James Lyons' record, which is in the set occurrence owned by CHEMISTRY.
Invalid Db-keys

CA IDMS/DB Audit reports three types of invalid db-keys:

1. Prior db-key does not point to the prior member of set

When walking a set in the next direction, the prior db-key does not point to the prior record in the set. For example, in Figure 2.14, walking next from the ENGLISH record, Harry Dean is retrieved. Walking next again, Carol Smith is retrieved but her prior db-key does not point back to Harry Dean; it points to another valid record occurrence (Joan Brown).

**Figure 2.14: Prior Pointer Does Not Point to Prior Record**

2. Next db-key does not point to the next member of set

When walking a set in the prior direction, the next db-key does not point to the next record in the set (the record just walked from). For example, in Figure 2.15, walking prior from the ENGLISH record, all next db-keys point back to the previous record until Harry Dean is retrieved. Harry Dean's next db-key does not point to Jane Doe, but to Carol Smith.

In both of these situations, CA IDMS/DB Audit evaluates the error by walking the set in the next and prior direction and fixes the set such that Harry Dean's next db-key points to Jane Doe.
IDMSDB--Invalid Db-keys (2)

3. Owner db-key does not point to the owner of the set

In Figure 2.16, Jane Doe's owner db-key does not point to the owner of the DEPT-TEACHER set. Instead it points to the HISTORY record, which is an owner record of another DEPT-TEACHER set occurrence.

CA IDMS/DB Audit corrects this error by making Jane Doe's owner db-key point to English.
IDMSDB--Invalid Db-keys (3)

Figure 2.16: Owner Db-key Does Not Point to Set Owner

CALC Sets

To identify any errors in the CALC set, CALCSETS must be specified on the AUDIT statement. There are four types of CALC set errors:

1. The CALC set is out of sequence

The sequence of the CALC set is determined by the same rules that govern any multi-member sorted set. This error is fully described in the earlier section on Sorted Sets.

2. A VIA or DIRECT record is connected into the CALC set

This error is fully described in the earlier section Pointer Errors. Next, prior, or owner db-key points to an invalid record type for the set.

3. A CALC record is disconnected from the CALC set

This error is fully described in the earlier section on Disconnected Record.
4. A CALC record is connected to the CALC set on an incorrect page

Every page of an area contains a system record (SR1) that is the owner of all CALC records whose keys randomize to that page. The CALC set is the only set for which CA IDMS/DB Audit can determine the accuracy of the set owner by invoking the IDMSCALC (or IDMSCLCX) routine for each CALC record. CA IDMS/DB Audit ensures that each CALC record is connected to the correct CALC set. (If a CALC record is not connected to the correct CALC set, it cannot be located by an OBTAIN CALC command.) In Figure 2.17, ART DEPARTMENT is an example of a CALC record connected to the CALC set on an incorrect page. CA IDMS/DB Audit corrects this so that ART DEPARTMENT is connected to the CALC set on page 88,004.

![Diagram of SR1, CALC, DEPT, and ART DEPARTMENT]

Integrated Index Integrity Errors

To detect integrated index errors, specify INDEX in the AUDIT parameter statement. In the diagrams of integrated index sets, each SR8 record is shown with its db-key and entries relevant to the illustrated physical integrity error. Entries not shown in the SR8 record are represented by the shaded portion. A thick black arrow shows the location of an integrity error.
IDMSDB—Integrated Index Integrity Errors

There are three major groups of integrated index errors:

- Pointer errors in the SR7/SR8 structure
- Integrated index orphans and orphan count
- Miscellaneous integrated index errors

**Pointer Errors in Integrated Index SR7/SR8 Structure**

Errors in next, prior, and owner pointers in integrated index sets are detected when you specify USERSETS in the AUDIT statement, and are fixed when you specify USERSETS in the FIX statement. CA IDMS/DB Audit generates the same messages for pointer errors in other sets in the database. Figure 2.18 shows a typical integrated index structure with a bad next pointer. The NEXT pointer in record 9000-03 points to the level above instead of the next record.
Orphans Records in Integrated Index Sets

CA IDMS/DB Audit detects integrated index orphan records. An integrated index orphan occurs when the upper level pointer of an SR8 record or a member record of an integrated index set does not point to the correct record in the level above it. Integrated index orphans are not integrity errors because this condition can occur normally when the database spawns new SR8 records to accommodate the addition of new member records. Their presence, however, can degrade performance.

For example, in Figure 2.19 upper level record 9000-06 points to record 9000-03 but record 9000-03 points back to record 9000-05. To fix this condition, specify INDEXORPHANS in the FIX parameter statement. INDEXORPHANS also causes the orphan counts in SR8 records to be corrected.
Miscellaneous Integrated Index Errors

CA IDMS/DB Audit can detect, but not fix, seven other types of integrity errors in an integrated index.

1. Index entry db-key errors detected in integrated index records

When checking the db-keys of records in an integrated index set, CA IDMS/DB Audit either could not find the record pointed to by the db-key, or found errors within the record.

- **Record not found.** The record indicated by the db-key could not be found. For example, in Figure 2.20, the bottom level SR8 record points to a member record 6-7 that cannot be found.

- **Record type incorrect.** The db-key points to a record whose type does not belong in the set. For example, the db-key in a TEACHER index set points to a CLASS record.

- **Record not within page range.** The page where the SR8 record resides is not within the designated page range. The page range for an SR8 is determined by the high and low pages of the set to which it belongs. These pages can be defined in the schema. If the pages are not specifically defined, the page range is the same as the page range for the owner record.

- **Record located in non-readied area.** An SR8 record points to a record that is in a non-readied area.

![SR8 TABLE ENTRIES](image)

IDMSDB--Miscellaneous Integrated Index Errors

*Figure 2.20: Integrated Index Entry Not Found*

2. SR8 upper level pointer is not high values

In a top level SR8 record, the upper level pointer should be null (equal to HIGH-VALUES). If the upper level pointer in a top level SR8 is not null, CA IDMS/DB Audit lists the error under db-key errors and indicates that the upper level pointer is not HIGH-VALUES.

In the example in Figure 2.21 the upper level pointer in record 9000-07 points to a record in another set.
3. Level sequence errors detected in integrated indexes

The level number of an SR8 record in an integrated index set is in error. For example, in Figure 2.22 the level number in Level 1 should be 1, but the records contain Level 0 and 3.

4. Symbolic key errors detected in integrated index records in set
In a set sorted by symbolic key, an SR8 record points to a member record, but the symbolic key contained in the SR8 record does not match the symbolic key in the member record. In Figure 2.23 the db-key in the SR8 record points to the JONES member record, but the symbolic key in the SR8 record is WHITE.

**Figure 2.23: Incorrect Bottom Level Symbolic Key**

5. **Sort key sequence errors detected in integrated index records contained in set**

Integrated index sets can be sorted either by db-key or by symbolic key, in ascending or descending order.

In a set sorted by db-key, the last db-key in an SR8 record should match the sort key in the upper level SR8 record just above. For example, in Figure 2.24, in the SR8 record on the right in the bottom level, the last db-key 3-6 does not match the sort key 4-8 in the SR8 record above it.

In a set sorted by symbolic key, the last symbolic key in an SR8 record should match the upper level symbolic key associated with the db-key. For example, in Figure 2.25, the symbolic key CLIFF does not match the symbolic key JONES in the top level SR8 record above.
IDMSDB--Miscellaneous Integrated Index Errors (5)

Figure 2.24: Incorrect Upper Level Db-key

IDMSDB--Miscellaneous Integrated Index Errors (6)

Figure 2.25: Incorrect Upper Level Symbolic Key

6. Sort flag inconsistent with symbolic sort flag in integrated index record
An SR8 record has flags that indicate whether the set is sorted and whether it is sorted by symbolic key. The sorted set flag ON and the symbolic key flag OFF indicates a set sorted by db-key. The sorted set flag ON and the symbolic key flag also ON indicates a set sorted by symbolic key. If the sorted set flag is OFF, the symbolic sort key flag should be OFF. Figure 2.26 shows the combinations of flag settings with their meanings.

<table>
<thead>
<tr>
<th>Sorted Set Flag</th>
<th>ON</th>
<th>ON</th>
<th>OFF</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbolic Flag</td>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>Set</td>
<td>Sorted by db-key</td>
<td>Sorted by symbolic key</td>
<td>Unsorted</td>
<td>Physical Integrity Error</td>
</tr>
</tbody>
</table>

IDMSDB--Miscellaneous Integrated Index Errors (7)

Figure 2.26: Integrated Index Sorted Flag Settings

7. Integrated index records disconnected from set

A member record is disconnected from an integrated index set. If INDEX is specified in the AUDIT statement, CA IDMS/DB Audit sweeps the area and creates a table of the db-keys of set member records. During index auditing, CA IDMS/DB Audit lists all the db-keys of member records of each mandatory-automatic integrated index set. Each member record should have a pair of entries—one obtained during sweep processing, and one obtained during the index auditing. If there is no pair, CA IDMS/DB Audit reports the member as disconnected.

In Figure 2.27, for example, three of the records in the SWEEP portion of the table are not paired with INDEX records, and two of the records in the INDEX portion of the table are not paired with SWEEP records. The five records are reported as disconnected.
Orphan Record Errors in Non-Integrated Index Sets

A record designated as an orphan record points at another set or record, but neither the owner of the set nor any member record in the set points back at the orphan record. This type of integrity error is detected during walknext and QuickCheck processing. Do not confuse an orphan record with an integrated index orphan (described earlier). An orphan record is an integrity error.

There are two types of orphan record errors:

1. Orphan string points into valid set

CA IDMS/DB Audit found a record that can not be retrieved when the set is walked from the owner, even though the record points into the set.

For example, in Figure 2.28, records Carol Smith and Ron Barker are connected into the set occurrence owned by the ENGLISH record. When the set is walked from the owner record (ENGLISH), however, only Harry Dean, Jane Doe, and Robin Hayes are retrieved. The Carol Smith and Ron Barker records can only be retrieved if stored CALC, or if the area is swept. CA IDMS/DB Audit corrects this situation by making Harry Dean point to Ron Barker.
CA IDMS - 19.0

IDMSDB--Orphan Record Errors in Non-Integrated Index Sets

Figure 2.28: Orphan String Points into Valid Set

2. Orphan string is fully disconnected

CA IDMS/DB Audit could not retrieve an owner record in a set occurrence that was entered either by retrieving one of the member records in an area sweep or by retrieving a member record that was stored CALC. In addition, the member record chain cannot be retrieved by walking from the owner of the set.

For example, in Figure 2.29, TEACHER records Ron Barker and Carol Smith are connected together; however, no valid owner of a DEPT-TEACHER set points to the string. Also, Carol Smith's next db-key is invalid. CA IDMS/DB Audit cannot correct this situation.

IDMSDB--Orphan Record Errors in Non-Integrated Index Sets (2)
Fragmented Record Errors

CA IDMS/DB Audit finds any instance of a fragmented (variable length) record whose fragments could not be properly assembled. For example, in Figure 2.30, the subject record Biology is variable length and should consist of the root portion and one fragment. The pointer within the root portion of the record, however, points to an SR4 record (fragment) that does not exist. CA IDMS/DB Audit cannot fix errors of this type.
Compressed Record Errors

If there are errors in the compressed portion of a record, it cannot be successfully decompressed. Improper decompression can cause a fragmented record to be improperly assembled. CA IDMS/DB Audit cannot fix errors of this type.
Page Errors

CA IDMS/DB Audit detects one type of page error:

Invalid page number

When an area is being swept, CA IDMS/DB Audit keeps track of the Page-Number of the page that is to be retrieved. A check of the page number within the page header and page footer is performed. For example, in Figure 2.31, the DEPT-REGION low and high page numbers are 5001 and 5100, respectively. When the 12th page is retrieved within the area, the page number of the returned page should be 5012; however, the page footer for page 5012 shows number 5094. CA IDMS/DB Audit corrects the page footer to be 5012.
Auditing Methods

Contents
- QuickCheck vs. Standard Audit (see page 49)
CA IDMS/DB Audit has two methods of auditing:

- Standard audit
- QuickCheck audit

**Standard auditing** examines your application databases and data dictionary for all types of physical integrity errors. Although Standard auditing is the most comprehensive, it requires numerous I/Os and potentially long processing times.

**QuickCheck** is a much faster processing alternative for detecting common physical integrity errors. QuickCheck requires significantly fewer I/Os and less CPU time because it does not walk every set occurrence to detect errors. Instead, during area sweeps QuickCheck creates pointer extract records that are then sorted. If CA IDMS/DB Audit detects any mismatched pointer extract records, it then walks only those sets that are in error. Finally, errors found by QuickCheck are reported and most can be corrected using the same fix parameters specified in Standard auditing.

### QuickCheck vs. Standard Audit

Deciding whether to use QuickCheck or Standard auditing depends upon the nature of the audit. Use QuickCheck when time is limited and your main concern is locating next, prior, or owner (N/P/O) pointer errors. Because Standard auditing is more thorough, it is the preferred alternative when you need to search for all error types.

Figure 2.32 summarizes the major differences between QuickCheck and Standard auditing. Features that are supported by one method or the other are marked with an X. Several features are supported under QuickCheck, but only under certain circumstances.

These features are qualified by a brief explanation.

<table>
<thead>
<tr>
<th>Features</th>
<th>Standard</th>
<th>QuickCheck</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detects N/P/O errors in user sets</td>
<td>X</td>
<td>X - faster performance</td>
</tr>
<tr>
<td>Detects N/P/O errors in IIX sets</td>
<td>X</td>
<td>X - faster performance</td>
</tr>
<tr>
<td>Detects ownerless sets</td>
<td>X</td>
<td>if located in N/P/O error set</td>
</tr>
<tr>
<td>Detects sets with 2 or more owners</td>
<td>X</td>
<td>if located in N/P/O error set</td>
</tr>
<tr>
<td>Detects sorted set sequence errors</td>
<td>X</td>
<td>if located in N/P/O error set</td>
</tr>
<tr>
<td>Detects fully disconnected records</td>
<td>with WALKNEXT and DISCONNECT</td>
<td>X - faster performance</td>
</tr>
<tr>
<td>Detects incorrect CALC membership</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Detects logically deleted records</td>
<td>X</td>
<td>if located in N/P/O error set</td>
</tr>
<tr>
<td>Detects non-N/P/O IIX set errors</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Audit Phase

In the audit phase, CA IDMS/DB Audit searches through a database looking for physical integrity errors. Audit creates an extract file containing each physical integrity error that is found and each logically deleted (LDEL) record that is found. This extract file is used by the fix and reports phases for further processing.

CA IDMS/DB Audit uses the following audit processes:

- **Sweep Processing** (see page 50)
- **Set Walk Processing** (see page 52)
- **Walknext Processing** (see page 57)
- **Set Disconnection Processing** (see page 60)
- **Integrated Index Processing** (see page 62)
- **Page Audit Processing** (see page 63)

The user initiates one or more of these processes through parameter statements and options. Although CA IDMS/DB Audit Parameters (see page 69) all parameters in detail, this section describes the audit processes in general and provides diagrammed examples.

Sweep Processing

The primary method that is used by CA IDMS/DB Audit to access a user database is sweep processing. Using sweep processing, CA IDMS/DB Audit starts retrieving the records in an area that is defined by the subschema. Audit retrieves all records according to AREA, SET, and RECORD parameter statements. After a record is retrieved, CA IDMS/DB Audit refers to the parameters you specify to determine how to audit the record and its sets.

If you specify the PAGES option in the AUDIT statement, page headers and footers are also audited during sweep processing. When CA IDMS/DB Audit retrieves a new page during sweep processing, it examines the header and footer before looking at the records on the page.
During sweep processing, CA IDMS/DB Audit retrieves records within an area by db-key, in numerical order. For example, in Figure 2.33, assume the DEPT-REGION area, DEPT and TEACHER record types, and DEPT-TEACHER sets have been copied into the subschema. When sweeping the DEPT-REGION area, the records HISTORY, Harry Dean, ENGLISH, Carol Smith, and John Doe are retrieved in that order.

**Figure 2.33: Sweep Processing**
Set Walk Processing

During Standard auditing, once a record has been selected by the sweep process, CA IDMS/DB Audit can perform set walk processing. The main function of set walk processing is to ensure the integrity of a set occurrence. If the record retrieved by CA IDMS/DB Audit is the owner of one or more set occurrences, CA IDMS/DB Audit walks the set from the owner record until it reaches the end-of-set.

If CA IDMS/DB Audit encounters any physical integrity errors during set walking, it returns to the owner record and walks the set again, creating an extract record for each record found in the set. The file of extract records is saved for later use by the FIX phase or the REPORTS phase.

The two options that are used to initiate set walking are USERSETS and CALCSETS. The SETLIMIT and DBKEYTBL options provide additional controls on set walking.

USERSETS Option

The USERSETS option initiates set walking for user-defined sets and integrated index sets.

During USERSETS processing, CA IDMS/DB Audit walks each set only one time if it does not detect any next, prior, or owner pointer errors. A retrieved record that does not own any set occurrences is bypassed.

In Figure 2.34, only records that are owners of sets initiate a set walk during the sweep of the DEPT-REGION area. Carol Smith and John Doe do not initiate a set walk by CA IDMS/DB Audit because they are not owners of any sets. When the owner record ENGLISH is retrieved, the DEPT-TEACHER set is walked (ENGLISH, Carol Smith, John Doe, and finally ENGLISH again). In addition, the DEPT-ACADEMIC set is walked (that is, ENGLISH, American Literature, English Literature, Remedial English, and ENGLISH).

During QuickCheck auditing, USERSETS also looks for orphans. During Standard auditing, orphans are detected through WALKNEXT processing (discussed in detail in Walknext Processing (see page 57)).

CALCSETS Option

If you specify CALCSETS in the AUDIT statement, each CALC set is walked to audit for next and prior pointer errors, in a manner similar to walking all other sets. Also, during STANDARD auditing, each CALC record is checked to ensure that it is a member of the correct CALC set.
SETLIMIT Option

To prevent CA IDMS/DB Audit from entering into a set loop while set walking, you can specify a limit on the number of records, in the set, that are retrieved before assuming that a set loop condition exists. For example, in Figure 2.35, assume SETLIMIT=5. When CA IDMS/DB Audit walks the DEPT-
ACADEMIC set owned by ENGLISH, the following records are retrieved: ENGLISH, American Literature, English Literature, Remedial English, American Literature, English Literature, and Remedial English. CA IDMS/DB Audit terminates processing of the set occurrence at this point because six member records have been retrieved, one more than the SETLIMIT value.

⚠️ Note: If the SETLIMIT value is reached for any one set occurrence, CA IDMS/DB Audit does not fix any sets during the fix phase. Specifying a SETLIMIT value that is too small can cause CA IDMS/DB Audit to stop processing a set occurrence prematurely and thus prevent fixing of any sets.

**DBKEYTBL Option**

To detect a set loop and allow fixing of these errors, you can specify DBKEYTBL in the AUDIT statement. CA IDMS/DB Audit maintains a table of all member record db-keys retrieved for each set occurrence during set walk processing. Whenever the next db-key of a record is found in the table already, CA IDMS/DB Audit stops processing the set. In Figure 2.35, when auditing the DEPT-ACADEMIC set, the following member records are retrieved and their db-keys that are placed in the table: American Literature, English Literature, and Remedial English. Because the Remedial English next db-key for Remedial English is already in the table, CA IDMS/DB Audit recognizes that the set contains a loop and terminates the set-walk for that set occurrence.

⚠️ Note: Even if DBKEYTBL is not specified in the Audit statement, CA IDMS/DB Audit activates the DBKEYTBL function during the creation of extract records to determine whether a next or prior db-key points to a record previously retrieved in the set. Therefore, you must plan storage for DBKEYTBL.
Prior Pointers

If the prior db-key of a record occurrence does not point back to its prior record in the set, when walking a set containing prior pointers, CA IDMS/DB Audit saves the db-key of the current record in a save table and continues set walking. At end-of-set, CA IDMS/DB Audit retrieves each saved db-key and attempts to walk prior from that point. This process of walking, saving, and retrieving db-keys is continued until all saved db-keys have been processed.

For example, in Figure 2.36, CA IDMS/DB Audit walks a set occurrence that is owned by ENGLISH. No error is detected until the David Lee record is retrieved: the prior db-key does not point to John Doe.
Because the David Lee record is the last member of the set and an integrity error was detected in the set, CA IDMS/DB Audit returns to the owner record, ENGLISH, and walks the set again, creating extract records for each record occurrence retrieved. When the David Lee record is retrieved, the record db-key is placed in the save table. CA IDMS/DB Audit continues set walking until end-of-set is reached. CA IDMS/DB Audit then retrieves the saved db-key and begins walking prior from the saved db-key to its prior record, Joan Brown.

The following situations demonstrate the effect of specifying DBKEYTBL or not:

- When DBKEYTBL is specified: Because the prior db-key (2501-08) for Joan Brown is in the DBKEYTBL, CA IDMS/DB Audit stops walking prior at this point.

- When DBKEYTBL is not specified: CA IDMS/DB Audit continues walking before the records for John Doe and Carol Smith until it reaches the owner record ENGLISH, or until the SETLIMIT value is reached

In either case, CA IDMS/DB Audit then checks the save table for more saved db-keys. After all saved db-keys have been processed, CA IDMS/DB Audit stops creating extract records for the set.
Walknext Processing

An orphan record is one type of error that cannot be detected and adequately diagnosed by simply walking each set once from its owner. If a record points to a set of which it is a member, but the owner or any of the members do not point to the record, then it is referred to as an orphan record.

One way to detect orphan records is to walk each set, not only when the owner record is detected during sweep processing, but also when each member record of the set is retrieved. In other words, as each record is retrieved during sweep processing, not only are all set occurrences that are owned by the record walked, but all sets where the current record is a member are walked.
For Standard auditing, you initiate walknext processing by specifying WALKNEXT in the AUDIT statement. For QuickCheck auditing, orphan detection occurs when you specify the USERSETS option. You cannot specify WALKNEXT with QuickCheck auditing.

If you initiate walknext processing, CA IDMS/DB Audit walks all sets of which the retrieved record is a member. If an owner record is encountered, CA IDMS/DB Audit begins walking again from the owner record until it reaches either the original retrieved record or the owner record for a second time. For example, in Figure 2.37, when record GR 79-1 is retrieved during the area sweep, CA IDMS/DB Audit walks forward until the owner record (Bill Smith) is retrieved. Once the owner record is detected, CA IDMS/DB Audit walks forward again until record GR 79-1 is retrieved for a second time.

To understand the impact of using WALKNEXT, assume there are 6,000 STUDENT records and 24,000 GREPORT records. If the STUDENT-GREPORT set is audited using USERSETS, CA IDMS/DB Audit accesses 30,000 records (6000 owners plus 24000 members; average set length is 4). When auditing with WALKNEXT, CA IDMS/DB Audit accesses 150,000 records (30,000 set walking from the owners and 120,000 set walking from each member record). If GREPORT members are not stored VIA the STUDENT-GREPORT set, CA IDMS/DB Audit generates 150,000 I/Os, based on an average of 4 member records per set times 30,000 record occurrences, plus 30,000 I/Os for the set walk process.

⚠️ **Note:** If you specify WALKNEXT, each set is walked as many times as there are member records. This multiple walking causes a substantial increase in processing time. You can, however, search for orphans through QuickCheck with the USERSETS option and use less processing time.
If an owner record is obtained for the second time, the retrieved record is an orphan. For example, in Figure 2.38, when the Creative Writing record occurrence is retrieved, CA IDMS/DB Audit detects an orphan record. CA IDMS/DB Audit walks from Creative Writing to Remedial English back to ENGLISH. Then it walks from ENGLISH to American Literature to English Literature to Remedial English back to ENGLISH. This sequence indicates that Creative Writing is connected into the set owned by ENGLISH, but cannot be retrieved by walking from the ENGLISH record. Therefore, Creative Writing is an orphan record.

**Figure 2.37: Walknext Processing**
NOSWEEP Option

If the owner of the set is in a NOSWEEP area, the set is not walked. In that case, CA IDMS/DB Audit cannot determine whether records in another area are orphan records of sets whose owners are in the NOSWEEP area.

Set Disconnection Processing

During sweep processing, when each record is retrieved, CA IDMS/DB Audit checks for the following:

- To see if the record is a member of a set to be audited for disconnections, or
If the record type is to be audited for disconnections.

A record is disconnected if no records that are members of a set point to it and it does not point into the set.

If you specify DISCONNECT=NONE in the AUDIT statement, no search for disconnections is made.

**Verifying Pointers**

When a record is to be audited for set disconnections, each pointer is verified by finding the indicated record. For example, if there are 24,000 GREPORT records with next, prior, and owner pointers in the STUDENT-GREPORT set, then CA IDMS/DB Audit verifies each pointer (72,000 pointers) by finding 72,000 records.

Two disconnected records are illustrated in Figure 2.39. If all the pointers of a record are null, the record is *fully disconnected*. If all the pointers of a record are a combination of invalid and null pointers, the record is *partially disconnected*. Because all of the pointers of the Joan Brown record are null (indicated by -1), it is fully disconnected. The David Lee record is partially disconnected, because its only pointer that is not null does not point into a set.

**Note:** Because of the large number of I/Os and calls to DBIO, DISCONNECT can increase processing time significantly. You can, however, reduce processing time through QuickCheck. QuickCheck DISCONNECT looks for fully disconnected records only. QuickCheck USERSETS looks for partially disconnected records only.

**DISCONNECT with WALKNEXT**

WALKNEXT takes precedence over DISCONNECT when both are specified in the same run. If an error is found in both walknext processing and in set disconnection processing, the record in error is reported as an orphan record. As in Figure 2.39, David Lee would be reported as an orphan record if you specified WALKNEXT, and Disconnect.
Integrated Index Processing

If INDEX is specified in the AUDIT statement, CA IDMS/DB Audit verifies each integrated index entry against its database record occurrence. You cannot specify the INDEX option with QuickCheck auditing.

To detect integrated index orphans, CA IDMS/DB Audit examines the upper level pointers in each SR8 record.

CA IDMS/DB Audit also verifies the entries in the SR8 records of the integrated index set by trying to retrieve any records pointed to by the db-keys in the entries.
CA IDMS/DB Audit verifies:

- Upper level pointers
- Db-keys:
  - Record not found
  - Record type incorrect
  - Record not within page range
  - Record that is located in non-readied area
- Symbolic keys
- Sort key sequences (set sorted by db-key or symbolic key)
- Level numbers
- Sort flags

A check for disconnections of member records is made when CA IDMS/DB Audit sweeps the area looking for mandatory-automatic member records of integrated index sets. Audit then compares the list of the member records with a list of member records that are found while walking down from the owners of integrated index sets.

Page Audit Processing

CA IDMS/DB Audit performs page auditing when you specify PAGES in the AUDIT statement. Whenever a new page is retrieved during sweep processing, CA IDMS/DB Audit checks the page-number within the page-header and page-footer against the requested page number.

⚠️ Note: If a test SUBSCHEMA/DMCL is used to audit a production database (or vice versa), numerous pages can be incorrectly flagged as being in error. Verify that the correct SUBSCHEMA/DMCL and load/core image library are specified when requesting PAGES.

Specifying Database Auditing

Contents

- Area Sweeping (see page 64)
- Set Auditing (see page 64)
- Limiting the Audit (see page 64)

CA IDMS/DB Audit determines which parts of the database to audit by comparing the subschema and the parameters that you specify. The subschema must define these entities:
- Areas to be swept
- Areas containing owners of sets to be audited
- Areas containing members of sets to be audited
- Set types to be audited
- Owner record types of sets to be audited
- All member record types of sets to be audited
- Index sets to be audited

**Area Sweeping**

If there are no restrictions by AREA, SET, or RECORD statements, an area is swept if any of the following AUDIT parameters are specified:

- USERSETS, if owners of sets are in the area
- WALKNEXT
- PAGES
- DISCONNECT
- CALCSETS
- INDEX

An area is not swept if only USERSETS is specified and no owners of sets to be audited are in the area. Access is still available, however, to member records of audited sets whose owners are in other areas.

**Set Auditing**

When you specify USERSETS, user-defined sets named in the subschema are audited if the set types, their owner record types, all member record types, and areas containing all of those record types are defined in the subschema.

**Limiting the Audit**

You can include or exclude parts of the database for auditing by using the STARTWITH option in conjunction with AREA, SET, and RECORD statements. Realize, however, that a limit placed on an area, record, or set may unintentionally affect the auditing of another set. For example, if an AREA is designated NOSWEEP, walknext processing is not performed on any sets whose owners are in the NOSWEEP area.
Parameter Controlled Functions

By specifying various parameters, you can direct CA IDMS/DB Audit to perform the following functions:

- Report disconnections. Records disconnected from sets are detected and reported when you specify either the ALL or the EXPLICIT option of the DISCONNECT parameter.

- Stop all processing after detecting a specified number of errors by using the STOPAFTER option of the AUDIT parameter statement.

- Audit the CA-IDMS CALC set, by specifying CALCSETS.

- Audit user-defined sets and integrated index sets by specifying USERSETS.

- Verify each integrated index entry against its database record by specifying INDEX.

- Restrict the portion of an area to be swept by the audit phase, by using the BEGIN and END options in the AREA parameter statement. By using this facility, you can audit a large area in sections. Fixing will not be allowed, however, unless the owners of all sets involved in integrity errors are in the areas audited.

- Restrict the auditing of areas, records, and sets defined in the subschema by using NOSWEEP, NOACCESS, and NOAUDIT.

  - NOSWEEP prevents CA IDMS/DB Audit from accessing specific records or all the records stored in an area during CA IDMS/DB Audit’s sweep processing. It does not prevent CA IDMS/DB Audit from accessing member records as part of USERSETS auditing. If owners of sets are in a NOSWEEP area, however, those sets are not processed during walknext processing.

  - NOACCESS prevents CA IDMS/DB Audit from accessing a specified area or record under any circumstances. It will not attempt to sweep an area or access a record during sweep processing, nor will it access any records stored in an area, or any sets whose owners or members are in the area, during set walk auditing.

  - NOAUDIT prevents CA IDMS/DB Audit from performing any of its audits on the specified set.

Fix Phase

CA IDMS/DB Audit fixes most physical integrity errors found in the audit phase. You can specify either simulated fixing or actual updating of the database.

- Errors Fixed (see page 66)
- Limits (see page 66)
- Errors Only Detected (see page 66)
- Logical Integrity (see page 67)
- Multiple Owner Fixing (see page 68)
Errors Fixed

By using information contained in the extract file, CA IDMS/DB Audit fixes the following physical integrity errors:

- User-defined sets and CALC sets
  - Null db-keys
  - Db-keys that point outside of the subschema page range
  - Next, prior, or owner db-keys that point incorrectly
  - Sort keys out of sequence
  - Set loops
  - Orphan records not properly connected to a set
  - CALC records not properly connected to the CALC set
- Pages: inconsistencies in header or footer page number
- LDELS: CA IDMS/DB Audit can erase an LDEL if it is not part of a set occurrence in error or the owner of an empty set
- Integrated index sets
  - Next, prior, and owner db-keys that point incorrectly
  - Upper level pointers in orphan records
  - Orphan count

Limits

If CA IDMS/DB Audit reaches any STOPAFTER, SETLIMIT or DBKEYTBL limits, it does not fix any detected errors because the audit may not be complete.

Errors Only Detected

There are some types of physical integrity errors that CA IDMS/DB Audit cannot fix:

- FROM/THRU page range errors
- Duplicate sort keys in a set occurrence where duplicates are not allowed
- Fully disconnected record occurrences, including orphan strings
CA IDMS - 19.0

- Fragmented (variable length) records that could not be properly assembled
- Compressed records that could not be properly decompressed
- Integrated index errors other than orphans, orphan counts, and next, prior, or owner pointers (upper level pointers, db-keys, symbolic keys, sort key sequences, level numbers, or sort flags)

You can fix many of these errors by using the information produced by CA IDMS/DB Audit in conjunction with the CA IDMS utility IDMSBCF, FIX PAGE function. See individual message descriptions in Messages (https://docops.ca.com/display/IDMSM/ADS+Alive+Messages) for more information.

Logical Integrity

CA IDMS/DB Audit fixes only physical integrity errors. Logical integrity errors cannot be fixed because CA IDMS/DB Audit does not know the logical basis used to store the data on the database.

For example, in Figure 2.40, the STUDENT-REPORTS set contains two record types: GR and DR. Each record type is sorted in descending order by year and grading period. DR records are stored whenever necessary and should occur in the set after the GR records.

Assume that one of the next pointers is broken. CA IDMS/DB Audit fixes the set as shown in Figure 2.41. If logically the DR records are to follow the GR records for the same year and grading period, then the “fixed” set occurrence is not in the correct logical sequence. Nevertheless, the physical integrity of the set is correct. Logical integrity can be fixed by a user program—a correction that may not have been possible before the physical integrity was fixed.

IDMSDB--Logical Integrity

*Figure 2.40: Physical Integrity Error*
Multiple Owner Fixing

When a record is found in more than one set, CA IDMS/DB Audit places the record in the set indicated by its owner pointer, or in the set where it appears closest to the owner record. For example, in Figure 2.42, when walking from owner record ENGLISH, member records John Doe, Joan Brown, James Jones, and Paul Cary are retrieved before an error is detected. Walking from owner record MATH, member records David Kraft, James Jones, and Paul Cary are retrieved. Record occurrences James Jones and Paul Cary appear in both sets.

If such a record has an owner pointer that matches the db-key of an owner record, the member record is placed in that owner’s set.

If the record does not have a valid owner pointer, the record is placed in the set where it is nearest the owner record. In Figure 2.42, James Jones and Paul Cary are closest to the MATH record (that is, they are the 2nd and 3rd records in the MATH set as opposed to 3rd and 4th records in the ENGLISH set). Figure 2.43 shows how the sets will be fixed by CA IDMS/DB Audit, by placing James Jones and Paul Cary in the MATH set.

If a record does not have a valid owner pointer and it is equidistant from multiple owner records, it is placed in the set whose owner has the lowest db-key.
The fixed sets are unbroken, but either James Jones or Paul Cary, or both, may belong in the ENGLISH set. You should verify the logical integrity of the sets by examining the After Image Report to see how the error is corrected by CA IDMS/DB Audit.

CA IDMS/DB Audit Parameters

This section presents CA IDMS/DB Audit parameter statements and options. Conventions and syntax are summarized in figure format. Detailed discussions of each parameter option follow a brief description of the parameter statements.

- Notation Conventions and Syntax (see page 70)
- Parameter Statements and Options (see page 74)
- CA IDMS/DB Audit PROCESS Statement (see page 75)
- The AUDIT Statement (see page 78)
- AREA, RECORD, and SET (see page 84)
- The AREA Statement (see page 85)
- The RECORD Statement (see page 87)
Notation Conventions and Syntax

Before using CA IDMS/DB Audit parameter statements, review the following exhibits:

- Figure 3.1 Notation Conventions
- Figure 3.2 Parameter Syntax Rules
- Figure 3.3 Parameter Syntax Summary

**Example** | **Function**
---|---
PROCESS | Keywords appear in UPPERCASE. The minimum required portion of each keyword is UNDERSCORED. If a portion of or an entire keyword is not underscored, you can omit that portion or that keyword.

AEA=area-name | Variables appear in lowercase italics. You must substitute an appropriate value for each variable.

[BEGIN=page-number] | Brackets indicate optional clauses.

/ CRITICAL \PROCESS= < ALL > \NONE / | Braces enclose two or more options. You must select one of the options

<table>
<thead>
<tr>
<th>Item</th>
<th>Rule</th>
</tr>
</thead>
</table>
| Order of Parameter Statements | You must enter the PROCESS statement first. You can enter all other parameter statements in any order. Within a parameter statement you can enter options in any order.
<p>| Continuing a Parameter Statement | To continue a parameter statement onto the next record, key in a trailing comma. Do not split a keyword phrase between two records. |</p>
<table>
<thead>
<tr>
<th>Item</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entering Blanks In Parameter Statements</td>
<td>You can enter blanks (character spaces) between keywords to improve readability in a parameter statement without affecting processing. Do not, however, embed blanks within a keyword or value field.</td>
</tr>
<tr>
<td>Entering Parameter Statements</td>
<td>On an 80-character input record, you must enter all parameter syntax between positions 1 through 72.</td>
</tr>
<tr>
<td>Comments</td>
<td>Enter an asterisk (*) in column 1 to indicate a comment.</td>
</tr>
</tbody>
</table>

*Figure 3.2: Parameter Syntax Rules*
IDMSDB--Notation Conventions and Syntax

Figure 3.3a: CA IDMS/DB Audit Parameter Syntax Summary
IDMSDB--Notation Conventions and Syntax (2)

Figure 3.3b: CA IDMS/DB Audit Parameter Syntax Summary
Parameter Statements and Options

Contents
- PROCESS Statement (see page 74)
- AUDIT Statement (see page 74)
- AREA, RECORD, and SET Statements (see page 74)
- FIX Statement (see page 75)
- REPORTS Statement (see page 75)

As the syntax summary on the preceding pages displays, each CA IDMS/DB Audit parameter statement contains a number of options. This section briefly describes the function of these statements and their options. The next section presents a detailed description of each statement and its associated options.

The function of a parameter option is the same for both Standard and QuickCheck processing, unless otherwise noted. Any differences are clearly identified and discussed separately.

PROCESS Statement

The PROCESS statement initiates CA IDMS/DB Audit processing and identifies the subschema that describes the database to be audited. The PROCESS statement is the only parameter statement that is required, and it must be entered first. In addition to the PROCESS statement, you must also include at least one AUDIT, FIX, or REPORT statement with each run.

PROCESS statement options enable you to specify the READYMODE for the database and to indicate which SORT messages to display on the Audit Report. Additional options initiate inclusive or exclusive processing, specify the audit type as either Standard or QuickCheck, and enable you to check your parameters before auditing begins.

AUDIT Statement

The AUDIT statement specifies the scope of the audit.

AUDIT options provide exact instructions on types of errors or data structures to audit: CALC sets, user-defined sets, index sets, pages, and record or set disconnections. By specifying additional options with CALC sets or USERSETS, you can limit processing even further.

AREA, RECORD, and SET Statements

The AREA, RECORD, and SET statements are used in conjunction with the AUDIT statement to designate which parts of the database to process.

AREA, RECORD, and SET options direct CA IDMS/DB Audit to include or exclude entire areas, parts of areas, and specific records or sets when auditing.
**FIX Statement**

The FIX statement enables you to request either simulated or actual correction of most types of errors. Simulated corrections allow you to examine the Integrity Analysis Report and review the projected results without actually updating the database.

FIX options specify which types of errors to correct: CALC or set errors, integrated index orphans, pointer errors, or page header and footer errors. You can apply corrections either to errors found in a previous audit, or to errors found in the current audit.

You can also use one of the FIX options to physically delete logically deleted records (LDELS).

**REPORTS Statement**

Every execution of CA IDMS/DB Audit results in an Audit Report of processing messages and statistics. The Integrity Analysis Report is optional, however.

The REPORTS statement allows you to specify whether or not to generate an Integrity Analysis Report. This report gives details of the physical integrity errors found in the database.

REPORT options enable you to modify the format of the Integrity Analysis Report. You can display the data in either character form or in hexadecimal and character form. Other options let you control how set errors are displayed. You can also generate a report on a previous audit.

**CA IDMS/DB Audit PROCESS Statement**

This section explains the following topics:

- PROCESS Parameter (see page 76)
- READYMODE Option (see page 76)
- SORTMSGS Option (see page 77)
- STARTWITH Option (see page 77)
- AUDITTYPE Option (see page 77)
- SCAN Option (see page 78)

In order to run CA IDMS/DB Audit, you must enter the PROCESS statement first. The PROCESS statement performs these functions:

- Initiates execution of CA IDMS/DB Audit
- Identifies the subschema
- Indicates the mode in which the database is to be readied for an audit or a fix
- Indicates the type of SORT messages to be displayed on the Audit Report
• Initiates inclusive or exclusive processing
• Specifies the audit type as either Standard or QuickCheck
• Scans your parameters for errors before auditing begins

```
,READYMODE=< UPDATE >
 \ RETRIEVAL /
 / CRITICAL \\
,SORTMSG=< ALL >
 \ NONE /

PROCESS, SUBSCHEMA=name ,STARTWITH=< ALL >
 \ NONE /
 / STANDARD \\
,AUDITTYPE=< QUICKCHECK>
 \ /

[,SCAN ]
```

*Figure 3.4: PROCESS Statement Syntax*

**PROCESS Parameter**

`PROCESS, SUBSCHEMA=name`

PROCESS initiates execution of CA IDMS/DB Audit and indicates that processing options follow. CA IDMS/DB Audit accesses and loads the subschema and its related DMCL from the load/core image library.

SUBSCHEMA identifies the subschema that describes the database to be processed. If you select an AUDIT, you can exclude specific areas, sets, or records by entering AREA, SET, or RECORD statements.

**Default:** There is no default value. You must enter a valid subschema name.

**Rules:**

- If you are auditing a multi-member set, you must include all member records of that set in the subschema; do not exclude any members from the audit. The CALC set is a multi-member set.
- All records and sets to be audited must allow the DML commands FIND, GET, MODIFY, and ERASE.
- The subschema cannot contain native VSAM records.

**READYMODE Option**

```
,READYMODE=< UPDATE >
 \ RETRIEVAL /
```

READYMODE specifies the mode in which CA IDMS/DB Audit prepares the database for processing.
UPDATE readies the database in CA IDMS's exclusive update mode. This mode prevents concurrent operations of any kind in the area. If you specify FIX=UPDATE, you must also specify READYMODE=UPDATE.

RETRIEVAL readies the database in CA IDMS's shared retrieval mode. Do not use this mode if you are specifying FIX=UPDATE. You can, however, run other jobs against the area at the same time. If another job is updating the database, CA IDMS/DB Audit may report errors that do not actually exist.

Default: UPDATE

SORTMSGs Option

/ CRITICAL \,SORTMSGs= ALL \,NONE /

SORTMSGs specifies which SORT program messages to display on the Audit Report.

CRITICAL displays only those messages that impede the execution of the SORT program.

ALL displays all SORT messages.

NONE displays no SORT messages.

Default: CRITICAL

STARTWITH Option

,STARTWITH= ALL \,NONE /

STARTWITH initiates either inclusive or exclusive processing.

ALL presumes that the entire database, as defined by the subschema, is to be audited. Optionally, you can then exclude data from processing by specifying AREA, RECORD, and SET statements.

NONE presumes that none of the database, as defined by the subschema, is to be audited. You include data for processing with AREA, RECORD, and SET statements. If you specify STARTWITH=NONE, then you must specify AREA, RECORD, or SET statements.

Default: ALL

AUDITTYPE Option

/ STANDARD \,AUDITTYPE= QUICKCHECK>

AUDITTYPE specifies the audit type as either STANDARD or QUICKCHECK.

QUICKCHECK auditing finds most common database errors with much less overhead than Standard auditing.
STANDARD auditing is the most thorough auditing method. It finds all physical integrity errors, but it also requires longer processing time.

**Default**: STANDARD

**SCAN Option**

```
,SCAN
```

The SCAN option gives you the opportunity to check and correct your parameter statements *before* processing begins. When you specify SCAN, CA IDMS/DB Audit performs parameter validation and reports on how areas, records, and sets are going to be audited based on the options specified.

**Default**: There is no default value.

**The AUDIT Statement**

The optional AUDIT statement specifies the types of physical integrity errors for CA IDMS/DB Audit to find. You can enter only one AUDIT statement for each execution of CA IDMS/DB Audit. The single AUDIT statement indicates which of the six types of auditing to perform. This statement can also include a limit on the number of errors to be detected so that CA IDMS/DB Audit terminates processing when it finds the specified number of errors.

- AUDIT Parameter (see page 79)
- PAGES Option (see page 79)
- INDEX Option (see page 80)
- DISCONNECT Option (see page 80)
- CALCSETS Option (see page 81)
- userSets Option (see page 81)
- SETLIMIT Option (see page 81)
- DBKEYTBL Option (see page 82)
- CORE and DISK Options (see page 82)
  - Storage for the Db-key Table (see page 82)
  - Storage for the Db-key Save Table (see page 83)
  - When to Use Core Storage (see page 83)
  - When to Use Disk Storage (see page 83)
  - Report of Allocated Storage (see page 83)
- WALKNEXT Option (see page 83)
- STOPAFTER Option (see page 84)

You can use AREA, RECORD, and SET statements with the AUDIT statement to include or exclude parts of the database described in the subschema. Descriptions and syntax of these statements follow this section about the AUDIT statement.
AUDIT Parameter

The AUDIT parameter initiates audit processing according to the audit options included in the statement. If you specify the AUDIT parameter, you must select one or more of its options: PAGES, INDEX, DISCONNECT, CALCSETS, or USERSETS.

As a result of audit processing, CA IDMS/DB Audit writes extract records to a general extract file and, optionally, to an index extract file, depending on the parameters you specify.

PAGES Option

The PAGES option writes extract records for pages of the database. If you specify the PAGES option, all other options are assumed to be specified without the need for another specification.

Not applicable to QuickCheck
PAGES directs CA IDMS/DB Audit to audit the page numbers in the header and footer of each database page encountered during area sweep processing.

**INDEX Option**

INDEX verifies each integrated index entry against its database record occurrence, and looks for index orphans by verifying each upper level node pointer.

INDEX does not check for next, prior, or owner (N/P/O) pointer errors in integrated index sets because this checking is performed during USERSETS processing.

If CA IDMS/DB Audit finds pointer errors, you can fix them during the fix phase.

**QuickCheck:** Do not specify INDEX with QuickCheck.

**DISCONNECT Option**

DISCONNECT specifies which member records are to be searched for set disconnections. The Audit Report displays those sets that are audited for disconnections.

NONE prevents a search for set disconnections.

EXPLICIT searches for set disconnections in:

- member records of mandatory automatic sets
- sets designated DISCONNECT in a SET statement
- records designated DISCONNECT in a RECORD statement

ALL searches for set disconnections in records that are members of OPTIONAL and MANUAL sets as well as MANDATORY AUTOMATIC sets. You can use the ALL option instead of specifying DISCONNECT EXPLICIT for each individual set in the subschema.

DISCONNECT increases run time significantly because CA IDMS/DB Audit generates a large number of I/Os to verify set disconnection.

DISCONNECT is an option in both the AUDIT, RECORD, and SET statements. In order for the DISCONNECT option of the RECORD or SET statement to have any effect, you must specify

DISCONNECT=EXPLICIT with the AUDIT statement.

**QuickCheck:** DISCONNECT processing identifies fully disconnected records only. Partially disconnected records are identified by QuickCheck USERSETS processing.
Standard: DISCONNECT processing identifies both partially disconnected and fully disconnected records.

Default: NONE

CALCSETS Option

The CALCSETS option examines the CALC set for next and prior pointer errors. For CALC set auditing to be complete, your subschema must contain a definition for all CALC records that reside in the area of the database designated for sweep processing. If your subschema does not contain definitions for all CALC sets, processing will be incomplete. If AUDITTYPE=STANDARD is specified, CALC records will also be checked to ensure they are connected to the correct CALC set.

USERSETS Option

USERSETS examines user-defined sets and integrated index sets for next, prior, or owner (N/P/O) pointer errors. The upper level node pointer in integrated index members is not examined during USERSETS processing.

Standard: CA IDMS/DB Audit performs LDEL processing for all sets. CA IDMS/DB Audit walks each set from the owner to determine whether the next, prior, and owner pointers of each record are correct. If you performed a complete area sweep, you can then physically delete these logically deleted records by specifying the LDEL parameter in a FIX=UPDATE statement.

QuickCheck: CA IDMS/DB Audit performs LDEL processing only for sets that contain N/P/O errors. If a set contains one or more N/P/O errors, CA IDMS/DB Audit automatically creates extract records for logically deleted member records (LDELs).

SETLIMIT Option

SETLIMIT specifies the maximum number of member records to retrieve from a set occurrence before assuming that a set loop condition exists. A set loop occurs when a member record's next db-key points back to a previous member record in the set chain.

The SETLIMIT value must be an integer from 1 through 9999999. For most efficient performance, the value should equal the maximum number of member records in any one set occurrence. Specifying too large a value causes unnecessary processing within a set when a set loop exists. Another result of specifying too large a value is unnecessarily large storage requirements for DBKEYTBL and the db-key save table.

SETLIMIT causes CA IDMS/DB Audit to stop processing a set only when it has retrieved the number of members specified by the SETLIMIT value. For this reason, you can detect a set loop much more quickly by specifying DBKEYTBL (explained in the next section).
CA IDMS/DB Audit does not perform any fixing if the SETLIMIT value is reached for any set occurrence.

**Default:** 3000

### DBKEYTBL Option

`,DBKEYTBL`

DBKEYTBL directs CA IDMS/DB Audit to build a table of the db-keys of all member records when auditing a set occurrence. CA IDMS/DB Audit uses this table to detect a set loop. A set loop occurs when a member record's next db-key points back to a previous member in the set chain.

If you specify DBKEYTBL, CA IDMS/DB Audit stops processing a set under either of these two circumstances:

- When CA IDMS/DB Audit retrieves a set member whose db-key is already in the db-key table
- When there is no room to store a db-key in the table.

If you do not specify DBKEYTBL, CA IDMS/DB Audit stops processing a set when it has retrieved the number of members indicated by the SETLIMIT value.

The maximum number of entries DBKEYTBL can hold is 120% of the SETLIMIT value. 80% of SETLIMIT is allocated to the primary portion of the table and 40% of SETLIMIT is allocated to an overflow portion of the table.

Because of the algorithm used for accessing the db-key table, there may be no room for a particular db-key in the table even though the SETLIMIT value has not been reached. Therefore, you may need to specify a SETLIMIT value slightly larger than what appears necessary.

CA IDMS/DB Audit always allocates storage for the db-key table during the audit phase because this table is required when creating extract records for set errors. Consequently, you should always specify DBKEYTBL when auditing CALCSETS or USERSETS.

### CORE and DISK Options

`,CORE \ <DISK`

The CORE and DISK options indicate whether to allocate core storage or disk storage for the db-key table and the db-key save table.

**Default:** CORE

### Storage for the Db-key Table

The amount of storage used for the db-key table depends on the SETLIMIT value. The size of the db-key table is equal to 9.6 times the SETLIMIT value. For example, if the SETLIMIT value is 3000 (the default), the storage size is: $9.6 \times 3000 = 28800$ bytes.
Storage for the Db-key Save Table

The amount of storage used for the db-key save table is equal to 8 times the SETLIMIT value. For example, if the SETLIMIT value is 3000 (default), the storage required for the db-key save table is: 8 * 3000=24000 bytes.

Core storage for the db-key save table is allocated whether or not it is used. The amount of storage actually used depends on the number of next, prior, and owner errors in the part of the database being audited. Disk storage for the db-key save table is used only if N/P/O pointer errors are found.

When to Use Core Storage

Specify CORE whenever the amount of storage used by the db-key table and db-key save table will be available in the CA IDMS/DB Audit region of your computer.

When to Use Disk Storage

If your system does not have enough core storage available, specify DISK. Very large sets, including integrated index sets, require a large SETLIMIT and, therefore, large amounts of storage for the tables. If you specify DISK, you must allocate space for two direct access files. These files are required for index processing whether or not you specify the DBKEYTBL parameter. See Operations [https://docops.ca.com/display/IDMS19/CA+ADS+Alive+Operations] for more information.

Report of Allocated Storage

The amount of core storage allocated for each table is reported on the Processing Messages and Statistics section of the Audit Report. If you specify DISK, the Audit Report indicates that zero bytes of core storage have been allocated.

WALKNEXT Option

WALKNEXT searches for orphan records within sets that are defined for USERSET or CALCSET processing. WALKNEXT examines each record encountered in an area sweep and walks all eligible sets of which the record is a member. If the owner does not point at the record, and none of the members point at the record, it is reported as an orphan of the set. That is, walking the set does not lead to an owner and back to the record.

The WALKNEXT parameter is usually used in a second run after a full-area audit that indicates orphan records. To determine the presence of orphan records, refer to the Audit Report, Audit Statistics. Compare the number reported for MBR RECORD OCCURRENCES plus SET DISCONNECTIONS for each set to the number reported for AREA SWEEP OCCURRENCES plus LOGICAL DELETIONS for each record that is a member of the set. Any difference between these numbers represents a possible orphan record. (If DISCONNECT was specified, the difference proves the presence of orphans for the set.)

Walknext processing causes an extraordinarily high number of I/Os because every set occurrence is walked as many times as there are member records plus once more for the owner. This type of processing increases operating time significantly.
CA IDMS - 19.0

**QuickCheck:** Do not specify WALKNEXT with QuickCheck. For QuickCheck processing, CA IDMS/DB Audit searches for orphans through the USERSETS option and uses significantly less processing time.

**STOPAFTER Option**

```
,STOPAFTER=number-of-errors
```

STOPAFTER specifies the maximum number of integrity errors which CA IDMS/DB Audit can encounter before terminating all audit processing.

When CA IDMS/DB Audit performs more than one type of audit, it may detect a particular integrity error more than once and increment the count for that error several times. Make the STOPAFTER value large enough to allow for these duplications.

⚠️ **Note:** If you reach the STOPAFTER value during processing, do not attempt any type of fixing because you cannot assume that CA IDMS/DB Audit has detected all integrity errors.

Default: 999999

**AREA, RECORD, and SET**

The AREA, RECORD, and SET statements are optional subsets of the AUDIT statement. Do not specify one of these statements unless you also specify the AUDIT statement.

The following sections describe each statement and its options.
IDMSDB—AREA, RECORD, and SET Statements

Figure 3.6: AREA, RECORD, and SET Statement Syntax

The AREA Statement

The AREA statement is used in conjunction with the PROCESS statement’s STARTWITH option to include or exclude an area from being audited. You can also use the AREA statement to sweep less than the full page range of an area. You must specify the AREA statement in conjunction with the AUDIT statement.

- AREA Parameter (see page 85)
- BEGIN and END Options (see page 86)
- NOACCESS Option (see page 86)
- NOSWEEP Option (see page 86)
- SWEEP Option (see page 87)

AREA Parameter

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

AREA defines the name of the area to include or exclude in processing.
BEGIN and END Options

\texttt{\textasciitilde BEGIN=page-number}

BEGIN specifies the page where CA IDMS/DB Audit begins an area sweep.

\textbf{Default:} The first page number of the area.

\texttt{\textasciitilde END=page-number}

END specifies the page where CA IDMS/DB Audit ends an area sweep.

\textbf{Default:} The last page number of the area.

SWEEP is implied with either BEGIN or END. Therefore, if you specify BEGIN or END, do not specify NOSWEEP or NOACCESS because doing so would create a conflict.

\textbf{QuickCheck:} You can specify BEGIN and END for QuickCheck as long as you are not also using the \texttt{USERSETS} or \texttt{CALCSETS} options. If you specify QuickCheck \texttt{USERSETS} or \texttt{CALCSETS}, do not specify BEGIN or END pages because QuickCheck relies on extracts from a complete area.

NOACCESS Option

\texttt{\textasciitilde NOACCESS}

The NOACCESS option omits from processing all records in the specified area. Because CA IDMS/DB Audit does not ready the specified area, it also does not audit any sets owned by the area records, nor any sets where those records are members.

You will get an error message if you specify NOACCESS along with the DISCONNECT option in either:

- A RECORD statement for a record in the NOACCESS area
- A SET statement for a set owned by a record in the NOACCESS area

Do not specify NOACCESS in conjunction with BEGIN or END.

NOSWEEP Option

\texttt{\textasciitilde NOSWEEP}

NOSWEEP prevents CA IDMS/DB Audit from processing sets owned by records in the designated area.

NOSWEEP impacts set owners only; it does not affect set members. If access is initiated from an area being swept, NOSWEEP allows each record in the area to be available.

If you specify NOSWEEP with either \texttt{USERSETS} or \texttt{CALCSETS}, NOSWEEP prevents CA IDMS/DB Audit from processing sets owned by NOSWEEP records.

Do not specify NOSWEEP in conjunction with BEGIN or END.
SWEEP Option

SWEEP causes the ACCESS of each record type residing in the area and the AUDIT of each set owned by records in the designated area.

SWEEP also causes the ACCESS of each set member record and, with QuickCheck, the SWEEP of each area containing these members.

Do not specify SWEEP in conjunction with BEGIN or END.

The RECORD Statement

The RECORD statement is used in conjunction with the PROCESS statement’s STARTWITH option to include or exclude a record type from begin audited. You can also use the RECORD statement to report a record disconnected from a set that is not mandatory automatic. You must specify the RECORD statement in conjunction with the AUDIT statement.

- RECORD Parameter (see page 87)
- DISCONNECT Option (see page 87)
- NOACCESS Option (see page 88)
- NOSWEEP Option (see page 88)
- SWEEP Option (see page 88)

RECORD Parameter

RECORD=record-name

RECORD specifies the name of the record type to include or exclude from auditing.

DISCONNECT Option

DISCONNECT reports all records of the specified type that are disconnected from any set that is not mandatory automatic.

In order to use the RECORD,DISCONNECT option, you must specify DISCONNECT=EXPLICIT in the AUDIT statement. CA IDMS/DB Audit automatically reports a record that is disconnected from a mandatory automatic set when you specify either DISCONNECT=EXPLICIT or DISCONNECT=ALL.

Do not specify DISCONNECT and NOACCESS together in the RECORD statement.

Specifying DISCONNECT=NONE in the AUDIT statement and DISCONNECT in the RECORD statement produces a conflict and results in an error message.
NOACCESS Option

NOACCESS omits the designated record from auditing. Therefore, CA IDMS/DB Audit does not audit any sets owned by this record nor any sets where the record is a member.

Do not specify DISCONNECT and NOACCESS together in the RECORD statement.

Do not specify NOACCESS in conjunction with BEGIN or END.

NOSWEEP Option

NOSWEEP prevents CA IDMS/DB Audit from processing the specified record when sweeping the area where the record resides. Therefore, any sets owned by this record are also not audited. NOSWEEP, however, does allow each record of the specified type to be audited if processing is initiated from another area being swept.

If you specify NOSWEEP with either USERSETS or CALCSETS, NOSWEEP prevents CA IDMS/DB Audit from processing sets owned by NOSWEEP records.

Do not specify NOSWEEP in conjunction with BEGIN or END.

SWEEP Option

SWEEP directs CA IDMS/DB Audit to ACCESS the designated record and SWEEP the area in which it resides. Sets owned by these records are also audited.

SWEEP also causes the ACCESS of each set member record and, with QuickCheck, the SWEEP of each area containing these members.

Do not specify SWEEP in conjunction with BEGIN or END.

The SET Statement

The SET statement is used in conjunction with the PROCESS statement's STARTWITH option to include or exclude a set from being audited. You must specify the SET statement in conjunction with the AUDIT statement.

You can also use the SET statement to report a member record that is disconnected from a set occurrence that is not mandatory automatic. To do so, specify the SET statement in conjunction with the AUDIT statement's DISCONNECT=EXPLICIT option.
SET Parameter

\texttt{SET=set-name}

SET specifies the name of the set to include or exclude from auditing.

DISCONNECT Option

\texttt{DISCONNECT}

DISCONNECT locates all records disconnected from the specified set. You must specify \texttt{DISCONNECT=EXPLICIT} in the \texttt{AUDIT} statement for the \texttt{DISCONNECT} option of the \texttt{SET} statement to have any effect. Do not specify \texttt{DISCONNECT} for mandatory automatic sets because records disconnected from mandatory automatic sets are automatically reported unless you specify \texttt{DISCONNECT=NULL} on the \texttt{AUDIT} statement.

Specifying \texttt{DISCONNECT=NULL} in the \texttt{AUDIT} statement and \texttt{DISCONNECT} in the \texttt{SET} statement produces a conflict and results in an error message.

AUDIT and NOAUDIT Options

\texttt{AUDIT}

AUDIT processes the specified set for integrity errors. If you specify \texttt{SET}, \texttt{DISCONNECT}, you must also specify \texttt{AUDIT}. \texttt{AUDIT} causes the \texttt{ACCESS} of the owner and members of the set. \texttt{AUDIT} also causes \texttt{SWEEP} of the area containing the owner, and, with \texttt{QuickCheck}, \texttt{SWEEP} of the area containing the members.

\texttt{NOAUDIT}

NOAUDIT prevents CA IDMS/DB Audit from processing the specified set. If you specify NOAUDIT, do not specify \texttt{SET}, \texttt{DISCONNECT}.

The FIX Statement

The FIX statement indicates the types of fix processing for CA IDMS/DB Audit to simulate or perform. Simulated fixing allows you to examine the projected results on the Integrity Analysis Report without actually updating the database. FIX options specify which types of errors to correct: CALC or set errors, integrated index orphans, pointer errors, or page header and footer errors. You can also use the FIX statement to physically delete logically deleted records (LDELS).

- FIX Parameter (see page 90)
- PAGES Option (see page 90)
- INDEXORPHANS Option (see page 90)
- CALCSETS Option (see page 91)
- USERSET Options (see page 91)
- LDEL Option (see page 91)
You can apply corrections either to errors found in a previous audit or to errors found in the current audit. You can specify the FIX statement without an AUDIT statement only if your JCL specifies extract files that were created during a previous run of the audit phase. If CA IDMS/DB Audit is to fix integrated index orphans that were detected in a previous run of AUDIT, you must also include the index extract file in your JCL.

You cannot specify more than one FIX statement for each run of CA IDMS/DB Audit.

```
/ SIMULATE \,PAGES
FIX = < UPDATE > ,INDEXORPHANS
    ,CALCSETS
    ,USERSSETS
    ,DEL
```

Figure 3.7: FIX Statement Syntax

**FIX Parameter**

```
/ SIMULATE \,FIX = < UPDATE >
```

The FIX parameter initiates either simulated or actual corrections of the database.

SIMULATE displays the intended corrections of detected integrity errors without actually applying the corrections to the database. You can inspect the corrections before updating the database in a subsequent audit run.

UPDATE applies corrections to integrity errors detected in the audit phase by updating the database. If you specify UPDATE in the FIX statement, you must also specify READYMODE=UPDATE in the PROCESS statement.

It is best to specify FIX=SIMULATE before using FIX=UPDATE. If the proposed corrections appear acceptable, you can then correct the detected errors by making a follow-up run using the extract files created during the audit phase.

⚠️ **Note:** Be sure not to apply any updates to the database between running the audit and fix phases. Such updates could change the results of the fix processing or cause the fix phase to terminate abnormally.

**PAGES Option**

```
/PAGES
```

PAGES corrects errors detected within the page header and page footer. If you specify PAGES in the FIX statement, you must also specify PAGES in the audit phase.

**INDEXORPHANS Option**

```
/INDEXORPHANS
```
INDEXORPHANS corrects the upper level pointers in integrated index orphans and decreases the appropriate orphan counts. Each corrected upper level pointer points to the correct SR8 record at the next height level. CA IDMS/DB Audit decreases the orphan counts in upper level SR8 records, starting with the upper level record designated by the old db-key (the one in error) and continuing to, but not including, the upper level record designated by the corrected upper level pointer. INDEX must have been specified in the audit phase.

If you specify INDEXORPHANS in the FIX statement, but do not specify an AUDIT statement, then you must include both the general and the index extract files as input files in the JCL.

**QuickCheck**

Do not specify INDEXORPHANS with QuickCheck.

**CALCSETS Option**

CALCSETS corrects integrity errors detected in the CALC set. If you specify CALCSETS in the FIX statement, you must also must have specified CALCSETS in the AUDIT statement.

**USERSETS Option**

USERSETS corrects errors detected in all audited user-defined sets. In addition, USERSETS corrects next, prior, and owner pointers of integrated index sets.

**LDEL Option**

LDEL processing uses any logically deleted (LDEL) extracts created by USERSETS processing to physically delete LDELS if possible.

LDEL records are not deleted in sets where integrity errors are found unless you also specified USERSETS in both the AUDIT and FIX statements. LDEL records that are fully disconnected are not deleted.

If you perform a partial area audit, logically deleted records may be detected through an area sweep, but not when walking a set occurrence. In this case, logically deleted records are not physically deleted during the fix phase.

**QuickCheck**

Both QuickCheck and Standard USERSETS processing create LDEL record extracts for fixing. However, because QuickCheck does not walk sets that are free of error, QuickCheck USERSETS produces LDEL record extracts only for those LDELS found in user-defined sets with broken N/P/O pointers.
Figure 3.8 summarizes each FIX option, the extract file required as input for the correction, and the type of errors corrected.

<table>
<thead>
<tr>
<th>FIX Options</th>
<th>Extract Files Required</th>
<th>Errors Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAGES</td>
<td>General</td>
<td>Page errors</td>
</tr>
<tr>
<td>INDEXORPHANS</td>
<td>General and Index</td>
<td>Integrated index orphans</td>
</tr>
<tr>
<td>CALCSETS</td>
<td>General</td>
<td>Integrity errors in CALC sets</td>
</tr>
<tr>
<td>USERSETS</td>
<td>General</td>
<td>N/P/O pointer errors in integrated index sets</td>
</tr>
<tr>
<td>USERSETS</td>
<td>General</td>
<td>Integrity errors in user-defined sets</td>
</tr>
<tr>
<td>LDEL</td>
<td>General</td>
<td>LDELs in user-defined sets with no integrity errors</td>
</tr>
<tr>
<td>USERSETS</td>
<td>General</td>
<td>LDELs in user-defined sets with integrity errors</td>
</tr>
</tbody>
</table>

*Figure 3.8: Summary of FIX Options*

The REPORTS Statement

The REPORTS statement allows you to specify whether or not to generate an Integrity Analysis Report. You can also select the format in which record data is displayed on the reports and the format in which set related integrity errors are to be displayed.

- REPORTS Parameter (see page 92)
- RECDATA Options (see page 93)
- SETMBRS Options (see page 93)

The REPORTS statement does not influence the Audit Report, which is generated automatically with every audit run. System Output (https://docops.ca.com/display/IDMS19/System+Output) describes the Audit Report in detail.

```
/ YES \            / HEX \                / ALL \ 
REPORTS=< NO > ,RECDATA= < CHAR > ,SERMBRS= < ERRSONLY > \
ONLY / 
```

*Figure 3.9: REPORTS Statement Syntax*

REPORTS Parameter

```
/ YES \REPORTS= < NO > \ ONLY / 
```

The REPORTS parameter indicates whether or not you want to generate an Integrity Analysis Report.

YES causes CA IDMS/DB Audit to generate an Integrity Analysis Report.

NO prevents CA IDMS/DB Audit from generating an Integrity Analysis Report.
ONLY causes CA IDMS/DB Audit to perform only the report phase, but not to perform an audit or a fix, even if you specified an AUDIT or FIX statement. ONLY generates the Integrity Analysis Report and verifies the syntax of all other parameter statements.

You can generate a report of a previous audit by specifying either YES or ONLY. To do so, include the general extract file as input to the current CA IDMS/DB Audit job. If you specified INDEX in the previous audit, you must also include the index extract file in your JCL. Then specify a PROCESS statement without any AUDIT, FIX, or REPORTS statements.

**Default:** YES

### RECDATA Options

/HEX/RECDATA=CHAR

RECDATA specifies the format of set related errors on the Integrity Analysis Report. RECDATA does not affect the format of page error displays, which shows both hexadecimal and decimal form.

HEX displays data in both hexadecimal and character display formats. Db-keys are displayed in both decimal and hexadecimal formats. Three lines of data are generated for each record displayed.

CHAR displays data in character notation only (alphanumeric and special characters). Binary data is not correctly displayed in this format. Db-keys are displayed in decimal only. One line of data is generated for each record displayed.

**Default:** CHAR

### SETMBRS Options

/ALL/SERMBS=ERRSONLY

SETMBRS specifies the report format of sets that contain errors.

ALL displays all of the member records in a set, even if only one record is in error.

ERRSONLY displays only the error records in a set rather than every record in the set. For each set that contains an integrity error, ERRSONLY displays:

- The owner record
- Each set member that is in error
- The prior and next member records for each member in error

**Default:** ALL
CA IDMS/DB Audit System Output

This section describes the two types of reports produced by CA IDMS/DB Audit: the Audit Report and the Integrity Analysis Report. Included in the discussion of the Integrity Analysis Report is a step-by-step example showing how to diagram a set in error.

CA IDMS/DB Audit produces the following Reports. The format of these reports is the same whether you use QuickCheck or Standard auditing.

Audit Report

The Audit Report lists error, informative, and warning messages. It consists of six reporting sections that summarize all audit processing.

A variation of the Audit Report is available to users through pre-audit scanning. Rather than reporting on the processing that has already taken place, an Audit Report produced by pre-audit scanning enables you to analyze your parameters before processing begins.

Integrity Analysis Report

The Integrity Analysis Report consists of two report sections that detail all physical integrity errors found and show the state of the database after CA IDMS/DB Audit has fixed those errors.

Pre-Audit Scan

Contents

- Processing Without SCAN (see page 95)
- Processing With SCAN (see page 95)

CA IDMS/DB Audit enables you to preview your audit instructions before processing begins. Pre-audit scanning analyzes the parameters you specify and provides you with the following feedback:

- Complete parameter validation

- Short description of processing options in effect

- Parameter-only processing and reporting

- Identification and matching of subschema areas, records, and sets with the processing options that affect them

If you need to correct syntax errors or modify your parameters in any way, you can do so before execution takes place.
To initiate pre-audit scanning, specify the SCAN option with the PROCESS statement.

The following section compares the sequence of events and the output if you generate an Audit Report with or without pre-audit scanning.

**Processing Without SCAN**

When you initiate audit processing without the SCAN option, the sequence of events is:

1. User specifies parameters without the SCAN option
2. CA IDMS/DB Audit produces three Audit Report sections as output:
   - Parameter Validation
   - Processing Options
   - Areas/Records/Sets
3. CA IDMS/DB Audit begins audit processing and simultaneously produces the Audit Report section Runtime Status/Statistics
4. After processing ends, CA IDMS/DB Audit produces the final two Audit Report sections:
   - Audit Statistics
   - Fix Statistics

**Processing With SCAN**

The major advantage of processing with the SCAN option is that you have an opportunity to review your parameters and make corrections before audit execution takes place. The SCAN option causes events to take place in the following sequence:

1. User specifies parameters with the SCAN option
2. CA IDMS/DB Audit produces three Audit Report sections as output:
   - Parameter Validation
   - Processing Options
   - Areas/Records/Sets
3. User modifies parameters as needed
4. User initiates parameter processing again with SCAN to verify that all parameter corrections have been properly made
5. If CA IDMS/DB Audit detects no further errors, user imitates actual audit processing by specifying the same parameters without SCAN.

Figure 4.7 presents a simplified example of SCAN output. The first report section, Parameter Validation, lists the parameters you specified and indicates if CA IDMS/DB Audit detects any parameter errors. The Processing Options section individually identifies each parameter option in the left column and provides a brief description of that option on the right. Finally, the Areas/Records/Sets section describes the impact of all parameters against the designated parts of the subschema.

**PARAMETER VALIDATION**

**PARAMETER PROCESSING OPTIONS**

**AREAS/RECORDS/SETS**

---

**PROCESS, SCAN, AUDITTYPE=QUICKCHECK, STARTWITH=ALL...**

**AUDIT, USERSETS...**

**GLOBAL PROCESSING DEFAULTS FOR SUBSCHEMA ARE:**

- AREA=Sweep
- RECORD=,Sweep
- SET=,AUDIT

**OPTION...SCAN----------------------------- CHECK PARAMETER ONLY**

**OPTION...AUDITTYPE=QUICKCHECK AUDIT TYPE AREA SWEEP AND SORT**

**OPTION...STARTWITH=ALL------------------------ GLOBAL PROCESSING DEFAULTS FOR SUBSCHEMA ARE:**

- AREA=Sweep
- RECORD=,Sweep
- SET=,AUDIT

**OPTION...AUDIT, USERSETS---------------------- USER SET AUDITING DONE BY QUICKCHECK**

**/OWNER POINTER AUDIT**

- NO USER RECORD TO IXDET AUDITING
- FOR MANDATORY AUTOMATIC SET S IN ERROR ONLY

**OPTION...NOINDEX----------------------------- NO IIX BASIC AUDITING**

**OPTION...NOWALKNEXT------------------------- USER SET ORPHAN DETECTION DONE BY QUICKCHECK**

**AREAS/RECORDS/SETS**

**AREA...DEPT-REGION**

- ... READIED
- AREA WILL BE SWEPT

**RECORD...DEPT**

- ACCESSED
The Integrity Analysis Report

The second report type produced by CA IDMS/DB Audit is the Integrity Analysis Report. This report consists of two report sections:

- Before Image Report Section
- After Image Report Section

The Before Image Report Section lists all physical integrity errors found by CA IDMS/DB Audit during the audit phase.

Before images are the extracts or record identifiers (db-key pointers, next, prior, or owner (N/P/O) pointers, sort keys) that appear on the Integrity Analysis Report.

The After Image Report Section shows the state of the database after CA IDMS/DB Audit has fixed all the physical integrity errors it can fix.

After images also appear on the Integrity Analysis Report and are the same record identifiers (db-key pointers, N/P/O pointers, sort keys) in fixed format—either the way CA IDMS/DB Audit will fix them (simulate) or the way in which CA IDMS/DB Audit has already fixed them (update).

- Diagramming Errors (see page 98)
- Pointer Errors (see page 106)
- Set Errors -- Before Image Report (see page 107)
- Set Errors -- After Image Report (see page 108)
- Set Disconnections (see page 109)
- Integrated Index Pointer Errors (see page 111)
- Integrated Index Orphans (see page 113)
- Integrated Index Orphans Fixed (see page 114)
- Additional Integrated Index Errors (see page 116)
- From/Thru Page Range Errors (see page 119)
- Orphans of Sets Other Than Integrated Index Sets (see page 121)
- Fragmented Record Errors (see page 122)
- LDEL Records Physically Deleted (see page 123)
- Page Errors (see page 124)
To generate the After Image Report without actually performing the fixes, specify SIMULATE in the FIX statement. You can have CA IDMS/DB Audit generate the Simulated After Image Report in one run, and perform the fixes in another separate run without re-auditing the database. To execute a separate fix run, you must save the extract file (or files) created by the audit phase to use later in the fix phase.

Diagramming Errors

The next several pages of this section contain step-by-step instructions that show how to draw a diagram of a set in error by using the information shown on a Before Image Report.

Following the discussion of diagramming errors, there are sample reports showing how various types of errors are reported. The types of errors listed in Figure 4.8 are illustrated in the exhibits indicated.

<table>
<thead>
<tr>
<th>Type of Error</th>
<th>To find this error specify</th>
<th>Report Illustrated in Figure:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pointer</td>
<td>USERSETS CALCSETS</td>
<td>Before After Image 4.11</td>
</tr>
<tr>
<td>Set Integrity Miscellaneous</td>
<td>USERSETS CALCSETS</td>
<td>4.12 4.13</td>
</tr>
<tr>
<td>Set Disconnections</td>
<td>DISCONNECT = ALL DISCONNECT = EXPLICIT</td>
<td>4.14</td>
</tr>
<tr>
<td>IIX Pointers</td>
<td>INDEX</td>
<td>4.15 4.16</td>
</tr>
<tr>
<td>IIX Orphans</td>
<td>INDEX</td>
<td>4.17 4.18</td>
</tr>
<tr>
<td>IIX Integrity Miscellaneous</td>
<td>INDEX</td>
<td>4.19, 4.20</td>
</tr>
<tr>
<td>From/Thru Page Range</td>
<td>USERSETS CALCSETS</td>
<td>4.21</td>
</tr>
<tr>
<td>Orphans of sets that are not IIX</td>
<td>USERSETS, WALKNEXT CALCSETS, WALKNEXT AUDITTYPE = QUICKCHECK</td>
<td>4.22</td>
</tr>
<tr>
<td>Fragmented Records</td>
<td>USERSETS CALCSETS</td>
<td>4.23</td>
</tr>
<tr>
<td>Compressed Records</td>
<td>USERSETS CALCSETS</td>
<td>Not Illustrated Not Illustrated</td>
</tr>
<tr>
<td>LDEL</td>
<td>USERSETS</td>
<td>4.24</td>
</tr>
<tr>
<td>Header Footer</td>
<td>PAGES</td>
<td>4.25 4.26</td>
</tr>
</tbody>
</table>

Figure 4.8: Integrity Analysis Report Illustrations

In order to understand better how CA IDMS/DB Audit detected certain errors, you can draw a picture of the set in error by using the information contained in the Before Image Report. For the following example, see the Before Image Report in Figure 4.9, and follow the step-by-step procedure for drawing the set in error.

- **Step 1.** Message xxx034I indicates integrity errors in the DEPT-TEACHER set occurrence.
**Step 2.** Starting with the OWNER record (88,008-001) the NEXT (first) member is listed as 88,010-001 and the PRIOR (last) member is 88,009-003.

**Step 3.** Message xxxx036I indicates that CA IDMS/DB Audit will walk next from the OWNER record.

**Step 4.** The NEXT record (88,010-001) points to 88,002-001 as the NEXT member in the set and points back to the OWNER record (88,008-001).
xxxx034I INTEGRITY ERRORS DETECTED IN SET...DEPT-TEACHER

PRIOR DBKEY-- --------------------------RECORD DATA--------------------------
DEPT 1000 88,008-001 88,010-001 88,009-003 MATH DEPARTMENT DANNY DUDE 000000209

xxxx036I RECORDS DETECTED WALKING NEXT FROM OWNER . . .

---RECORD NAME-- -ID- --DBKEY-- -NEXT DBKEY- -PRIOR DBKEY- -
OWNER DBKEY-- --------------------------RECORD DATA--------------------------
TEACHER 1001 88,010-001 88,002-001 88,008-001 88,008-003 JOHN SMITH MATH MA/MATH

TEACHER 1001 88,002-001 88,005-002 88,008-003 88,008-
Diagramming Errors--Before Image Report

- **Step 5.** Member 88,002-001 points to 88,005-002 as the NEXT member in the set and points back to record 88,008-003.
\begin{itemize}
  \item \textbf{Step 6.} Error message xxxx068E indicates that the previous step is in error. The error is that record 88,002-001’s PRIOR pointer does not point back to record 88,010-001. At this point, CA IDMS/DB Audit saves db-key 88,002-001 for later processing. See Concepts (see page 19).
  
  \item \textbf{Step 7.} Continuing with the next member in the set (88,005-002), it points to 88,009-003 as the NEXT member in the set and points back to record 88,002-001.
\end{itemize}

\begin{center}
\includegraphics[width=\textwidth]{diagram.png}
\end{center}

\begin{itemize}
  \item \textbf{Step 8.} Member 88,009-003 points to the OWNER record (88,008-001) as the NEXT record in the set, and points back to record 88,005-002. At this point, CA IDMS/DB Audit stops walking the set in the next direction.
\end{itemize}
IDMSDB--Diagramming Errors (5)

- **Step 9.** Message xxxx038I indicates that CA IDMS/DB Audit has retrieved db-key 88,002-001 and will start walking the set in the prior direction. The first record found is 88,008-003. This record points to record 88,002-001 as the NEXT record in the set, and points to record 88,010-001 as the PRIOR record in the set.
**Step 10.** Message xxxx039I indicates that this is as far as CA IDMS/DB Audit can process this set occurrence because record 88,008-003 points to a record that was already processed. This step completes the diagramming of the indicated error.

In Figure 4.10 is a Simulated After Image Report that shows how CA IDMS/DB Audit would fix the error that was diagramed.

The following diagram shows how the fixed set occurrence will look. The physical integrity error is fixed. The logical integrity should now be examined. See Logical Integrity (see page 67) for an example of a logical integrity error.
IDMSDB--Diagramming Errors (7)

CA-TOOLS /DB AUDIT
RELEASE DATE PAGE
CA IDMS /DB AUDIT
INTEGRITY ANALYSIS REPORT

Rnn.nn
mm/dd
hh:mm:ss

************************************************************************************
********************************
******************************************* SIMULATED AFTER IMG
E ********************************
************************************************************************************

xxxx035I FIXING PERFORMED ON SET...DEPT-TEACHER

---OWNER RECORD-- ID-- DBKEY-- NEXT DBKEY--
PRIOR DBKEY-- ------------------RECORD DATA------------------

DEPT 1000 88,008-001 88,010-001 88,009-003 MATH DEPARTMENT DANNY DUDE 0000000208

---RECORD NAME-- ID-- DBKEY-- NEXT DBKEY-- PRIOR DBKEY--
OWNER DBKEY-- ------------------RECORD DATA------------------

TEACHER 1001 88,010-001 88,008-003 88,008-001 88,008-001 JOHNN SMITH MATH MA/MATH
TEACHER 1001 88,008-003 88,002-001 88,010-001 88,008-001
001 00003 THOMAS WHITE MATH BS/MATH
TEACHER 1001 88,002-001 88,005-002 88,008-003 88,008-001
001 00004 JACK WORD MATH BS/MATH
TEACHER 1001 88,005-002 88,009-003 88,002-001 88,008-001
001 00007 DANNY DUDE MATH BS/MATH
TEACHER 1001 88,009-003 88,008-001 88,005-002
00029 CLARE CLIFFORD MATH BS,MS,P

Figure 4.10: Simulated After Image Report

Pointer Errors

An example of a record that contains an invalid prior db-key is shown in Figure 4.11. Records displayed on this section of the report are grouped by set name.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates the set found to contain pointer errors.

RECORD NAME - Name of the record.

ID - Record identifier.

DBKEY - Db-key of the database record.

NEXT DBKEY - Next pointer.

PRIOR DBKEY - Prior pointer (if present).

OWNER DBKEY - Owner pointer (if present).

RECORD DATA - First 45 characters of data.

Error messages - Indicate the types of errors detected.

---------RECORD NAME--ID--DBKEY--NEXT DBKEY--PRIOR DBKEY--OWNER DBKEY--RECORD DATA---------
Set Errors -- Before Image Report

Examples of the Before Image Report fields for set errors are shown in Figure 4.12.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates the set found to contain pointer errors.

Line 7 - OWNER record information:

- **OWNER RECORD** - Name of the owner.
- **ID** - Record identifier.
- **DBKEY** - Db-key of the database record.
- **NEXT DBKEY** - Next pointer.
- **PRIOR DBKEY** - Prior pointer (if present).
- **RECORD DATA** - First 59 characters of data.

Line 10 - Member record information:

- **RECORD NAME** - Name of the record.
- **ID** - Record identifier.
- **DBKEY** - Db-key of the database record.
- **NEXT DBKEY** - Next pointer.
- **PRIOR DBKEY** - Prior db-key (if present).
- **OWNER DBKEY** - Owner pointer (if present).
- **RECORD DATA** - First 45 characters of data.

Error messages - Indicate the types of errors detected.
Set Errors -- After Image Report

Here is a description of the fields that make up the After Image Report. See Figure 4.13.

**Lines 1 thru 5** - After Image Report header--this example shows a Simulated After Image report banner. This report shows what the database will look like if the fixes are performed by CA IDMS/DB Audit.

⚠️ **Note**: The fixes shown on this report have not been performed. In order for the fixes to be performed, submit another job that specifies FIX=UPDATE,USERSETS.

**Line 6** - Informative message--indicates the set for which a fix was simulated.

**Line 8** - OWNER record information:

- **OWNER RECORD** - Name of the owner.
- **ID** - Record identifier.
- **DBKEY** - Db-key of the database record.
- **NEXT DBKEY** - Next pointer.
- **PRIOR DBKEY** - Prior pointer (if present).
- **RECORD DATA** - First 59 characters of data.

**Line 11** - Member record information:

- **RECORD NAME** - Name of the record.
- **ID** - Record identifier.
- **DBKEY** - Db-key of the database record.
- **NEXT DBKEY** - Next pointer.
- **PRIOR DBKEY** - Prior db-key (if present).
- **OWNER DBKEY** - Owner pointer (if present).

- **RECORD DATA** - First 45 characters of data.

Figure 4.13: Set Errors -- After Image Report

**Set Disconnections**

An example of a record that was fully disconnected from a set is shown in Figures 4.14a and 4.14b. Records displayed on this section of the report are grouped by set name, and are members of either a mandatory automatic or a DISCONNECT set.

Here is a description of the fields that make up the Before Image Report.

**Lines 1 thru 5** - Before Image Report header.

**Line 6** - Informative message--indicates the set name where record was disconnected.

- **RECORD NAME** - Name of the record.
- **ID** - Record identifier.
**DBKEY** - Db-key of the database record.

**RECORD DATA** - First 88 characters of data.

---

**Figure 4.14a: Set Disconnections**

---

**Figure 4.14b: Set Disconnections**
Integrated Index Pointer Errors

An example of an integrated index set that contains pointer errors is shown in Figure 4.15.

CA IDMS/DB Audit searches for this type of error if INDEX is specified in the AUDIT statement.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message—indicates the set found to contain pointer errors.

RECORD NAME - Name of the record.

ID - Record identifier.

DBKEY - Db-key of database record.

NEXT DBKEY - Next pointer.

PRIOR DBKEY - Prior pointer (if present).

OWNER DBKEY - Owner pointer (if present).

RECORD DATA - First portion of data.

Error message - Indicates the type of error detected.

Figure 4.15: Integrated Index Pointer Errors
An example of a report that illustrates a set whose pointer errors have been fixed is shown in exhibit 4.16. CA IDMS/DB Audit fixes this type of error if USERSETS is specified in the FIX statement. (Integrated index pointer errors were detected when INDEX was specified in the AUDIT statement.)

Here is a description of fields that make up the After Image Report.

**Lines 1 thru 5** - After image header--this example shows a Simulated After Image report banner. This report shows what the database looks like if the fixes are performed by CA IDMS/DB Audit.

![Note: The fixes shown on this report have not been performed. In order for the fixes to be performed, submit another job that specifies FIX=UPDATE,USERSETS.](image)

**Line 6** - Informative message--indicates the set that had a simulated fix performed.

**RECORD NAME** - Name of the record.

**ID** - Record identifier.

**DBKEY** - Db-key of the database record.

**NEXT DBKEY** - Next pointer.

**PRIOR DBKEY** - Prior pointer (if present).

**OWNER DBKEY** - Owner pointer (if present).

**RECORD DATA** - First several characters of data.

---

Figure 4.16: Integrated Index Pointer Errors
### Integrated Index Orphans

The report in Figure 4.17 shows a list of records detected as orphans of integrated index sets. A record is an orphan record of an integrated index set if the upper level db-key does not point to the correct record in the level above it.

CA IDMS/DB Audit searches for orphan records of integrated index sets if INDEX is specified in the AUDIT statement.

Here is a description of the fields that make up the Before Image Report.

**Lines 1 thru 5** - Before Image Report header.

**Line 6** - Informative message—indicates that integrated index orphans were found.

**SET NAME** - Integrated index set name.

**RECORD NAME** - Name of the record.

**ID** - Record identifier.

**DBKEY** - Db-key of the record.

**OWNER DBKEY** - Owner pointer.

**UPPER DBKEY** - Db-key of the correct SR8 record in the level above.

**MEM-UPPER-DBKEY** - Upper level pointer of orphan record (member or lower level SR8) -- the pointer in error.

---

<table>
<thead>
<tr>
<th>SET NAME</th>
<th>RECORD NAME</th>
<th>ID</th>
<th>DBKEY</th>
<th>OWNER DBKEY</th>
<th>UPPER DBKEY</th>
<th>MEM-UPPER-DBKEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIX-STUD-ACT-1</td>
<td>SCHEDULE</td>
<td>8107</td>
<td>91,010-0007</td>
<td>91,018-0001</td>
<td>91,018-0013</td>
<td>91,010-0001</td>
</tr>
<tr>
<td>IIX-STUD-ACT-1</td>
<td>SCHEDULE</td>
<td>8107</td>
<td>91,010-0027</td>
<td>91,018-0001</td>
<td>91,018-0013</td>
<td>91,010-0001</td>
</tr>
<tr>
<td>IIX-STUD-LNAME</td>
<td>SR8</td>
<td>0008</td>
<td>92,129-0005</td>
<td>92,129-0001</td>
<td>92,129-0029</td>
<td>92,129-0008</td>
</tr>
<tr>
<td>IIX-STUD-LNAME</td>
<td>SR8</td>
<td>0008</td>
<td>92,129-0011</td>
<td>92,129-0001</td>
<td>92,129-0027</td>
<td>92,129-0007</td>
</tr>
<tr>
<td>IIX-STUD-LNAME</td>
<td>SR8</td>
<td>0008</td>
<td>92,129-0012</td>
<td>92,129-0001</td>
<td>92,129-0024</td>
<td>92,129-0008</td>
</tr>
<tr>
<td>IIX-STUD-LNAME</td>
<td>SR8</td>
<td>0008</td>
<td>92,129-0013</td>
<td>92,129-0001</td>
<td>92,129-0022</td>
<td>92,129-0003</td>
</tr>
</tbody>
</table>

---

*xxx*059I INTEGRATED INDEX ORPHAN RECORDS DETECTED......
Integrated Index Orphans Fixed

Here is a description of the fields that make up the After Image Report. See Figure 4.18.

**Lines 1 thru 5** - After Image Report header--this example shows a Simulated After Image report banner. This report shows what the database will look like if the fixes are performed by CA IDMS/DB Audit.

**Note:** The fixes shown on this report have not been performed. In order for the fixes to be performed, submit another job that specifies `FIX=UPDATE,INDEXORPHANS`.

**Line 6** - Informative message--indicates the set that had a simulated fix performed.

**SET NAME** - Integrated index set name.

**RECORD NAME** - Name of the record.

**ID** - Record identifier.

**DBKEY** - Db-key of the database record.
**OWNER DBKEY** - Owner pointer.

**UPPER DBKEY** - Db-key of the correct SR8 record in the level above.

**MEM-UPPER-DBKEY** - Upper level pointer of record (member or lower level SR8)--the pointer that was fixed.

---

**Figure 4.18: Integrated Index Orphans Fixed**
Additional Integrated Index Errors

Several examples of errors detected in integrated index sets are shown in Figures 4.19a, 4.19b, and 4.19c. These are errors other than orphan or pointer errors. CA IDMS/DB Audit searches for these types of errors if you specify INDEX in the AUDIT statement. The specific type of error on each report is indicated by the error message.

Here is a description of the fields that make up the Before Image Reports.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates the set found to contain errors.

RECORD NAME - Name of the record.

ID - Record identifier.

Record not found error:

DBKEY - SR8 table entry db-key of a record not found.

UPPER DBKEY - Db-key of the SR8 record.

Record type incorrect error:

DBKEY - SR8 table entry db-key of a record of an incorrect type.

UPPER DBKEY - Db-key of SR8 record whose table entry points to the record that is an incorrect type for the set.

SR8 upper level pointer not HIGH-VALUES error:

DBKEY - Db-key of the record whose upper level pointer is incorrect.

UPPER DBKEY - Db-key of the owner of the SR8 record.

Figure 4.19a: Additional Integrated Index Errors
Here is a description of the fields that make up the Before Image Reports shown in Figures 4.20a and 4.20b.

**Symbolic key errors:**

**DBKEY** - Db-key of the record named.

**SYMBOLIC KEY** - Symbolic key of the record named.

**UPPER DBKEY** - Db-key of the SR8 record whose table entries were examined.

**UPPER SYMB KEY** - Symbolic key in the SR8 table entry. This symbolic key does not match the symbolic key in the SYMBOLIC KEY field.

**Sort key sequence errors:**

**DBKEY** - Db-key of SR8 record.

**SORT KEY** - Sort key entry in the SR8 record.

**UPPER DBKEY** - Db-key of the upper level SR8 record whose db-key table entry points to the SR8 record identified in DBKEY field.
**PREV SORT KEY** - Sort key in the table entry in the upper level record identified in the UPPER DBKEY field. The sequence error means that this sort key does not match the sort key in the SORT KEY field.

**Disconnected records:**
List of records disconnected from the indicated integrated index set.

<table>
<thead>
<tr>
<th>CA-TOOLS</th>
<th>DB AUDIT</th>
<th>RELEASE</th>
<th>DATE</th>
<th>TIME</th>
<th>PAGE</th>
<th>INTEGRITY ANALYSIS REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>/yy</td>
<td>hh:mm:ss</td>
<td>0011</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

************************************************************************************
*******************************
****************************************************************
B E F O R E I M A G E
*******************************
************************************************************************************

xxxx063I SORT KEY SEQUENCE ERRORS DETECTED IN INTEGRATED INDEX RECORDS CONTAINED IN SET IX-TCHR-LNAME

SR8 0008 88,008-0082 CLIFF 88,008-
SR8 0008 88,013-0003 HARDI 88,013-
SR8 0010 88,013-0013 GOBEL 88,013-

Figure 4.20c: Additional Integrated Index Errors

From/Thru Page Range Errors

An example of a record that falls outside its designated from/thru page range is shown in Figure 4.21. The record does not participate in a set that was audited and that does not contain other integrity errors.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates a from/thru page range error.

RECORD NAME - Name of the record.

ID - Record identifier.
DBKEY - Db-key of the database record.

FROM PAGE - From page number.

THRU PAGE - Thru page number.

RECORD DATA - First 67 characters of data.

<table>
<thead>
<tr>
<th>DBKEY</th>
<th>FROM PAGE</th>
<th>THRU PAGE</th>
<th>RECORD DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEACHER</td>
<td>0004</td>
<td>1001 88,003-88,002</td>
<td>00012TOM</td>
</tr>
<tr>
<td>TEACHER</td>
<td>006</td>
<td>1001 88,003-88,002</td>
<td>00034KEVIN</td>
</tr>
<tr>
<td>TEACHER</td>
<td>005</td>
<td>1001 88,003-88,002</td>
<td>00026RONNIE</td>
</tr>
<tr>
<td>TEACHER</td>
<td>003 PHD</td>
<td>01058000</td>
<td>1001 88,004-88,002</td>
</tr>
<tr>
<td>TEACHER</td>
<td>002</td>
<td>1001 88,006-88,002</td>
<td>00030JOHN</td>
</tr>
<tr>
<td>TEACHER</td>
<td>002</td>
<td>1001 88,004-88,002</td>
<td>00021JILL</td>
</tr>
<tr>
<td>TEACHER</td>
<td>005</td>
<td>1001 88,015-88,002</td>
<td>00031MAUREEN</td>
</tr>
<tr>
<td>TEACHER</td>
<td>002</td>
<td>1001 88,005-88,002</td>
<td>00033ROGER</td>
</tr>
<tr>
<td>TEACHER</td>
<td>003</td>
<td>1001 88,006-88,002</td>
<td>0003STRACIE</td>
</tr>
<tr>
<td>TEACHER</td>
<td>003 MATH</td>
<td>00950000</td>
<td>1001 88,003-88,002</td>
</tr>
<tr>
<td>TEACHER</td>
<td>001 MATH</td>
<td>00960000</td>
<td>1001 88,012-88,002</td>
</tr>
<tr>
<td>TEACHER</td>
<td>002 MATH</td>
<td>00999900</td>
<td>1001 88,015-88,002</td>
</tr>
<tr>
<td>TEACHER</td>
<td>001</td>
<td>1001 88,011-88,002</td>
<td>00013ARTHUR</td>
</tr>
<tr>
<td>TEACHER</td>
<td>003</td>
<td>1001 88,011-88,002</td>
<td>00018PAT</td>
</tr>
<tr>
<td>TEACHER</td>
<td>002</td>
<td>1001 88,011-88,002</td>
<td>00014DON</td>
</tr>
<tr>
<td>TEACHER</td>
<td>004</td>
<td>1001 88,011-88,002</td>
<td>00019JOE</td>
</tr>
</tbody>
</table>
Orphans of Sets Other Than Integrated Index Sets

An example of a record that could not be associated to any OWNER record in a user set is shown in Figure 4.22. Records can fall into this category only if WALKNEXT was requested.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates that records were detected as orphans.

SET NAME - Set name where record is an orphan.

RECORD NAME - Name of the record.

ID - Record identifier.

DBKEY - Db-key of the database record.

NEXT DBKEY - Next pointer.

PRIOR DBKEY - Prior pointer (if present).

OWNER DBKEY - Owner pointer (if present).

RECORD DATA - First 28 characters of data.
Fragmented Record Errors

An example of a fragmented (variable length) record that could not be properly assembled is shown in Figure 4.23. The record does not participate in a set that was audited and that does not contain other integrity errors.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates a fragmented record assembly error.

RECORD NAME - Name of the record.

ID - Record identifier.

DBKEY - Db-key of the database record.

RECORD DATA - First 88 characters of the CONTROL portion of the data.

Figure 4.23: Fragmented Record Errors
LDEL Records Physically Deleted

An example where a logically deleted (LDEL) record was physically deleted from two sets is shown in Figure 4.24. The record was also erased since it was physically deleted from all sets where it participated as a member.

Here is a description of the fields that make up the After Image Report.

**Lines 1 thru 5** - After Image Report header--this example shows a Simulated After Image Report banner. This report shows what the database will look like if the fixes are performed by CA IDMS/DB Audit.

![Note: The fixes shown on this report have not been performed. In order for the fixes to be performed, submit another job that specifies FIX=UPDATE,LDEL.]

**Line 6** - Informative message--indicates physical deletion of LDEL records.

**SET NAME** - Indicates set name where record was physically deleted.

**SET OWNER DBKEY** - Db-key of the owner of the set where the record was deleted.

**RECORD NAME** - Name of the record.

**ID** - Record identifier.

**DBKEY** - Db-key of the database record.

**Informative message** - Indicates record was erased.

![xxxx044I LDEL RECORDS PHYSICALLY DELETED FROM SET ...]

![xxxx045I RECORD ERASED]

*Figure 4.24: LDEL Records Physically Deleted*
Page Errors

Two examples of the Before Image Report fields for Page Errors illustrating invalid Header/Footer page numbers are shown in Figures 4.25a and 4.25b. Page errors are grouped by area name.

Note: The RECDATA operand does not affect the format of Page Error displays.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates area where page errors were detected.

REQUESTED PAGE NUMBER - Requested page number in hexadecimal and requested page number in decimal.

HEADER PAGE NUMBER - Header page number in hexadecimal and header page number in decimal.

FOOTER PAGE NUMBER - Footer page number in hexadecimal and footer page number in decimal.

Error message - Indicates the type of error detected.

Figure 4.25a: Page Errors
Figure 4.25b: Page Errors

Here is a description of the fields that make up the After Image Report. See Figures 4.26a and 4.26b.

**Lines 1 thru 5** - After Image Report header--this example shows a Simulated After Image Report banner. This report shows what the database will look like if the fixes are performed by CA IDMS/DB Audit.

Note: The fixes shown on this report have not been performed. In order for the fixes to be performed, submit another job that specifies FIX=UPDATE,PAGES.

**Line 6** - Informative message--indicates area where pages were fixed.

**REQUESTED PAGE NUMBER** - Requested page number in hexadecimal and requested page number in decimal.

**HEADER PAGE NUMBER** - Header page number in hexadecimal and header page number in decimal.

**FOOTER PAGE NUMBER** - Footer page number in hexadecimal and footer page number in decimal.

Figure 4.26a: Page Errors

Figure 4.26b: Page Errors
CA IDMS/DB Audit Operations

This section describes auditing methods and operational procedures. It provides information for determining storage requirements and for operating CA IDMS/DB Audit efficiently. It also provides sample JCL for executing CA IDMS/DB Audit in a z/OS, z/VSE, or z/VM environment.

- Database Access (see page 126)
- Modes of Operation (see page 126)
- Inclusive and Exclusive Processing (see page 127)
- Processing Examples (see page 128)
- Operating Preparations (see page 132)
- Performance Factors (see page 134)
- CA IDMS/DB Storage Requirements (see page 142)
- CA IDMS/DB Audit Sample JCL and Key (see page 145)
- z/OS Operations (see page 147)
- z/VSE Operations (see page 154)
- z/VM Operations (see page 158)

Database Access

CA IDMS/DB Audit does not use CA IDMS to access a user database. Instead, CA IDMS/DB Audit uses an access method that was developed to provide for CA IDMS/DB Audit’s unique needs for accessing a CA IDMS database.

All CA IDMS conventions regarding locking the database against access are observed. While CA IDMS/DB Audit is executing, you must ensure that no central version (CV) or local CA IDMS update jobs access the database.

Modes of Operation

Contents
- Using the Extract File (see page 127)

CA IDMS/DB Audit performs auditing, fixing, and reporting functions. You can execute CA IDMS/DB Audit to perform any one or combination of these functions:

- **Mode 1** - AUDIT only
- **Mode 2** - AUDIT and FIX (Simulated or Real)
- **Mode 3** - FIX only (Simulated or Real)
- **Mode 4** - REPORTS only
Modes 1, 2, and 3 produce an Integrity Analysis Report unless you specifically suppress it. Mode 4 always produces an Integrity Analysis Report.

All modes automatically produce an Audit Report; you cannot suppress it.

Using the Extract File

The AUDIT function of CA IDMS/DB Audit generates an extract file. This extract file is used during the FIX and the REPORTS phases. You can run FIX and REPORTS either in the same execution as AUDIT, or in a later execution.

The recommended procedure is to run CA IDMS/DB Audit twice: first in Mode 2 using simulated FIX, and then in Mode 3 using update FIX. This procedure allows you to review the results of the audit with proposed corrections before updating the database. It is important not to change the database between the audit and fix jobs and to retain the extract file.

In a z/OS environment, you can catalog the extract file at the end of the audit and simulated fix job. In all environments, do not delete the extract file or write over it before running the fix phase.

Inclusive and Exclusive Processing

CA IDMS/DB Audit enables you to selectively include or exclude subschema areas, records, or sets from an audit run. You can also mix inclusive and exclusive processing. For example, you can include named areas and exclude records or sets from those areas. You can also include a record and exclude particular sets that the record owns.

The following parameter options influence inclusive and exclusive processing. These parameters are fully explained in the "Parameters" section.

PROCESS Statement: STARTWITH
AREA Statement: NOACCESS/SWEEP/NOSWEEP
BEGIN and END
RECORD Statement: NOACCESS/SWEEP/NOSWEEP
SET Statement: AUDIT/NOAUDIT

To understand inclusive and exclusive processing, you must understand the relationship between STARTWITH and the AREA, RECORD, and SET parameter options.

STARTWITH defines only the starting point of how much is to be audited based on the subschema named in the PROCESS statement. AREA, RECORD, and SET options further identify the scope of processing to be performed.

To initiate inclusive processing, first specify STARTWITH=NONE to exclude all portions of the database from processing. Then supply AREA, RECORD, and SET statements to select portions of the database to be included in processing.
To initiate **exclusive processing**, first specify STARTWITH=ALL to include all portions of the database for processing. Then supply AREA, RECORD, and SET statements to select portions of the database to be excluded from processing. Because STARTWITH=ALL is the default parameter, it need not be explicitly specified. For clarity, this section always specifies STARTWITH=ALL for exclusive processing.

**Mixed processing** combines elements of inclusive and exclusive processing to make very select auditing possible. For example, you could include an area and exclude sets from within that area. You can specify either STARTWITH=ALL or STARTWITH=NONE depending on which portions of the database are to be audited.

### Processing Examples

#### Contents
- Parameter Guidelines (see page 128)
- Inclusive Processing Examples (see page 129)
- Exclusive Processing Examples (see page 130)
- Mixed Processing Examples (see page 131)

This section contains examples of parameter syntax used to initiate various kinds of inclusive, exclusive, and mixed audit processing. The processing examples presented on the following pages are based on the subschema TESTSUB, presented as a data structure diagram in the "Concepts" section of this guide.

### Parameter Guidelines

General guidelines on parameter syntax structure and audit processing follow:

- The STARTWITH option and AREA, RECORD, and SET statements are hierarchical: STARTWITH has the broadest impact, and SET has the narrowest. You can use a lower level parameter to reverse an effect caused by a higher level parameter, thereby mixing inclusive and exclusive processing.

- AREA options propagate to logically related records and sets.

- RECORD options propagate to logically related areas and sets.

- SET options propagate to logically related areas and records.

- Use the positive subparameters (AUDIT and SWEEP) in AREA, RECORD, and SET statements to include elements that were excluded through STARTWITH=NONE.

- Specify negative subparameters (NOACCESS and NOAUDIT) to exclude elements that were included through STARTWITH=ALL.

- NOSWEEP is unique in that it can be considered as either a positive (allows access) or negative (do not sweep) option, depending upon use.
• If you specify NOACCESS for either AREA or RECORD statements, do not specify AUDIT for SET statements that are associated with the same area or record. Similarly, if you specify NOACCESS on an AREA statement, do not specify a record SWEEP or set AUDIT for records and sets that are associated with the area.

• CA IDMS/DB Audit identifies conflicting parameters as either errors or warnings. Processing terminates if any parameter presents a direct conflict between AREA, RECORD, or SET parameters. Audit processing continues when conflicting options produce warnings, but some options may be overridden, especially when conflicts arise during propagation.

• The Audit Report Areas/Records/Sets report section documents the processing that will occur.

• In the PROCESS statement, specify the SCAN option to allow you to review the Areas/Records/Sets section of the Audit Report and verify what processing is to take place before CA IDMS/DB Audit actually begins that processing.

Inclusive Processing Examples

The following examples are based on the subschema TESTSUB and invoke inclusive processing.

Example 1 - Audit one set only

PROCESS, SUBSCHEMA=TESTSUB, STARTWITH=None
AUDIT, USERSETS
SET=ROOM-CLASS, AUDIT

When you audit a single set, you do not have to specify the areas because CA IDMS/DB Audit automatically accesses the areas containing the set. The area LOC-REGION contains the set owner and is automatically set to SWEEP.

If you specify Standard as the audit type, the member area CLASS-REGION is made accessible. If QuickCheck is the audit type, member area CLASS-REGION is set to SWEEP.

Example 2 - Sweep a single area

PROCESS, SUBSCHEMA=TESTSUB, STARTWITH=None
AUDIT, USERSETS
AREA=DEPT-REGION, SWEEP

In this example, only those sets whose owners reside in DEPT-REGION are audited. Specifically, the following sets are audited:

DEPT-TEACHER
DEPT-ACADEMIC
DEPT-GENERAL
PREREQSFOR
PREREQSARE
SUBJECT-CLASS
TEACHER-CLASS

The index (IX) sets would not be audited because their owners are in the INDEX-REGION.

Example 3 - Sweep more than one area

PROCESS, SUBSCHEMA=TESTSUB, STARTWITH=None
AUDIT, USERSETS
All index and user sets except ROOM-CLASS are audited. In this particular case, however, it would be easier to use exclusive processing (STARTWITH=ALL) to exclude the one set ROOM-CLASS.

Example 4 - Audit all sets owned by a single record

PROCESS, SUBSCHEMA=TESTSUB, STARTWITH=NONE
AUDIT, USERSETS
RECORD=SUBJECT, SWEEP

In this case, the following sets are audited:

PREREQSFOR
PREREQSARE
SUBJECT-CLASS

To sweep a record, you do not need to specify SWEEP for the area where the record resides because CA IDMS/DB Audit handles that automatically. All sets that the record owns are swept when you specify record sweep.

Under QuickCheck, sweeping an owned set implies sweeping the area containing the records that participate in the set. Multiple areas may also be swept if the record owns cross-area sets. A cross-area set is a set whose owners and members reside in different areas. Under Standard auditing, sweeping all owned sets may also cause multiple areas to be accessed if the record owns cross-area sets.

Example 5 - Confine processing to specific pages

PROCESS, SUBSCHEMA=TESTSUB, STARTWITH=NONE, AUDITTYPE=STANDARD
AUDIT, USERSETS
AREA=STUDENT-REGION, BEGIN=00010, END=00030

Because BEGIN and END imply SWEEP, you do not need to specify SWEEP.

In this example, the audit type can only be Standard. You cannot specify USERSETS with QuickCheck if you also want to perform partial page range audits with BEGIN and END. For QuickCheck, the USERSETS option requires entire area page ranges to be swept in order to detect errors properly. Therefore, you cannot specify BEGIN or END in QuickCheck when USERSETS is also specified.

The same restriction against specifying BEGIN or END applies to QuickCheck CALCSETS as well.

Exclusive Processing Examples

The examples contained in this section represent parameter syntax for exclusive processing. For clarity, the default STARTWITH=ALL is included in all of the following examples.

Example 1 - Sweep and audit all areas except CLASS-REGION

PROCESS, SUBSCHEMA=TESTSUB, STARTWITH=ALL
AUDIT, USERSETS
AREA=CLASS-REGION, NOACCESS
Any set whose owner or members reside in CLASS-REGION is not audited. Specifically, the following sets are not audited:

- SUBJECT-CLASS
- TEACHER-CLASS
- ROOM-CLASS
- PERIOD-CLASS
- CLASS-SCHEDULE
- STUDENT-SCHEDULE

Example 2 - Do not sweep area CLASS-REGION, but access its member records from cross-area sets

```
PROCESS, SUBSCHEMA=TESTSUB, STARTWITH=ALL, AUDITTYPE=STANDARD
AUDIT, USERSETS
AREA=CLASS-REGION, NOSWEEP
```

A cross-area set is a set whose owners and members reside in different areas. In this example, all sets whose owners reside outside of CLASS-REGION are audited, even if the members reside in CLASS-REGION. Consequently, the only set not audited is CLASS-SCHEDULE.

To satisfy the conditions of Example 2, the audit type must be Standard. Example 3, which follows, demonstrates the effect of specifying QuickCheck as the audit type with all other parameters being the same.

Example 3 - Do not audit owned sets in area CLASS-REGION with QuickCheck

```
PROCESS, SUBSCHEMA=TESTSUB, STARTWITH=ALL, AUDITTYPE=QUICKCHECK
AUDIT, USERSETS
AREA=CLASS-REGION, NOSWEEP
```

Again, in this example, only the CLASS-SCHEDULE set is not audited. Because the audit type is QuickCheck, however, the sets with owner records that point into the NOSWEEP CLASS-REGION from another area must be set either to AUDIT or NOAUDIT. This parameter is required because, by definition, NOSWEEP does not impact member sets, yet QuickCheck must sweep member records if their owners are swept and their sets are audited. One way to audit owned sets that point into CLASS-REGION is to audit one of the sets that point into the CLASS-REGION, thereby forcing the CLASS-REGION to be swept:

```
AREA=CLASS-REGION, NOSWEEP
SET=SUBJECT-CLASS, AUDIT
```

Example 4 - Audit an entire subschema except for sets IX-SUBJ-NAME and IX-STUD-LNAME

```
PROCESS, SUBSCHEMA=TESTSUB, STARTWITH=ALL
AUDIT, USERSETS
SET=IX-SUBJ-NAME, NOAUDIT
SET=IX-STUD-LNAME, NOAUDIT
```

Integrated index sets that are stored in the same area page range must be audited—or not audited—together. If IX-SUBJ-NAME and IX-TCHR-LNAME were in the same page range, you would have no choice but to set them both to AUDIT or to NOAUDIT.

**Mixed Processing Examples**

Mixed processing combines elements of inclusive and exclusive processing. The following examples demonstrate a few ways mixed processing can be used.
Example 1 - Sweep a single area, but do not audit the set TEACHER-CLASS contained in area DEPT-REGION:

PROCESS, SUBSCHEMA=TESTSUB, STARTWITH=NONE
AUDIT, USERSETS
AREA=DEPT-REGION, SWEEP
SET=TEACHER-CLASS, NOAUDIT

In this example, the following sets are audited:

DEPT-TEACHER
DEPT-ACADEMIC
DEPT-GENERAL
PREREQSFOR
PREREQSRARE
SUBJECT-CLASS

If QuickCheck is the audit type, then CLASS-REGION is additionally swept for CLASS records because the set SUBJECT-CLASS is audited.

Example 2 - Sweep and audit all areas, but, within the area DEPT-REGION, audit only the set DEPT-TEACHER

PROCESS, SUBSCHEMA=TESTSUB, STARTWITH=ALL
AUDIT, USERSETS
AREA=DEPT-REGION, NOSWEEP
SET=DEPT-TEACHER, AUDIT

Although the AREA statement specifies NOSWEEP for the area DEPT-REGION, it must be swept in order to audit set DEPT-TEACHER. Other sets in DEPT-REGION are not audited because the audit set overrides the area NOSWEEP for records in DEPT-TEACHER only. However, the area NOSWEEP accomplishes the NOAUDIT for the other sets whose owners are in DEPT-REGION.

Example 3 - Specify NOACCESS in DEPT-REGION and AUDIT DEPT-TEACHER in DEPT-REGION

PROCESS, SUBSCHEMA=TESTSUB, STARTWITH=ALL
AUDIT, USERSETS
AREA=DEPT-REGION, NOACCESS
SET=DEPT-TEACHER, AUDIT

The parameters in this example create a more serious conflict than those in Example 2. In this case, CA IDMS/DB Audit cannot reasonably determine whether area NOACCESS or set AUDIT is more important to the user. Consequently, processing cannot continue and CA IDMS/DB Audit displays an error message.

Operating Preparations

Before running CA IDMS/DB Audit, prepare for processing as follows:

- **Prevent additional updates** - During the time the database is being audited, CA IDMS/DB Audit follows the normal CA IDMS conventions regarding locking the area against access. If you specify READYMODE=RETRIEVAL, you must ensure that no other update jobs run against an area being audited.
  
  If you run CA IDMS/DB Audit twice (once for audit and once for fix), you must prevent any CA
IDMS updates from taking place in the audited areas until after you apply the fixes. If any updates are made between the time the audit is run and the time the fixes are applied, the corrections may not be valid or new integrity errors may be introduced. Similarly, do not perform any updates until you have completed the REPORT function.

- **Back up the database** - CA IDMS/DB Audit does not write before and after images of changed data to a journal file. Therefore, you should back up the database before initiating fix processing and again after fix processing.

Perform the following preparations within CA IDMS/DB Audit:

- **Include index area in the subschema** - If you specify AUDIT INDEX, you must copy the index areas and associated records and sets into the subschema. If you specify INDEX, copy the area containing the index members and owners into the subschema.

- **In the subschema, define all members of sets to be audited** - You cannot successfully audit a set unless you define all the set's member record types in the subschema. This includes all members of the CALC set residing in areas for which CALCSET processing will occur.

- **Specify correct SUBSCHEMA/DMCL and load/core image library** - Be sure to specify the correct subschema for the database you wish to audit. If you specify the wrong subschema, CA IDMS/DB Audit can incorrectly flag a large number of pages as having header or footer errors. Because CA IDMS/DB Audit does not access the subschema/DMCL from a dictionary, the correct copies must be in the job's step library or core image library.

- **Specify RETRIEVAL only for runs with no updates** - In the PROCESS statement, specify RETRIEVAL as the readymode for runs that do not update the database. Legitimate runs are AUDIT, FIX=SIMULATE or AUDIT, REPORT. With READYMODE=RETRIEVAL, CA IDMS/DB Audit does not lock the database, so you must ensure that no other job updates the database. Otherwise, if the database being audited is in the process of being updated, CA IDMS/DB Audit may incorrectly indicate that the database contains integrity errors.

- **Specify UPDATE for actual fixing** - When you are ready to fix the database, specify READYMODE=UPDATE in the job that is to perform the fixing. Be sure that the database to be fixed is not modified between the AUDIT and the FIX or REPORTS jobs.

- **Allow sufficient time to audit the database** - Because CA IDMS/DB Audit may access every record and set occurrence in the database several times, it can take a long time to audit the database. Recognize that in most cases Standard processing takes longer than QuickCheck processing.

- **Specify DISK if sets to be audited are large** - Large sets, including integrated index sets, require large amounts of storage for the tables used in processing. If you specify DISK to contain these tables, you must allocate two VSAM files before executing CA IDMS/DB Audit.

- **Allocate enough space for the DBKEYWORK file** - This step applies to Standard auditing only. The DBKEYWORK file can require a substantial amount of space if you audit integrated indexes whose areas contain many mandatory-automatic (MA) index set members. For every MA index set member, CA IDMS/DB Audit writes two 30-byte records to the DBKEYWORK file. If a member record belongs to more than one index set, two records are written for each set to which it belongs.
- **Allocate enough space for INDXEXTR and INDXWORK files** - The INDXEXTR and INDXWORK files are used for integrated index orphan extract records and for other integrated index error extract records. For each index orphan found, CA IDMS/DB Audit writes one index orphan extract record to the INDXEXTR file. If you are fixing (simulate or update) index orphans, CA IDMS/DB Audit also writes one index orphan extract record to the INDXWORK file. Be sure to allocate enough sort work space for integrated index orphans.

- **Include a sufficient area portion** - CA IDMS/DB Audit allows partial area auditing and fixing. CA IDMS/DB Audit does not allow fixing, however, when you perform a partial area audit and either of the following situations exists:
  - A set containing an integrity error points into a second set with errors, and the owner of the second set is outside the partial area being swept.
  - Records are detected as orphans of a set containing integrity errors, and the owner of the set is outside the partial area being swept.

- **Review the output** - Review the reports CA IDMS/DB Audit generates to understand the errors detected and the way they have been fixed.

Validate the *logical integrity* of any set occurrences CA IDMS/DB Audit fixed. Because CA IDMS/DB Audit cannot analyze logical integrity conditions enforced by user programs, you must correct any logical errors once the physical errors have been fixed. You can use the CA IDMS utility IDMSBCF, FIX PAGE function to fix errors that CA IDMS/DB Audit does not fix. For a list of the types of integrity errors that CA IDMS/DB Audit does not fix, refer to Concepts (see page 19).

### Performance Factors

Several factors affect the performance of CA IDMS/DB Audit:

- **Size of the database** - The number of pages, the number of record occurrences, and the number of set types affect the amount of time required to audit the database

- **Record location mode** - The more member records that are stored VIA the audited set and the less VIA overflow count, the better the performance

- **Auditing method** - QuickCheck auditing is faster than Standard auditing because it requires less database access.

- Estimating USERSETS I/Os with Standard Auditing (see page 135)
- Estimating USERSETS I/Os with QuickCheck Auditing (see page 137)
- Estimating CALCSETS I/Os with Standard or QuickCheck Auditing (see page 137)
- Estimating WALKNEXT I/Os with Standard Auditing (see page 137)
- Estimating DISCONNECT I/Os with Standard Auditing (see page 139)
  - Effect of Specifying Both WALKNEXT and DISCONNECT (see page 141)
- Improving Performance (see page 141)
Estimating USERSETS I/Os with Standard Auditing

You can use the following formula to calculate the approximate number of I/Os to audit USERSETS in your database (when you have not specified WALKNEXT or DISCONNECT). Additional I/Os are required if CA IDMS/DB Audit detects integrity errors.

\[ I/Os = p + n + (s * v) \]

- \( p \) -- represents the number of pages swept
- \( n \) -- represents the number of CALC or DIRECT member record occurrences for sets being audited
- \( s \) -- the number of set occurrences whose VIA members are stored in a different page range than the set owner
- \( v \) -- represents the number of VIA member record types stored in different page ranges than the set owner

Refer to Figure 5.1 for an example. If you were to audit USERSETS only in AREA-1 (AREA-2 is NOSWEEP), you would estimate the number of I/Os as follows:

- 10,000 pages swept
- + 75,000 type E records for set F-E
- + 120,000 type G records for set E-G
- + 120,000 type G records for set H-G
- + 185,000 type D-B set occurrences
- = 510,000 I/Os

\((v=1, \text{because all B records are in AREA-2.})\)

If record type G were stored VIA set E-G, it would take approximately 120,000 fewer I/Os to audit the database for a total of 390,000 I/Os. If, in addition, record type E were stored VIA set F-E, it would take approximately 75,000 fewer I/Os to audit the database for a total of 315,000 I/Os.
IDMSDB--Estimating USERSETS I/Os with Standard Auditing

Figure 5.1: AREA-1 and AREA-2, Estimating USERSETS I/Os
Estimating USERSETS I/Os with QuickCheck Auditing

The number of I/Os required to audit USERSETS during QuickCheck processing is approximately equal to the number of pages swept.

Estimating CALCSETS I/Os with Standard or QuickCheck Auditing

The approximate number of I/Os to audit only CALCSETS should be equal to the number of pages swept, because members of the CALC set are VIA the SR1 record for the page to which the record calculated.

Estimating WALKNEXT I/Os with Standard Auditing

WALKNEXT applies to Standard auditing only; it does not apply to QuickCheck.

WALKNEXT causes each record in the area to be accessed several times. When you specify WALKNEXT, each set occurrence is walked as many times as there are member record occurrences plus once more for the owner. The result is a significant increase in processing time over normal set walk processing. WALKNEXT not only requires numerous I/Os to complete the audit, but also exponentially increases the number of calls to DBIO regardless of whether or not members are stored VIA the sets being audited.

Use the following formula to calculate the approximate number of I/Os required to audit your database with WALKNEXT. Additional I/Os are needed if CA IDMS/DB Audit detects integrity errors.

\[ I/Os = p + (o \times l(1)) \]

- \( p \) -- represents the number of pages swept
- \( o \) -- represents the number of owner records for each CALC or DIRECT set
- \( l \) -- represents the average set length of the CALC or DIRECT set

Figure 5.2 provides an example. If you are auditing USERSETS within AREA-1 with WALKNEXT (set H-G is the only non-VIA set), you would estimate the number of I/Os as follows:

\( l = \frac{120,000}{20,000} = 6 \)

10,000 pages swept
+ 720,000 I/Os for set H-G (20,000 owner records * 36) = 730,000 I/Os

If the same area were audited without WALKNEXT, the number of I/Os would be approximately:

10,000 pages swept
+ 120,000 type G records for set H-G
= 130,000 I/Os
Use the WALKNEXT option only if you know or suspect that there are record occurrences that would not be found during normal set walking. To determine whether your database contains orphan records, refer to The Integrity Analysis Report (see page 97).

IDMSDB—Estimating WALKNEXT I/Os with Standard Auditing

Figure 5.2: AREA-1 and AREA-2, Estimating WALKNEXT I/Os
Estimating DISCONNECT I/Os with Standard Auditing

The DISCONNECT option slows performance because of the large number of records retrieved. CA IDMS/DB Audit examines each record retrieved in an area sweep to see if it is a member of a set that is to be audited for disconnections. If all of such a record’s db-keys for the set being audited are null (equal to high-values), CA IDMS/DB Audit indicates that the record is fully disconnected from the set. If a next, prior, or owner pointer is not null, CA IDMS/DB Audit retrieves the record indicated by the pointer to determine whether it is a valid owner or member of the set being audited.

DISCONNECT not only requires numerous I/Os to complete the audit, but also exponentially increases the number of calls to DBIO regardless of whether or not the sets being audited are VIA or non-VIA (DIRECT or CALC).

You can calculate the approximate number of I/Os required to audit your database for set disconnections (assuming no integrity errors are detected and most of the records are not fully disconnected) using the following formula:

\[ I/Os = p + (m \times k) \]

- \( p \) -- represents the number of pages swept
- \( m \) -- represents the number of member records in the area being swept
- \( k \) -- represents the number of non-VIA member record db-keys in the record’s prefix

For example, in Figure 5.3, auditing for set disconnections in AREA-2 would take approximately:

- 2,500 pages swept
- + 975,000 I/Os for set C-B (325,000 \( \times 3 \))
- = 977,500 I/Os

Disconnect processing is independent of CALCSETS and USERSETS auditing. The number of I/Os generated by disconnect processing is in addition to the number of I/Os generated by CALCSETS and USERSETS auditing (except for the number of I/Os generated by area sweeping).
Effect of Specifying Both \textsc{Walknext} and \textsc{Disconnect}

When you specify both \textsc{Walknext} and \textsc{Disconnect}, walknext processing takes precedence over disconnect processing. During area sweeping, if CA IDMS/DB Audit retrieves a record that is a member of a set being audited for disconnections, it examines the next and prior db-keys. If either the next or the prior db-key is not null, CA IDMS/DB Audit performs walknext processing rather than simply retrieving the records pointed to by the next, prior, and owner db-keys.

Improving Performance

CA IDMS/DB Audit provides several options for improving processing performance:

- Omit one or more areas if the database has several areas.

- Perform a partial area audit. Partial area audits always require less processing time than full area audits, although you must decide what action to take for any sets that cross area boundaries. You can perform a partial area audit in one of the following ways:
  - Select specific set types to audit by including or excluding other areas, records, and sets.
  - Specify \textsc{Begin} and \textsc{End} pages in an \textsc{Area} statement if the database has only one large area (Standard auditing only). You can fix errors in the partial area if the owners of all sets involved are included in the partial area. See \textit{Operating Preparations} (see page 132) for more information.
CA IDMS - 19.0

- Perform a QuickCheck audit rather than a Standard audit.

CA IDMS/DB Storage Requirements

Contents

- Main Storage (see page 142)
- External File Storage (see page 144)
  - EXTRACTS File (see page 144)
  - WORKFILE File (see page 144)
  - DBKYWORK File (see page 144)
  - INDXEXTR and INDXWORK Files (see page 144)
  - SORT File (see page 145)

You need to consider CA IDMS/DB Audit's storage requirements for both main storage and external file storage.

Main Storage

CA IDMS/DB Audit storage requirements vary depending on the size of the subschema table being processed, the maximum database page size, the number of buffers, sort storage requirements, whether or not VSAM is used, and the selected AUDIT options.

After processing a valid PROCESS statement, CA IDMS/DB Audit loads the subschema and DMCL load modules or phases into main storage and builds a pseudo-subschema (PSUB) from the real CA IDMS subschema. Once the pseudo-subschema is built, the real CA IDMS subschema and DMCL are deleted from main storage. After parameter processing is completed, a BIND RUN-UNIT is issued. At this time, all of DBIO's work areas are obtained.

CA IDMS/DB Audit then calculates the amount of storage required for its work areas and issues a GETMAIN/GETVIS to obtain this amount from the region/partition. The amount of main storage not obtained by CA IDMS/DB Audit (and DBIO) is available for operating system use (such as CA IDMS/DB Audit's file buffers, VSAM, and sort).

Note: In a z/VSE environment, all of CA IDMS/DB Audit's and DBIO's work areas are obtained from the GETVIS area, not from the Problem Program Area (PPA). Only the SORT program and associated sort work areas are allocated in the PPA.

Use the formula shown in Figure 5.4 to estimate the amount of virtual storage required to run CA IDMS/DB Audit.

\[ \text{VIRTUAL STORAGE} = \]
<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA IDMS/DB Audit SIZE</td>
<td>Sum of block size times the number of buffers for each non-database file.</td>
</tr>
<tr>
<td>+ CA IDMS/DB Audit Buffers</td>
<td>Approximately 3 times the total size of the subschema and DMCL load modules or core image phases.</td>
</tr>
<tr>
<td>+ Databases Buffers</td>
<td>Maximum page size times number of buffers specified in the DMCL named in the subschema.</td>
</tr>
<tr>
<td>+ Sort Storage</td>
<td>Use your installation’s default sort size.</td>
</tr>
<tr>
<td>+ VSAM Storage</td>
<td>Optional. Used if database has VSAM files or the DISK option is used. Refer to the appropriate system manual to determine the size of the ACB, EXT list, and RPL. If the DISK option is used to store the db-key table and the db-key save table on VSAM disk, add two 4K buffers.</td>
</tr>
<tr>
<td>+ DBKEYTB SETLIMIT value times 9.6 (result is in bytes). Used only if the CORE option is used and auditing CALCSETS and/or USERSETS and/or INDEX. If a SETLIMIT value is not specified, use 28K.</td>
<td></td>
</tr>
<tr>
<td>+ DBKEY Save Table SETLIMIT value times 8 (result is in bytes). Used only if the CORE option is used and auditing CALCSETS and/or USERSETS and/or INDEX. If a SETLIMIT value is not specified, use 24K.</td>
<td></td>
</tr>
<tr>
<td>+ Record Bind Area</td>
<td>Total length of all audited records defined in the subschema (including prefixes) plus an additional 64 bytes for each record being audited. If auditing integrated index sets, add the largest page size of containing SR8 records.</td>
</tr>
<tr>
<td>+ Sort Key Area</td>
<td>Total length of the symbolic keys of all records with sets being audited. This area is the work area required for validating a member record’s symbolic key within a sorted set.</td>
</tr>
<tr>
<td>+ Page Bind Area</td>
<td>Used for page audit. Sum of number of areas audited times 40.</td>
</tr>
<tr>
<td>+ IIX Level Save Area</td>
<td>If auditing integrated index sets, add the maximum SR8 size times the maximum number of index levels.</td>
</tr>
</tbody>
</table>

*Figure 5.4: Formula for Estimating Virtual Storage*
External File Storage

CA IDMS/DB Audit makes use of five sequential files: EXTRACTS, WORKFILE, DBKYWORK, INDXEXTR, and SORT. The following sections explain how to estimate space allocation for each file.

Allocate space only for the files you need for a particular CA IDMS/DB Audit run. You must always allocate space for EXTRACTS and WORKFILE. If you are processing index sets, allocate space for INDXEXTR and INDXWORK. If you are processing mandatory automatic index sets, allocate space for DBKYWORK.

EXTRACTS File

Space allocation for the EXTRACTS file depends on the number of errors detected during the audit phase and the size of the extract record. Determine the size of the extract record using this formula:

\[
\text{Record Size} = 260 + \text{length of longest symbolic key} + 4 \text{ bytes for the record descriptor word (RDW)}
\]

Because it is difficult to estimate the number of errors, you can allocate the EXTRACTS file to tape. In a typical situation, a DASD cylinder allocation of 10,2 should be adequate.

WORKFILE File

Space allocation for the WORKFILE file should be twice the space allocation of the EXTRACTS file. In a typical situation, a DASD cylinder allocation of 20,4 should be adequate.

DBKYWORK File

To estimate the total number of blocks for the DBKYWORK file, use the following formula:

\[
\text{Total Number of Blocks} = (30 \times (\text{total # integrated index member records of MA sets})) \\
\times \frac{2}{\text{block size}}
\]

In a typical situation, a DASD cylinder allocation of 5,2 should be adequate.

INDXEXTR and INDXWORK Files

Space for the INDXEXTR and INDXWORK files is primarily used for integrated index orphan records and, secondarily, for integrated index errors.

If your database contains a small number of member records in integrated index sets, then you can estimate the total blocks using this formula:

\[
\text{Total Number of Blocks} = ((\text{total # integrated index member records}) \times \text{record size}) \times 0.5 \\
\times \text{block size}
\]

Record Size = 76 + length of member symbolic key + length of owner symbolic key + 4 bytes for the record descriptor word (RDW)
If your database contains a large number of member records in integrated index sets, multiply the result by 3. In a typical situation, a DASD cylinder allocation of 20,4 should be adequate.

Both INDXEXTR and INDXWORK should use the same space allocation.

**SORT File**

For standard auditing, space allocation for the SORT file depends on the number of error extract records created in the audit phase. If your database has numerous errors or the sets containing errors are long, the extract file may also be large. In that case, you may need more than four sort work datasets.

If you are auditing integrated index sets and you have index orphans, then you need enough sort space to accommodate twice the number of index orphans written to the index extract file.

In most cases, the number of extracts created is small. In a typical situation with Standard auditing, four sort work files with a cylinder allocation of 5,2 should be adequate. The total sort work file allocation should be twice the space allocation of the EXTRACTS file.

For QuickCheck auditing, allocate space to sort 26-byte records for every record swept in the database. For each record swept, CA IDMS/DB Audit creates a 26-byte extract record and a 26-byte match record for each set in which the swept record participates. Additionally, two 26-byte records are created for each SR7 and SR8 for index sets in the swept area.

**Total Number of Bytes to Sort = 26 * 2mn**

- **m**  
  Represents the number of records swept

- **n**  
  Represents the number of sets of which those records are participants

- **2mn**  
  Represents the number of records sorted

- **26**  
  Represents the length of sort record

**CA IDMS/DB Audit Sample JCL and Key**

**Contents**

- Allocating Work Files for the DISK Option (see page 146)

Sample JCL and key for running CA IDMS/DB Audit in a z/OS, z/VSE, or z/VM environment is listed below and supplied in source library member USAEXEC. The version of USAEXEC you receive is specific to your operating environment.

Sample IDCAMS control statements for allocating work files are contained in source library member USADEFN. This member is the same for all environments.
Allocating Work Files for the DISK Option

If you specify DISK in the AUDIT statement, you must allocate two direct access disk files: one for the db-key hash table and one for the db-key save table. These direct access files may be VSAM or, for z/OS and z/VM only, BDAM. If these files are allocated as VSAM, they must be defined through IDCAMS. This optional step is identical for all environments.

Sample control statements for VSAM file allocation are shown in Figure 5.5 and are contained in source library member USADEFN. A key to the variables (shown in bold) is shown in Figure 5.6.

DELETE (vsam.dbkhash) CLUSTER
DELETE (vsam.dbksave) CLUSTER
DEFINE CLUSTER (-
  NAME(vsam.dbkhash) -
  RECORDS(record.numbersh) -
  NUMBERED -
  RECORDSIZE(4084 4084) -
  SPEED -
  usetype -
  FILE(dname) -
  VOLUMES(volume.name) -
) -
DATA (-
  NAME(vsam.dbkhash.DATA) -
)
DEFINE CLUSTER (-
  NAME(vsam.dbksave) -
  RECORDS(record.numberss) -
  NUMBERED -
  RECORDSIZE(4064 4064) -
  SPEED -
  usetype -
  VOLUMES(volume.name) -
) -
DATA (-
  NAME(vsam.dbksave.DATA) -
)

Figure 5.5: Control Statements for Allocating VSAM Work Files for the DISK Option

vsam.dbkhash -- The dataset name of the VSAM file for the db-key table. This file is needed only if you specify DISK in the AUDIT statement.

record.number -- The number of records to be allocated in the VSAM work space for the db-key table. The number of records is equal to:

$$1.2 \ast \text{SETLIMIT-value} + 1$$

For example, if the SETLIMIT value is 3000 (default), you should allocate:

$$((1.2 \ast 3000/510) + 1 = 8$$

vsam.dbksave -- The dataset name of the VSAM file for the db-key save table. This file is needed if you specify DISK in the AUDIT statement.
record.numbers -- The number of records to be allocated in the VSAM work space for the db-key save table. The number of records depends on the number of next, prior, or owner pointer errors that you expect CA IDMS/DB Audit to find while set walking. Up to 127 pointer errors can be accommodated by one record. Therefore, you should allocate 1 primary record with extra secondary space, unless you expect more errors. An allocation of (1 10) should be adequate. The maximum number of records is equal to:

\[ SETLIMIT-value = 127 + 1 \]

usetype -- The appropriate type: REUSE or UNIQUE. If you specify UNIQUE, you must DELETE and DEFINE this cluster prior to each execution of CA IDMS/DB Audit.

volume.name -- Volume to contain the cluster or component.

dname -- The DD name of the JCL statement defining the file. Depending upon your environment, the parameter is optional.

*Figure 5.6: Key to Figure 5.5*

---

**z/OS Operations**

Contents

- z/OS JCL (see page 148)
  - 1. Allocating Work Files for the DISK Option (see page 148)
  - 2. Allocating Files (see page 148)
  - Key to Sample z/OS JCL for Allocating Files (see page 149)
  - 3. Running CA IDMS/DB Audit (see page 151)
  - Key to Sample z/OS JCL to Run CA IDMS/DB Audit (see page 152)
- Restarting CA IDMS/DB Audit (see page 153)
- Obtaining Reports for a Previous Run (see page 154)

Sample JCL for using CA IDMS/DB Audit in a z/OS environment is presented below in three parts:

1. (Optional) Allocating work files for the DISK option.

2. Allocating CA IDMS/DB Audit files.

3. Running CA IDMS/DB Audit.

An explanation of various restart options, showing the use of files created in a previous audit run, follows the JCL.
z/OS JCL

1. Allocating Work Files for the DISK Option

(USADEFN) If you specify DISK in the AUDIT statement and decide to use VSAM for the direct access files, you must allocate two VSAM files to disk: one for the db-key hash table and one for the db-key save table. See the Allocating Work Files for the DISK Option (see page 146) and Figure 5.5 above for sample IDCAMS control statements for file allocation.

2. Allocating Files

The sample z/OS JCL shown below in Figures 5.7a and 5.7b deletes any previous files and then allocates space for the general files and index files. Target or Distribution source library member USAEXEC contains similar sample JCL. The key to the variables (shown in bold) is shown in Figures 5.8a and 5.8b.

```jcl
//* OPTIONAL: SAMPLE JCL TO DELETE
//* PREVIOUSLY CREATED FILES
//*
//DELETE EXEC PGM=IEFBR14
//EXTRACTS DD DSN=dbaudit.extract,
// DISP=(MOD,DELETE),
// UNIT=disk,
// SPACE=(TRK,0)
//WORKFILE DD DSN=dbaudit.workfile,
// DISP=(MOD,DELETE),
// UNIT=disk,
// SPACE=(TRK,0)
//DBKYWORK DD DSN=dbaudit.dbkywork,
// DISP=(MOD,DELETE),
// UNIT=disk,
// SPACE=(TRK,0)
//INDEXEXTR DD DSN=dbaudit.indexextr
// DISP=(MOD,DELETE),
// UNIT=disk,
// SPACE=(TRK,0)
//INDEXWORK DD DSN=dbaudit.indexwork,
// DISP=(MOD,DELETE),
// UNIT=disk,
// SPACE=(TRK,0)
//* ALLOCATING GENERAL FILES
//*
//ALLOCATE EXEC PGM=IEFBR14
//EXTRACTS DD DSN=dbaudit.extract,
// DISP=(,CATLG,DELETE),
// VOL=SER=volser,
// UNIT=disk,
// SPACE=(CYL,(pe,se)),
// DCB=BLKSIZE=blkse
//WORKFILE DD DSN=dbaudit.workfile,
// DISP=(,CATLG,DELETE),
// VOL=SER=volser,
// UNIT=disk,
// SPACE=(CYL,(pw,sw)),
// DCB=BLKSIZE=blkse
```

Figure 5.7a: z/OS JCL for Allocating Files

```jcl
//* ALLOCATING INDEX FILES
//*
```
Figure 5.7b: z/OS JCL for Allocating Files

```plaintext
/*
  OPTIONAL: SAMPLE JCL TO DELETE & ALLOCATE THE 2 DIRECT ACCESS FILES AS BDAM
  */
DETERM EXEC PGM=IEFBR14
//DBKHASH DD DSN=dbaudit.dbkhash,
// DISP=(MOD,DELETE),
// UNIT=disk,
// SPACE=(TRK,0),
//DBKSAVE DD DSN=dbaudit.dbksave,
// DISP=(MOD,DELETE),
// UNIT=disk,
// SPACE=(TRK,0),
//ALLOCAT EXEC PGM=IEFBR14
//DBKHASH DD DSN=dbaudit.dbkhash,
// DISP=(CATLG,DELETE),
// VOL=SER=volser,
// UNIT=disk, SPACE=(4084,(blkcnt.hash)),
// DCB=DSORG=DA
//DBKSAVE DD DSN=dbaudit.dbksave,
// DISP=(CATLG,DELETE),
// VOL=SER=volser,
// UNIT=disk, SPACE=(4064,(blkcnt.save)),
// DCB=DSORG=DA
```

Figure 5.7c: z/OS JCL for Allocating Files

Key to Sample z/OS JCL for Allocating Files

- **dbaudit.extract** -- The dataset name assigned to your extract file.
- **disk** -- An appropriate unit designation for each file.
- **dbaudit.workfile** -- The dataset name assigned to your work file.
- **dbaudit.dbkywork** -- The dataset name assigned to your db-key work file. This file is needed only if you are auditing or fixing integrated index sets, and you have mandatory-automatic index sets.
- **dbaudit.indexextr** -- The dataset name assigned to your integrated index extract file. This file is needed only if you are auditing or fixing integrated index sets.
- **dbaudit.indexwork** -- The dataset name assigned to your integrated index work file. This file is needed only if you are auditing or fixing integrated index sets.
**dbaudit.dbkhash** -- The dataset name assigned to your BDAM db-key hash table. This file is only needed if you specify the DISK option of the AUDIT statement.

**dbaudit.dbksave** -- The dataset name assigned to your BDAM db-key save table. This file is only needed if you specify the DISK option of the AUDIT statement.

**volser** -- The volume serial number of the disk that will contain a particular file.

**pe,se** -- The primary/secondary space allocation for the EXTRACTS file allocated to DASD. Space allocation for a DASD file depends on the number of errors detected during the audit phase; the maximum record size is 512 bytes and must be blocked by the user (as noted in **blkse**). Because it is difficult to estimate the number of errors, you can allocate this file to tape. For most cases, DASD can be used. For example, if you are using a 3330 device with a blocksize of 6144, one cylinder can hold 456 extract records. A primary/secondary specification of 10,2 should be adequate.

**blkse** -- The block size of the EXTRACTS and WORKFILE files. The maximum record size is 512 bytes. A blocksize of 6144 is recommended for all device types.

**pw,sw** -- The primary/secondary space allocation for the WORKFILE file allocated to DASD. Space allocation should be twice the space allocation of the EXTRACTS file.

A primary/secondary allocation of 20,4 should be adequate.

**pd,sd** -- The primary/secondary space allocation for the DBKYWORK file allocated to DASD. To estimate the total number of blocks, multiply the total number of integrated index member records of mandatory-automatic sets by 30 (record length) times 2. The result is the total number of bytes. Divide by the block size to get the total number of blocks.

A primary/secondary allocation of 5,2 should be adequate.

**blksd** -- The block size of the DBKWORK file. The record size is 30 bytes. A block size of 6000 is recommended for all device types.

**pi,si** -- The primary/secondary space allocation for the INDXEXTR file and for the INDXWORK file, allocated to DASD. This space is primarily used for integrated index orphan records and secondarily for integrated index errors. If you have inserted a small number of member records in integrated index sets in the database, then you can estimate the total blocks by multiplying the total number of integrated index member records times the maximum record length (588) times 0.5, dividing by the block size. If a large number of member records have been inserted in integrated index sets, multiply the result by 3.

A primary/secondary allocation of 20,4 should be adequate.

**blksi** -- The block size of the INDXEXTR and INDXWORK files. The maximum record size is 588 bytes. A blocksize of 6144 is recommended for all device types.

**blkcnt.hash** -- The number of blocks for the BDAM db-key hash file. See Figure 5.6.

**blkcnt.save** -- The number of blocks for the BDAM db-key save file. See Figure 5.6.

*Figure 5.8: Key to Figure 5.7*
3. Running CA IDMS/DB Audit

Sample z/OS or JCL to audit and simulate fixing the database described by your subschema is shown in Figures 5.9a and 5.9b. A key to the variables (shown in bold) and file use is shown in Figures 5.10a and 5.10b. Similar JCL is contained in Target or Distribution source library member USAEXEC.

```plaintext
//* THIS JCL IS FOR AUDIT AND SIMULATED FIXING
/**
//DBAUDIT EXEC PGM=USADRVR,
  REGION=#K,
  PARM='NOSPIE'/
//STEPLIB DD DSN=your.loadlib, DISP=SHR
// DD DSN=idms.loadlib, DISP=SHR
//SORTLIB DD DSN=sort.loadlib, DISP=SHR
//SORTWK01 DD UNIT=disk,
  SPACE=(CYL,(ps,ss))
//SORTWK02 DD UNIT=disk,
  SPACE=(CYL,(ps,ss))
//SORTWK03 DD UNIT=disk,
  SPACE=(CYL,(ps,ss))
//SORTWK04 DD UNIT=disk,
  SPACE=(CYL,(ps,ss))
/dbfil01 DD DSN=user.dbfil01, DISP=SHR
/dbfilnn DD DSN=user.dbfilnn, DISP=SHR

Figure 5.9a: z/OS JCL to Run CA IDMS/DB Audit

/*GENERAL FILES NEEDED FOR ALL RUNS.
/**
//EXTRACTS DD DSN=daudit.extract, DISP=OLD
//WORKFILE DD DSN=daudit.workfile, DISP=OLD
/**
//sortmsg DD SYSOUT=a
//SYSLST DD SYSOUT=a AUDIT REPORT
//SYSSOUT DD SYSOUT=a INTEGRITY ANALYSIS REPORT
//SNAPIT DD SYSOUT=a
//SYSUDUMP DD SYSOUT=a
/**
/*FILES NEEDED FOR AUDITING OR FIXING
/*INTEGRATED INDEX SETS.
/**
//DBKYWORK DD DSN=daudit.dbkywork, DISP=OLD
//INDEXE XTR DD DSN=daudit.indexextr, DISP=OLD
//INDEXWORK DD DSN=daudit.indexwork, DISP=OLD
/**
/*FILES NEEDED ONLY IF DISK OPTION IS SPECIFIED
/*AND VSAM IS CHOSEN
/**
//DBKHASH DD DSN=vsam.dbkhash, DISP=SHR
//DBKSAVE DD DSN=vsam.dbksave, DISP=SHR
/**
```

Figure 5.9a: z/OS JCL to Run CA IDMS/DB Audit

/*GENERAL FILES NEEDED FOR ALL RUNS.
/**
//EXTRACTS DD DSN=daudit.extract, DISP=OLD
//WORKFILE DD DSN=daudit.workfile, DISP=OLD
/**
//sortmsg DD SYSOUT=a
//SYSLST DD SYSOUT=a AUDIT REPORT
//SYSSOUT DD SYSOUT=a INTEGRITY ANALYSIS REPORT
//SNAPIT DD SYSOUT=a
//SYSUDUMP DD SYSOUT=a
/**
/*FILES NEEDED FOR AUDITING OR FIXING
/*INTEGRATED INDEX SETS.
/**
//DBKYWORK DD DSN=daudit.dbkywork, DISP=OLD
//INDEXE XTR DD DSN=daudit.indexextr, DISP=OLD
//INDEXWORK DD DSN=daudit.indexwork, DISP=OLD
/**
/*FILES NEEDED ONLY IF DISK OPTION IS SPECIFIED
/*AND VSAM IS CHOSEN
/**
//DBKHASH DD DSN=vsam.dbkhash, DISP=SHR
//DBKSAVE DD DSN=vsam.dbksave, DISP=SHR
/**
```
// *THE INPUT STATEMENTS SHOWN HERE ARE EXAMPLES
// *FOR AUDIT AND SIMULATED FIX
// *
// SYSIPT DD *
// PROCESS, SUBSCHEMA=subschema-name,
// READYMODE=RETRIEVAL, SORTMSG=CRITICAL
// AUDIT, PAGES, USERSETS, CALCSETS, INDEX,
// CORE, SETLIMIT=3000, DBKEYTBL
// FIX=SIMULATE, PAGES, USERSETS, CALCSETS, INDEX, INDEXORPHANS
// REPORTS=YES, RECDATA=HEX, SETMBRS=ALL
// *
// SYSIDMS DD *
// SYSIDMS PARAMETERS...

Figure 5.9b: z/OS JCL to Run CA IDMS/DB Audit

/*
FILES NEEDED ONLY IF DISK OPTION IS SPECIFIED
AND BDAM IS CHOSEN
*/

/*
//DBKHASH DD DSN=dbaudit.dbkhash,
// DISP=SHR
// DCB=DSORG=DA
//DBKHA@@ DD DSN*.DBKHASH, VOL=REF=*.DBKHASH,
// DISP=SHR
// DCB=DSORG=DA
//DBKSAVE DD DSN=dbaudit.dbksave,
// DISP=SHR
// DCB=DSORG=DA
//DBKSA@@ DD DSN*.DBKSAVE, VOL=REF=*.DBKSAVE,
// DISP=SHR
// DCB=DSORG=DA

Figure 5.9c: z/OS JCL to Run CA IDMS/DB Audit

Key to Sample z/OS JCL to Run CA IDMS/DB Audit

#K -- The number of kilobytes of storage to be made available to the program execution phase. For more information, see the Storage Requirements (https://docops.ca.com/display/IDMS19/Storage+Requirements) section.

your.loadlib -- The dataset name of the MVS/PDS library into which you downloaded CA IDMS/DB Audit.

idms.loadlib -- The dataset name of the MVS/PDS library in which your DMCL and subschema reside.

sort.loadlib -- The dataset name of the MVS/PDS library in which your sort modules reside.

disk -- The unit address of the disk drive you want to use for the CA IDMS/DB Audit execution file, or a generic assignment to indicate a disk drive allocation.

ps,ss -- Primary/secondary space allocation for the SORT. Space allocation depends on the number of error extract records created in the audit phase. In most cases, the number of extracts created is small. A primary/secondary specification of 5,2 should be adequate. The total sortwork space allocated should be twice the space allocated to the EXTRACTS file.

If your database has numerous errors or the sets containing errors are long, the extract file may be large. In that case, you may need more than four SORTWK datasets.

If you are auditing integrated index sets and you have index orphans, then you need enough extra space to accommodate twice the number of index orphans written to the index extract file.
dbfil01/dbfilnn -- The DDNAMEs of the database files to be accessed.

user.dbfil01/user.dbfilnn -- The dataset names of the database files to be accessed.

dbaudit.extract -- The dataset name assigned to your extract file.

dbaudit.workfile -- The dataset name assigned to your work file.

sortmsg -- The DDNAME of the message file for your SORT product.

a -- The appropriate SYSOUT class for your installation.

dbaudit.dbkywork -- The dataset name assigned to your db-key work file. This file is needed only if you are auditing or fixing integrated index sets, and you have mandatory-automatic integrated index sets.

dbaudit.idxextr -- The dataset name assigned to your integrated index extract file. This file is needed only if you are auditing or fixing integrated index sets.

dbaudit.idxwork -- The dataset name assigned to your integrated index work file. This file is needed only if you are auditing or fixing integrated index sets.

vsam.dbkhash -- The dataset name of the VSAM file for the db-key hash table. This file is needed only if you specify DISK in the AUDIT statement. You must redefine this file using the IDCAMS control statements shown in Figure 5.5. Optionally, this file may be defined as BDAM (see Figures 5.7c, 5.8a, and 5.9c).

vsam.dbksave -- The dataset name of the VSAM file for the db-key save table. This file is needed only if you specify DISK in the AUDIT statement. You must redefine this file using the IDCAMS control statements shown in Figure 5.5. Optionally, this file may be defined as BDAM (see Figures 5.7c, 5.8a, and 5.9c).

subschema-name -- The name of the subschema that describes the database you want audited.

SYSIDMS -- This file is always required for the CA IDMS physical environment parameter to be read. For instance, DMCL=xxxxxxxx must be specified, where xxxxxxxx is the name of the specific DMCL.

dbaudit.dbkhash -- The dataset name assigned to your BDAM db-key hash table. This file is only needed if you specify the DISK option of the AUDIT statement.

dbaudit.dbksave -- The dataset name assigned to your BDAM db-key save table. This file is only needed if you specify the DISK option of the AUDIT statement.

*Figure 5.10: Key to Figure 5.9*

**Restarting CA IDMS/DB Audit**

When an audit and simulated fix have been successfully completed, you can restart CA IDMS/DB Audit and actually update the database. The JCL for the update run is the same as the previous JCL, except for the parameter statements. Be sure that no other updates are applied to the database between the audit run and the fix run.
Here are sample parameter statements:

PROCESS, SUBSCHEMA=subschema-name,
READYMODE=UPDATE, SORTMSGS=CRITICAL,
FIX=UPDATE, PAGES, USERSETS, CALCSETS, INDEXORPHANS,
REPORTS=YES, RECDATA=CHAR, SETMBRS=ERRSONLY

Obtaining Reports for a Previous Run

You may want only the reports for a previous run of CA IDMS/DB Audit. Again, you can use the same JCL and change only the parameter statements:

PROCESS, SUBSCHEMA=subschema-name,
READYMODE=RETRIEVAL, SORTMSGS=CRITICAL,
REPORTS=YES, RECDATA=HEX, SETMBRS=ALL

z/VSE Operations

Contents

- z/VSE File Assignments (see page 154)
- z/VSE JCL (see page 155)
  - 1. Allocating Work Files for the DISK Option (see page 155)
  - 2. Running CA IDMS/DB Audit (see page 155)
  - Key to Sample z/VSE JCL to Run CA IDMS/DB Audit (see page 156)
- Restarting CA IDMS/DB Audit (see page 157)
- Obtaining Reports for a Previous Run (see page 158)

Sample JCL for using CA IDMS/DB Audit in a z/VSE environment is presented below in two parts:

1. (Optional) Allocating work files for the DISK option

2. Running CA IDMS/DB Audit

An explanation of various restart options, showing the use of files created in a previous audit run, follows the JCL.

z/VSE File Assignments

Even if you use a storage management tool such as CA-DYNAM, CA IDMS/DB Audit requires an ASSGN statement for every file except SORTWKnn. This ASSGN is required because CA IDMS/DB Audit has its own device-independent support which dynamically builds a DTF based on the device type indicated by the ASSGN. Unless the ASSGN specifies VSAM or BDAM, the file may be defined with either DLBL or TLBL.
z/VSE JCL

1. Allocating Work Files for the DISK Option

(USADEFN) If you specify DISK in the Audit statement, you must allocate two VSAM files to disk: one for the db-key table and one for the db-key save table. See Allocating Work Files for the DISK Option (see page 146) and Figure 5.5 above for sample IDCAMS control statements for file allocation.

2. Running CA IDMS/DB Audit

Sample z/VSE JCL to audit and simulate fixing the database described by your subschema is shown in Figures 5.11a and 5.11b. A key to the variables (shown in bold) is shown in Figures 5.12a and 5.12b. Similar JCL is contained in TOOLJCL library member USAEXEC.

```jcl
// JOB DBAUDIT
// OPTION LOG,PARTDUMP
*
*   VSAM WORK FILE'S FOR AUDIT DISK OPTION
*
// DLBL   DBKHASH,'vsam.dbkhash',0,VSAM
// EXTENT SYS014,volser,,,rel-trk-blk,amount
// ASSGN SYS014,DISK,VOL=volser,SHR
// DLBL   DBKSAVE,'vsam.dbksave',0,VSAM
// EXTENT SYS015,volser,,,rel-trk-blk,amount
// ASSGN SYS015,DISK,VOL=volser,SHR
// EXEC IDCAMS,SIZE=AUTO
*
*   See source library member USADEFN for IDCAMS control statements
*
/*
*   SORT FILES
*//* DLBL SORTWK1,'SORT.WORK1',0,SD   SORT WORK #1
// EXTENT SYS001,volser,,,rel-trk-blk,amount
// ASSGN SYS001,DISK,VOL=volser,SHR
// DLBL SORTWK2,'SORT.WORK2',0,SD   SORT WORK #2
// EXTENT SYS002,volser,,,rel-trk-blk,amount
// ASSGN SYS002,DISK,VOL=volser,SHR
// DLBL SORTWK3,'SORT.WORK3',0,SD   SORT WORK #3
// EXTENT SYS003,volser,,,rel-trk-blk,amount
// ASSGN SYS003,DISK,VOL=volser,SHR
// DLBL SORTWK4,'SORT.WORK4',0,SD   SORT WORK #4
// EXTENT SYS004,volser,,,rel-trk-blk,amount
// ASSGN SYS004,DISK,VOL=volser,SHR
*
*   USER DATABASE FILES
*//* DLBL dbfil01,'user.dbfil01',DA  USER DATADBASE FILE #01
// EXTENT SYSnnn,volser
// ASSGN SYSnnn,DISK,VOL=volser,SHR
*.
*.
// DLBL dbfilnn,'user.dbfilnn',DA  USER DATADBASE FILE #nn
// EXTENT SYSnnn,volser
// ASSGN SYSnnn,DISK,VOL=volser,SHR
*
*   VSAM WORK FILE'S FOR AUDIT DISK OPTION
*
// DLBL   DBKHASH,'vsam.dbkhash',0,VSAM
// EXTENT SYS010,volser
// ASSGN SYS010,DISK,VOL=volser,SHR
```
Figure 5.11a: z/VSE JCL to Run CA IDMS/DB Audit

* DBAUDIT WORK FILES
* // ASSGN SYS020,DISK, VOL=volser, SHR
// EXTENT SYS020, volser, 1,0, rel-trk-blk, amount
// DLBL AFILE04,'AUDIT.extract',0,SD EXTRACT OUTPUT FILE
// DLBL AFILE05,'AUDIT.extract',0,SD EXTRACT INPUT FILE
// EXTENT SYS020, volser, 1,0, rel-trk-blk, amount
// DLBL AFILE06,'AUDIT.workfile',0,SD WORK OUTPUT FILE
// EXTENT SYS021, volser, 1,0, rel-trk-blk, amount
// DLBL AFILE07,'AUDIT.workfile',0,SD WORK INPUT FILE
// EXTENT SYS021, volser, 1,0, rel-trk-blk, amount
// ASSGN SYS021,DISK, VOL=volser, SHR
// DLBL AFILE08,'AUDIT.indexwork',0,SD INDEX WORK INPUT FILE
// EXTENT SYS022, volser, 1,0, rel-trk-blk, amount
// DLBL AFILE09,'AUDIT.indexwork',0,SD INDEX WORK OUTPUT FILE
// EXTENT SYS022, volser, 1,0, rel-trk-blk, amount
// ASSGN SYS023,DISK, VOL=volser, SHR
// DLBL AFILE10,'AUDIT.indexextr',0,SD INDEX EXTR INPUT FILE
// EXTENT SYS023, volser, 1,0, rel-trk-blk, amount
// DLBL AFILE11,'AUDIT.indexextr',0,SD INDEX EXTR OUTPUT FILE
// EXTENT SYS023, volser, 1,0, rel-trk-blk, amount
// ASSGN SYS024,DISK, VOL=volser, SHR
// DLBL AFILE12,'AUDIT.dbkywork',0,SD DBKEY INPUT FILE
// EXTENT SYS024, volser, 1,0, rel-trk-blk, amount
// DLBL AFILE13,'AUDIT.dbkywork',0,SD DBKEY OUTPUT FILE
// EXTENT SYS024, volser, 1,0, rel-trk-blk, amount
// ASSGN SYSIPT,SYSRDR INPUT PARAMETER FILE
// ASSGN SYS006,SYSLST AUDIT REPORT
// ASSGN SYS007,uuu INTEGRITY ANALYSIS REPORT
* * LIBRARY DEF'S
* // DLBL TOOL,'tool.loadlib'
// EXTENT ,volser
// DLBL IDMS,'idms.loadlib'
// EXTENT ,volser
// LIBDEF PHASE, SEARCH=(TOOL.sublib,IDMS.sublib)
* // DLBL SYSIDMS,'#SYSIPT',0,SD
// EXEC USADRVR, SIZE=(AUTO,#K), PARM='NOSPIE/
* * R14.1 OPTIONAL SYSIDMS PARAMETERS
* *
* // PROCESS, SUBSCHEMA=subschema-name,.....
* AUDIT,.........
* FIX=SIMULATE,.........
* &
* $ E0J

Figure 5.11b: z/VSE JCL to Run CA IDMS/DB Audit

Key to Sample z/VSE JCL to Run CA IDMS/DB Audit

vsam.dbkhash -- The data set name of the VSAM file for the db-key table. This file is needed only if you specify DISK in the AUDIT statement. You must pre-define this file by using the IDCAMS control statements.
volser -- The volume serial number or generic assignment of the disk volume on which the file, as specified in the preceding DLBL statement, resides.

rel-trk-blk -- Relative track or relative block number: the starting position on the DASD for storage of the work file specified in the previous statement.

amount -- The space allocation you need to store the file specified in the previous statement. Refer to the specific information on estimating space earlier in this section.

vsam.dbksave -- The data set name of the VSAM file for the db-key save table. This file is needed only if you specify DISK in the AUDIT statement. You must pre-define this file by using the IDCAMS control statements.

user.dbfil01/user.dbfilnn -- The file ID of each of the database files to be accessed.

SYSnnn -- The programmer logical units of the database files to be accessed.

extract -- The data set name assigned to your extract file.

workfile -- The data set name assigned to your work file.

indxwork -- The data set name assigned to your integrated index work file. This file is needed only if you are auditing or fixing integrated index sets.

indxextr -- The data set name assigned to your integrated index extract file. This file is needed only if you are auditing or fixing integrated index sets.

dbkywork -- The data set name assigned to your db-key work file. This file is needed only if you are auditing or fixing integrated index sets, and you have mandatory-automatic integrated index sets.

uuu -- SYS007 must be assigned to a second SYSLST device (uuu).

tool.loadlib -- The data set name of the core image library in which your DMCL and subschema reside.

tool.sublib/idms.sublib -- The sublibrary name of the z/VSE library specified in the previous file name.

subschema-name... -- The parameters you specify to execute CA IDMS/DB Audit.

Figure 5.12: Key to Figure 5.11

Restarting CA IDMS/DB Audit

When an audit and simulated fix have been successfully completed, you can restart CA IDMS/DB Audit and actually update the database. The JCL for the update run is the same as the previous JCL, except for the parameter statements. Be sure that no other updates are applied to the database between the audit run and the fix run.

Here are sample parameter statements:
Obtaining Reports for a Previous Run

You may want only the reports for a previous run of CA IDMS/DB Audit. Again, you can use the same JCL and change only the parameter statements:

```
PROCESS, SUBSCHEMA=subschema-name,
READYMODE=RETRIEVAL, SORTMSG=CRITICAL
REPORTS=YES, RECDATA=HEX, SETMBRS=ALL
```

z/VM Operations

Contents

- z/VM EXEC (see page 158)
  - 1. Allocating Work Files for the DISK Option (see page 158)
  - 2. Running CA IDMS/DB Audit (see page 159)
  - Key to Sample z/VM EXEC to Run CA IDMS/DB Audit (see page 160)
  - Restarting CA IDMS/DB Audit (see page 161)
  - Obtaining Reports for a Previous Run (see page 161)

Sample EXEC for using CA IDMS/DB Audit in a z/VM environment is shown below in two parts:

1. (Optional) Allocating work files for the DISK option

2. Running CA IDMS/DB Audit

An explanation of various restart options, showing the use of files created in a previous audit run, follows the EXEC.

z/VM EXEC

1. Allocating Work Files for the DISK Option

(USADEFN) If you specify DISK in the Audit statement, you must allocate two direct access disk files: one for the db-key hash table and one for the db-key save table. These direct access files may be VSAM or BDAM. If you choose to allocate these files as VSAM, they must be defined through AMSERV. See Allocating Work Files for the DISK Option (see page 146) and Figure 5.5 above for sample AMSERV control statements for file allocation.

If you choose to allocate these files as BDAM, you do not need to use AMSERV. Instead, revise your EXEC (as shown in Figure 5.13b) to contain the statements shown in Figure 5.13a.

```
* FILES NEEDED ONLY IF DISK OPTION IS SPECIFIED
* AND BDAM IS CHOSEN
*
FILEDEF  DBKHASH    fn ft fm6 (DSORG DA
```
FILEDEF DBKHA@@ fn ft fm6 (DSORG DA)
FILEDEF DBKSAVE fn ft fm6 (DSORG DA)
FILEDEF DBKSA@@ fn ft fm6 (DSORG DA)

* DBKHA@@ MUST HAVE THE SAME FILEID AS DBKHASH
* AND DBKSA@@ MUST HAVE THE SAME FILEID AS DBKSAVE

Figure 5.13a: z/VM EXEC to Run CA IDMS/DB Audit

2. Running CA IDMS/DB Audit

Sample z/VM EXEC to audit and simulate fixing the database described by your subschema is shown in Figure 5.13a and 5.13b. A key to the variables (shown in **bold**) is shown in Figures 5.14. A similar EXEC is contained in the source library member USAEXEC.

TOOL_LOADLIB_FN = 'toollib'
IDMS_LOADLIB_FN = 'idmslib'
SORTLIB_FN = 'sortlib'

/* Link and access the Minidisks containing the required library(s) /
/* and database file(s).
/*
/* 'CP SPOOL PRINTER NOCONT CLOSE'
/* 'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
/* 'GLOBAL LOADLIB ' TOOL_LOADLIB_FN IDMS_LOADLIB_FN
/* 'GLOBAL TXTLIB ' SORTLIB_FN
/*
/* Insert FILEDEFS as needed to define all required database files.
/*
/* 'FILEDEF dbfil01 DISK fn ft fm6 ( dcb'
/* 'FILEDEF dbfilnn DISK fn ft fm6 ( dcb'
/*
/* Files needed for all runs.
/*
/* 'FILEDEF SYSLST PRINTER'
/* 'FILEDEF SYSUDUMP PRINTER'
/* 'FILEDEF SYSOUT PRINTER'
/* 'FILEDEF SNAPIT PRINTER'
/* 'FILEDEF EXTRACTS DISK USAEXEC EXTRACTS fm'
/* 'FILEDEF WORKFILE DISK USAEXEC WORKFILE fm'
/*
/* You must create a file 'SYSIDMS PARMS A' containing the SYSIDMS /
/* parameters you use to specify your runtime environment.
/*
/* 'FILEDEF SYSIDMS DISK SYSIDMS PARMS A'
/*
/* You must create a file 'USAEXEC SYSIPT A' containing the input /
/* parameter statements prior to executing this EXEC.
/*
/* 'FILEDEF SYSIPT DISK USAEXEC SYSIPT A'
/*
/* Insert FILEDEF statements for SORT work space as required by /
/* your SORT product.
/*
/* Files needed for auditing or fixing integrated index sets.
/*
/* 'FILEDEF DBKYWORK Disk USAEXEC DBKYWORK fm'
/* 'FILEDEF INDXEXTR Disk USAEXEC INDXEXTR fm'
/* 'FILEDEF INDXWORK Disk USAEXEC INDXWORK fm'
/*
/* The remaining VSAM files are needed only if you specify DISK /
/* option on the "AUDIT" parameter statement and VSAM is chosen.
/*
/* Remove the comments from around the next 4 statements.
/* 'DLBL IJSYSCT vsam fm DSN vsamcat'
/* 'DLBL IJSYSUC vsam_fm DSN vsamcat'
Figure 5.13b: z/VM EXEC to Run CA IDMS/DB Audit

```plaintext
SAY 'START DATABASE AUDIT'
SIGNAL OFF ERROR
'EXECOS OSRUN USADRVR PARM='''NOSPIE'''
USADRVR RC = RC
IF USADRVR RC > 4 THEN DO
   CALL ERROR
END
'CP SPOOL PRINTER NOCONT'
'CP SPOOL PRINTER NAME DBAUDIT LISTING'
'CP SPOOL PRINTER OFF'
SAY 'USAEXEC FINISHED WITH A RETURN CODE OF' USADRVR_RC
'GLOBAL LOADLIB'
'GLOBAL TXTLIB'
'FILEDEF * CLEAR'
EXIT USADRVR_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 SPOOL PRINTER NAME DBAUDIT LISTING'
'CP SPOOL PRINTER OFF'
'GLOBAL LOADLIB'
'GLOBAL TXTLIB'
'FILEDEF * CLEAR'
EXIT ERROR_RC
*/
```

Figure 5.13b (cont.): z/VM EXEC to Run CA IDMS/DB Audit

**Key to Sample z/VM EXEC to Run CA IDMS/DB Audit**

- **toolib** -- The file name of the load library into which you downloaded CA IDMS/DB Audit.
- **idmslib** -- The file name of the load library containing your CA IDMS subschema and DMCL modules.
- **sortlib** -- The file name of the text library containing your sort modules.
- **dbfil01/dbfilnn** -- The name of each of the database files to be accessed.
- **fn ft fm** -- The file name, file type, and file mode of the relevant file. The file mode number of your database file(s) must be 6.
- **dcb** -- The data control block (DCB) information for the files.

**Note:** The following definitions are required if you specify theDISK option in the AUDIT statement. You must predefine these files using IDCAMS control statements. See Operations (https://docops.ca.com/display/IDMS19/Operations) of the CA IDMS/DB Audit Using section.
**dbksave** -- The data set name of the VSAM file for the db-key save table.

**dbkhash** -- The data set name of the VSAM file for the db-key hash table.

**vsamcat** -- The data set name of the VSAM catalog.

**vsam_fm** -- The file mode of the accessed minidisk containing your VSAM catalog(s) and file(s).

*Figure 5.14: Key to Figure 5.13a*

### Restarting CA IDMS/DB Audit

When an audit and simulated fix have been successfully completed, you can restart CA IDMS/DB Audit and actually update the database. The EXEC for the update run is the same as the previous EXEC, except for the parameter statements. Be sure that no other updates are applied to the database between the audit run and the fix run.

Here are sample parameter statements:

```
PROCESS, SUBSCHEMA=subschema-name,
READYMODE=UPDATE, SORTMSGS=Critical
FIX=UPDATE, PAGES, USRSETS, CLASSETS, INDEXORPHANS
REPORTS=YES, RECDATA=CHAR, SETMBRS=ERRONLY
```

### Obtaining Reports for a Previous Run

You may want only the reports for a previous run of CA IDMS/DB Audit. Again, you can use the EXEC and change only the parameter statements:

```
PROCESS, SUBSCHEMA=subschema-name,
READYMODE=RETRIEVAL, SORTMSGS=Critical
REPORTS=YES, RECDATA=HEX, SETMBRS=ALL
```