

IBM IMS Solution Packs

IMS Database Space Tuning Guide



Note:

Before using this information and the product it supports, read the information in [“Notices” on page 125.](#)

Fourth Edition (July 2022)

This edition applies to Version 2.2 of IBM IMS Database Solution Pack for z/OS (program number 5655-DSP), Version 2.1 of IBM IMS Database Utility Solution for z/OS® (program number 5698-DUL), Version 2.1 of IBM IMS Fast Path Solution Pack for z/OS (program number 5698-FPP), and to all subsequent versions and releases until otherwise indicated in new editions.

This edition replaces SC19-3395-02.

© **Copyright International Business Machines Corporation 2011, 2022.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

- About this information..... V**

- Part 1. Introduction to IMS database space tuning tasks..... 1**

 - Chapter 1. Overview of database space tuning.....3
 - What's new in the IMS Database Space Tuning Guide..... 3
 - Why is database tuning important?..... 4
 - Database tuning environment..... 5
 - Terminology..... 7
 - Product documentation and updates..... 9
 - Accessibility features..... 10

 - Chapter 2. Scenario: Policy-based database monitoring and tuning..... 11

- Part 2. Analyzing and tuning IMS full-function databases..... 15**

 - Chapter 3. Overview of full-function database analysis and tuning.....17
 - Exception categories for full-function databases.....18
 - Scenario for using the analysis and resolution topics..... 19
 - Understanding how IMS uses space in database data sets.....21
 - Information resources for analyzing exceptions..... 25

 - Chapter 4. Analyzing space use exceptions for all full-function database types..... 29

 - Chapter 5. Analyzing and resolving fragmentation and optimization exceptions for HDAM and PHDAM databases..... 37
 - Analyzing fragmentation exceptions for HDAM and PHDAM databases..... 37
 - Analyzing optimization exceptions for HDAM and PHDAM databases..... 41
 - Resolving space use exceptions for HDAM and PHDAM databases..... 45
 - Resolving fragmentation exceptions for HDAM and PHDAM databases..... 46
 - Resolving optimization exceptions for HDAM and PHDAM databases..... 47
 - Average number of I/O operations per database record..... 47
 - Tuning a randomizer..... 48
 - Tuning HDAM database that have long database records..... 59

 - Chapter 6. Analyzing and resolving fragmentation and optimization exceptions for HIDAM and PHIDAM databases.....63
 - Analyzing fragmentation exceptions for HIDAM and PHIDAM databases..... 63
 - Analyzing optimization exceptions for HIDAM and PHIDAM databases..... 67
 - Resolving space use exceptions for HIDAM and PHIDAM databases..... 69
 - Resolving fragmentation exceptions for HIDAM and PHIDAM databases..... 71
 - Resolving optimization exceptions for HIDAM and PHIDAM databases..... 71
 - Average number of I/O operations per database record..... 72
 - Is there enough free space defined in DBDGEN?..... 72
 - Is the block size or CI size appropriate?..... 73
 - Is the problem with long database records solved appropriately?..... 73
 - Resolving space use exceptions for PHIDAM primary indexes.....74
 - Resolving fragmentation exceptions for PHIDAM primary indexes..... 75

Chapter 7. Analyzing and resolving fragmentation and optimization exceptions for HISAM and SHISAM databases.....	77
Analyzing fragmentation exceptions for HISAM and SHISAM databases.....	77
Analyzing optimization exceptions for HISAM and SHISAM databases.....	80
Resolving space use exceptions for HISAM and SHISAM databases.....	82
Resolving fragmentation exceptions for HISAM and SHISAM databases.....	83
Resolving optimization exceptions for HISAM and SHISAM databases.....	83
Average number of I/O operations per database record.....	83
Is the length of the KSDS record appropriate?.....	84
Is the ESDS CI size appropriate for the record length of this database?.....	85
Does the segment fit on KSDS?.....	86
Chapter 8. Analyzing and resolving fragmentation and optimization exceptions for index databases..	87
Analyzing fragmentation exceptions for index databases.....	87
Analyzing optimization exceptions for index databases.....	88
Resolving space use exceptions for index databases.....	89
Resolving fragmentation exceptions for index databases.....	90
Resolving optimization exceptions for index databases.....	91
Part 3. Analyzing and tuning IMS data entry databases.....	93
Chapter 9. Overview of DEDB analysis and tuning.....	95
Chapter 10. Understanding the status of DEDB areas.....	97
Analyzing free space usage.....	100
Analyzing database record profile.....	105
Analyzing the randomizing routine.....	109
Analyzing DEDB areas at the UOW level.....	112
Chapter 11. Determining actions to resolve exceptions.....	115
Planning free space reorganization and usage monitoring.....	116
Chapter 12. Tuning DEDB areas.....	119
Optional: Customizing the randomizer for exceptionally large database records.....	119
Tuning randomizer and DBD parameters for DEDB.....	120
Notices.....	125
Index.....	129

About this information

IBM® IMS Solution Packs: IMS Database Space Tuning Guide (also referred to as IMS Database Space Tuning Guide) provides information that can help you monitor and tune IMS full-function databases and IMS data entry databases (DEDBs).

The monitoring and tuning procedures that are provided in these topics use the components of IBM IMS Database Solution Pack for z/OS, IBM IMS Database Utility Solution for z/OS, and IBM IMS Fast Path Solution Pack for z/OS.

These topics provide instructions for monitoring, analyzing, and tuning IMS databases. These topics are designed to help database administrators perform these tasks:

- Perform policy-based monitoring of IMS databases
- Understand the statuses of IMS databases
- Resolve database exceptions



Attention: These topics contain specific settings and target values for tuning databases. See [“IMPORTANT NOTICE”](#) on page 3 before you use these topics.

To use these topics, you should have a working knowledge of:

- The z/OS operating system
- ISPF
- SMP/E

Always check the IMS Tools Product Documentation page for complete product documentation resources:

<https://www.ibm.com/support/pages/node/712955>

The IMS Tools Product Documentation page includes:

- Links to [IBM Documentation](#) for the user guides ("HTML")
- Links to the PDF versions of the user guides ("PDF")
- Program Directories for IMS Tools products
- Recent updates to the user guide, known as "Tech docs" ("See updates to this information!")
- Technical notes from IBM Software Support, known as "Tech notes"
- White papers that describe product business scenarios and solutions

Part 1. Introduction to IMS database space tuning tasks

IMS Solution Packs: IMS Database Space Tuning Guide provides guidance for monitoring, analyzing, and tuning IMS databases.

The following topics explain policy-based database monitoring and tuning, the system environment, and database administrator tasks for monitoring, analyzing, and tuning IMS databases.

Topics:

- [Chapter 1, “Overview of database space tuning,” on page 3](#)
- [Chapter 2, “Scenario: Policy-based database monitoring and tuning,” on page 11](#)

Chapter 1. Overview of database space tuning

IBM IMS Database Solution Pack for z/OS (also referred to as IMS Database Solution Pack), IBM IMS Database Utility Solution for z/OS (also referred to as IMS Database Utility Solution), and IBM IMS Fast Path Solution Pack for z/OS (also referred to as IMS Fast Path Solution Pack) provide various tools to help you monitor and tune IMS databases.

IMS Solution Packs: IMS Database Space Tuning Guide provides guidance for using the tools that are provided by IMS Database Solution Pack, IMS Database Utility Solution, and IMS Fast Path Solution Pack to achieve optimal performance and to efficiently use space in your IMS databases.

IMPORTANT NOTICE

The settings and target values for some key indicators that are suggested throughout this IMS Database Space Tuning Guide are based on experiments on simulated databases and applications in a controlled laboratory environment. The experiments were run in non-production, test environments.

Any suggested target values are provided as guidance for database administrators who do not have practical experience with tuning of their databases. Database performance may be affected by numerous factors, including, but not limited to, the specific applications run against the database, how the database is maintained, and factors beyond those described in this guide that may exist. After gaining experience with the tuning of the real-life databases of their installation, database administrators should review and adapt the proposed target values to their specific databases and operational environments. Therefore, the provided target values must be regarded as general rules of thumb that provide a reasonable starting point when concrete tuning experience with the real-life databases of the installation is lacking.

The following topics provide overview of database space tuning tasks.

Topics:

- [“What's new in the IMS Database Space Tuning Guide” on page 3](#)
- [“Why is database tuning important?” on page 4](#)
- [“Database tuning environment” on page 5](#)
- [“Terminology” on page 7](#)
- [“Product documentation and updates” on page 9](#)
- [“Accessibility features” on page 10](#)

What's new in the IMS Database Space Tuning Guide

This topic summarizes the technical changes for this edition.

New and changed information is indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

Revision markers follow these general conventions:

- Only technical changes are marked; style and grammatical changes are not marked.
- If part of an element, such as a paragraph, syntax diagram, list item, task step, or figure is changed, the entire element is marked with revision markers, even though only part of the element might have changed.
- If a topic is changed by more than 50%, the entire topic is marked with revision markers (so it might seem to be a new topic, even though it is not).

Revision markers do not necessarily indicate all the changes made to the information because deleted text and graphics cannot be marked with revision markers.

SC19-3395-03 (July 2022)

Description	Related APARs
Documentation updates to support IMS Administration Foundation, which activates the IMS administration web-browser interface of IBM Unified Management Server for z/OS to enable the management IMS systems and resources.	N/A

SC19-3395-02 (September 2016)

Description	Related APARs
Updated several topics to support IBM IMS Database Utility Solution for z/OS.	N/A

SC19-3395-01 (October 2014)

Description	Related APARs
The following topics were updated to support IBM Management Console for IMS and Db2® for z/OS (also referred to as Management Console). <ul style="list-style-type: none">• “Database tuning environment” on page 5• Multiple topics in Part 2, “Analyzing and tuning IMS full-function databases,” on page 15• Chapter 10, “Understanding the status of DEDB areas,” on page 97	N/A
The Tune function of Fast Path Advanced Tool, which is included in IMS Fast Path Solution Pack, can be used for tuning DEDB areas. For more information, see “Tuning randomizer and DBD parameters for DEDB” on page 120.	N/A
Two new exception detection rules, which can be used for detecting free space fragmentation exceptions in DEDB areas, are added. For more information, see Chapter 10, “Understanding the status of DEDB areas,” on page 97.	N/A

Why is database tuning important?

Tuning IMS databases so that they provide the best possible performance for IMS applications is one of the most important responsibilities of IMS database administrators (DBAs).

In the daily use of IMS databases, segments are added, updated, and deleted. When free space is fragmented by a multitude of segment replacements and delete operations, and if segment overflow occurs, the performance of applications can suffer. When available space is used up, new segments cannot be added. When the randomizing logic does not match the distribution of root segment keys or database record lengths, unexpected overflow of segments can occur, and the performance of physical I/O of root segments or database records can suffer. Monitoring and tuning databases is critical to ensure that databases use space efficiently and to prevent performance degradation.

What happens if databases are not tuned regularly?

Free space fragmentation occurs frequently in IMS databases that are used on a regular basis. Having many pieces of fragmented free space that are too small to reuse leads to poor space utilization. Databases that are not as frequently used tend to have less fragmented free space. However, if data is periodically added, available space will eventually be used up. Also, database types that use randomizers, such as HDAM and Data Entry Databases (DEDBs), might have many synonyms that have high multiplicity or an excessive amount of overflow data due to changes in distribution of root keys and the database record profile that was initially projected.

Without appropriate database tuning, lack of space can lead to an unexpected service outage of the database. Also, other circumstances can increase the number of physical I/Os to retrieve data, and

cause the performance of applications to suffer. These circumstances eventually result in performance problems.

To prevent such problems, DBAs must proactively maintain their databases by monitoring space utilization on a daily basis, detecting exceptional statuses as early as possible, and establishing tuning plans before problems occur.

How are IMS solution packs beneficial in database monitoring and tuning?

IMS Database Solution Pack, IMS Database Utility Solution, and IMS Fast Path Solution Pack include the Database Sensor component. This component enables DBAs to collect database statistics within their daily operations. The collected database statistics can be kept in a central repository as *sensor data* and can be accumulated as historical database statistics.

Autonomics Director, an IBM IMS Tools Base for z/OS (also referred to as IMS Tools Base) component, schedules and automates policy evaluations of sensor data and notifies DBAs about the status of their databases. The Autonomics Director ISPF user interface simplifies the processes of scheduling and running policy evaluations and of checking monitoring results.

IMS Administration Foundation features of IBM Unified Management Server for z/OS (also referred to as IMS Administration Foundation) and IBM Management Console for IMS and Db2 for z/OS (also referred to as Management Console) consolidate key IMS statistics into an intuitive, graphical web interface. Through the web interface, users can view aggregate information for customizable groups of databases, graphical charts that show historical sensor data, database exceptions and corresponding recommendations, and the reports that are generated by IMS Tools.

The combination of IMS solution packs, IMS Tools Base components, and IMS Administration Foundation or Management Console greatly simplifies the tasks that are involved in day-to-day database monitoring and detecting database exceptions. By using these tools, IMS DBAs no longer need to develop and maintain their own database monitoring programs.

What is policy-based database monitoring and tuning?

As the number of databases within an organization increases, so does the amount of sensor data that is accumulated on a daily basis. This accumulation of data affects the productivity of DBAs, who must expend considerable time and effort analyzing this data.

Policy-based database monitoring and tuning is the ability to associate rules with three different threshold levels (warning, severe, and critical). When a threshold is met, a notification is issued to the DBA, who can then take appropriate measures. For example, analyzing the state of the database and performing free space reorganization. Policy-based monitoring and tuning reduces the amount of time that a DBA must manually monitor and evaluate database statistics.

Policy-based database monitoring and tuning is provided by integrating several components of IMS Tools Base with IMS Database Solution Pack, IMS Database Utility Solution, or with IMS Fast Path Solution Pack. A set of sensor data evaluation rules is included with these solution packs. DBAs can establish a policy-based monitoring and tuning environment simply by specifying thresholds for the three severity levels, establishing an analysis plan for each severity level, and specifying notification methods.

The three different severity levels help DBAs monitor changing trends within their database environment. Lower-level warnings provide the first indication that a database might need attention. Typically, warning signs do not indicate problems that require immediate attention; however, DBAs can act on a warning notification to mitigate a problem before it reaches the severe or critical level.

Database tuning environment

Policy-based database monitoring and tuning is achieved by using a set of IMS Tools products.

This topic summarizes the tools and components that are used in policy-based database monitoring and tuning.

To achieve policy-based database monitoring and tuning of IMS databases, you must install and configure IMS Tools Base. You must also install and configure one of the following IMS solution packs:

- IMS Database Solution Pack
- IMS Database Utility Solution
- IMS Fast Path Solution Pack

IMS Tools Base

You must install IMS Tools Base and then configure the following components of IMS Tools Base:

IMS Tools Knowledge Base

IMS Tools Knowledge Base (IMS Tools KB) functions as the server for the report repository, sensor data repository, and policy repository. IMS Tools KB provides an ISPF user interface to set up the IMS Tools KB environment and to manage repositories and reports.

Policy Services

Policy Services detects database exceptions based on user-defined policies. Policy Services provides an ISPF user interface to define and manage policies, rules, and notification lists.

Autonomics Director

Autonomics Director schedules sensor data collection and policy evaluations. A policy evaluation can detect exceptions and provide recommended user actions. Autonomics Director provides an ISPF user interface to schedule sensor data collection and policy evaluations and to view the results of policy evaluations.

IMS Administration Foundation and Management Console

IMS Administration Foundation features of IBM Unified Management Server for z/OS (also referred to as IMS Administration Foundation) and IBM Management Console for IMS and Db2 for z/OS (also referred to as Management Console) display key statistics and information about database resources in an intuitive graphical web interface.

IMS Database Solution Pack or IMS Database Utility Solution

To monitor IMS full-function databases, you must install IMS Database Solution Pack or IMS Database Utility Solution. Sensor data can be collected by using the utilities and services that are provided by these solution packs.

The following tools support sensor data collection:

- The HD Pointer Checker utility of IMS HP Pointer Checker
- IMS HP Image Copy
- FF Stand-alone DB Sensor
- The Smart Reorg utility of IMS Database Reorganization Expert

These tools also generate reports that assist you in database analysis and tuning.

For information about installing and configuring IMS Database Solution Pack or IMS Database Utility Solution, see the following information:

- *IMS Database Solution Pack: Overview and Customization*
- *IMS Database Utility Solution: Overview and Customization*

IMS Fast Path Solution Pack

To monitor IMS DEDB databases, you must install IMS Fast Path Solution Pack. Sensor data can be collected by using the utilities and services that are provided by this solution pack.

The following functions and tools support sensor data collection:

- Fast Path Advanced Tool functions

- Online Pointer Checker utility and Online Expert Reorganization utility of Fast Path Online Tools
- Online Space Management utility
- IMS HP Image Copy
- FP Stand-alone DB Sensor

These tools also generate reports that assist you in database analysis and tuning.

For information about installing and configuring IMS Fast Path Solution Pack, see *IMS Fast Path Solution Pack: Overview and Customization*.

Terminology

The IMS Database Space Tuning Guide topics use several unique terms that you should understand before you begin policy-based database tuning.

Unique terms used in this information

The following unique terms are used in this information:

Data element

A data element consists of a data element tag and a data element value pair based on the collected sensor data.

Exception or database exception

Refers to a database exception that is detected while a database is being evaluated. Exception conditions are based on threshold values that are defined in a policy in the REORG domain of Policy Services. This type of policy is called a reorganization policy.

Policy or reorganization policy

Refers to a set of criteria that is used in the policy evaluation process for evaluating database status. Reorganization policies belong to the REORG domain of Policy Services.

Rule

Refers to a criteria for evaluating the status of a database. Rules are contained in one or more reorganization policies.

Sensor data

Sensor data is the information collected from the databases for the purpose of measuring the state of a specific database condition.

These topics use the following IMS unique terms.

Table 1. IMS unique terms

Abbreviation	IMS system resource
ACB	IMS application control block
CI	Control interval of a VSAM data set
DBD	IMS database description
DBRC	IMS Database Recovery Control facility
DEDB	IMS data entry database
DOVF	Dependent overflow section of a DEDB area or a UOW
DMAC	DEDB area control list
DMB	IMS data management block
DMCB	DEDB master control block

Table 1. IMS unique terms (continued)

Abbreviation	IMS system resource
FFDB	Generic name for all IMS full-function database types. The following database types are included: HDAM, HIDAM, HISAM, SHISAM, INDEX, PHDAM, PHIDAM, and PSINDEX.
HALDB	IMS High Availability Large Database type. The following database types are included: PHDAM, PHIDAM, and PSINDEX.
IOVF	Independent overflow section of a DEDB area
MADS	DEDB multiple area data sets
RAA BASE or RAA	Root addressable part of a DEDB area or of a UOW
RAP	Root anchor point in the RAA BASE CI of a UOW
RECON	IMS Recovery Control
SDEP	Sequential dependent part of a DEDB area
UOW	Unit of work of a DEDB area

Product short names

Except where noted, the various versions of IMS are referred to simply as IMS.

These topics use the following abbreviations for product and component short names.

Table 2. Product and component short names

Short name	Product name
Autonomics Director	IBM IMS Tools Base for z/OS Autonomics Director
FF Stand-alone DB Sensor	Full-function Stand-alone Database Sensor component of IMS Database Solution Pack or IMS Database Utility Solution
FP Stand-alone DB Sensor	Fast Path Stand-alone Database Sensor component of IMS Fast Path Solution Pack
IMS Administration Foundation	IMS Administration Foundation features of IBM Unified Management Server for z/OS
IMS Database Reorganization Expert	IBM IMS Database Reorganization Expert for z/OS
IMS Database Solution Pack	IBM IMS Database Solution Pack for z/OS
IMS Database Utility Solution	IBM IMS Database Utility Solution for z/OS
IMS Fast Path Solution Pack	IBM IMS Fast Path Solution Pack for z/OS
IMS HP Fast Path Utilities or IMS HPFPU	IBM IMS Fast Path Solution Pack for z/OS: IMS High Performance Fast Path Utilities
IMS HP Image Copy	IBM IMS High Performance Image Copy for z/OS
IMS HP Load	IBM IMS High Performance Load for z/OS
IMS HP Pointer Checker	IBM IMS High Performance Pointer Checker for z/OS

Table 2. Product and component short names (continued)

Short name	Product name
IMS Tools Base	IBM IMS Tools Base for z/OS
IMS HP Unload	IBM IMS High Performance Unload for z/OS
IMS Tools Knowledge Base, IMS Tools KB, or ITKB	IBM IMS Tools Base for z/OS IMS Tools Knowledge Base
Management Console	IBM Management Console for IMS and Db2 for z/OS
Policy Services	IBM IMS Tools Base for z/OS Policy Services

Product documentation and updates

IMS Tools information is available at multiple places on the web. You can receive updates to IMS Tools information automatically by registering with the IBM My Notifications service.

Information on the web

Always refer to the IMS Tools Product Documentation web page for complete product documentation resources:

<https://www.ibm.com/support/pages/node/712955>

The IMS Tools Product Documentation web page includes:

- Links to [IBM Documentation](#) for the user guides ("HTML")
- PDF versions of the user guides ("PDF")
- Program Directories for IMS Tools products
- Recent updates to the user guides, referred to as "Tech docs" ("See updates to this information!")
- Technical notes from IBM Software Support, referred to as "Tech notes"
- White papers that describe product business scenarios and solutions

IBM Redbooks® publications that cover IMS Tools are available from the following web page:

<http://www.redbooks.ibm.com>

The IBM Information Management System website shows how IT organizations can maximize their investment in IMS databases while staying ahead of today's top data management challenges:

<https://www.ibm.com/software/data/ims/>

Receiving documentation updates automatically

To automatically receive automated emails that notify you when new technote documents are released, when existing product documentation is updated, and when new product documentation is available, you can register with the IBM My Notifications service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Notifications service:

1. Go to <http://www.ibm.com/support/mysupport>
2. Enter your IBM ID and password, or create one by clicking **register now**.
3. When the My Notifications page is displayed, click **Subscribe** to select those products that you want to receive information updates about. The IMS Tools option is located under **Software > Information Management**.
4. Click **Continue** to specify the types of updates that you want to receive.

5. Click **Submit** to save your profile.

How to send your comments

Your feedback is important in helping us provide the most accurate and highest quality information. If you have any comments about this or any other IMS Tools information, you can take one of the following actions:

- Click the Feedback button at the top of the IBM Documentation topic that you are commenting on.
- Click the Contact Us tab at the bottom of any IBM Documentation topic.
- Send an email to ibmdocs@us.ibm.com. Be sure to include the book title, topic or section title, specific text, and your comment.

To help us respond quickly and accurately, please include as much information as you can about the content you are commenting on, where we can find it, and what your suggestions for improvement might be.

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully.

The major accessibility features in the components that are discussed in these topics enable users to:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
 - *z/OS ISPF User's Guide, Volume 1*
 - *z/OS TSO/E Primer*
 - *z/OS TSO/E User's Guide*

These guides describe how to use ISPF, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.

Chapter 2. Scenario: Policy-based database monitoring and tuning

Policy-based database monitoring notifies you about the state of your databases. When an exceptional state is detected during a scheduled health check that is based on pre-defined user policies, a message is sent to a single user or multiple users based on the notification settings in Policy Services. Exception notification can be used as a trigger to start detailed database analysis and tuning.

This topic summarizes the tasks that are required to set up policy-based database monitoring and to receive notifications. It also provides instructions for analyzing and tuning databases, which you can follow when you receive database exception notifications.

Policy-based database monitoring and tuning is available for both full-function databases and DEDB databases.

The following figure and steps explain the typical procedures for setting up policy-based database monitoring and tuning:

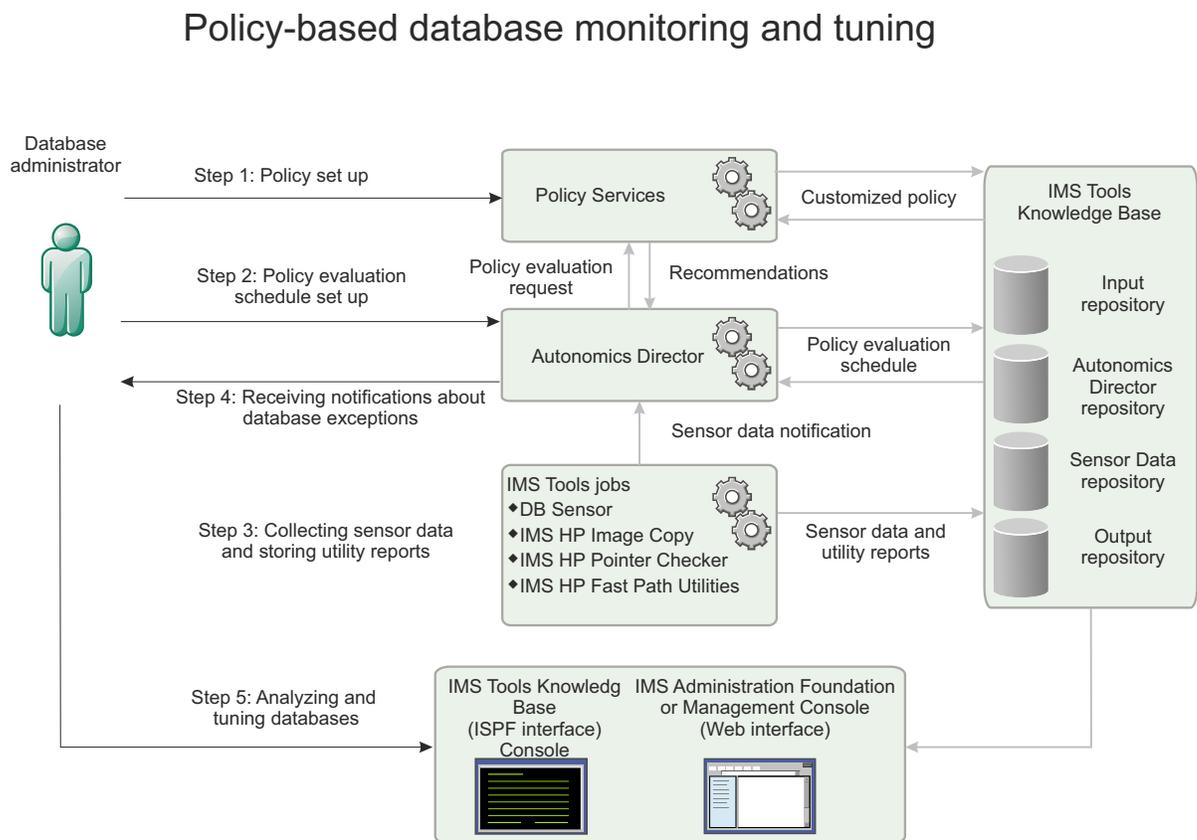


Figure 1. Policy-based database monitoring and tuning

Subsections:

- [“Step 1: Preparing reorganization policies” on page 12](#)
- [“Step 2: Setting up a policy evaluation schedule” on page 12](#)
- [“Step 3: Collecting sensor data and storing utility reports” on page 12](#)
- [“Step 4: Receiving notifications about database exceptions” on page 13](#)
- [“Step 5: Analyzing and tuning databases” on page 14](#)

Step 1: Preparing reorganization policies

Define the criteria for detecting exceptional database states in reorganization policies. The database that you want to monitor can be a non-partitioned full-function database or index, a HALDB partition, or a DEDB area. In the policies, you can also define the criteria for sending notifications about the detected exceptions.

Use the Policy Services ISPF user interface to prepare a set of policies for your environment by customizing policy templates that are provided with Policy Services. The customization of a policy template includes setting specific criteria for detecting exceptions and for determining when a database needs to be reorganized or restructured. You can also specify whether to generate exception notification messages from Autonomics Director and, if so, how and to whom those messages will be sent (for example, to a specific TSO ID, to a group of TSO IDs, to email addresses, or as WTO messages).

After you have created customized policies for your site, commit the changes. The changes and additions to the policies and the accompanying policy definition resources and environment information are stored in the Input repository of IMS Tools Knowledge Base (IMS Tools KB).

See the topic "Using Policy Services" in the *IMS Tools Base Policy Services User's Guide* for information about customizing IBM-provided policies and rules, and how to create your own policies. This information also provides a complete reference for IBM-provided policies and rules.

Step 2: Setting up a policy evaluation schedule

Autonomics Director schedules and automates the policy evaluation process. During the policy evaluation, the sensor data about the database is evaluated by the selected policy.

You can use the Autonomics Director ISPF user interface to set up the schedule.

For details, see the *IMS Tools Base Autonomics Director User's Guide*.

Step 3: Collecting sensor data and storing utility reports

Sensor data is collected and stored by several utilities of IMS Database Solution Pack, IMS Database Utility Solution, and IMS Fast Path Solution Pack. Various database analysis reports can also be generated by these utilities. These reports can be stored in the Output repository of IMS Tools KB. You can automate utility execution and sensor data collection by using Autonomics Director.

The following list summarizes the different types of data that can be collected and the utilities that can be used to collect that data:

Sensor data collection for full-function databases and indexes

You can use the following IMS Tools and components to collect sensor data from full-function databases and index databases:

- IMS HP Pointer Checker in IMS Database Solution Pack
- IMS HP Image Copy in IMS Database Solution Pack or IMS Database Utility Solution
- FF Stand-alone DB Sensor in IMS Database Solution Pack or IMS Database Utility Solution

The Smart Reorg utility of IMS Database Reorganization Expert can also collect sensor data for full-function databases except for index databases.

For details about collecting sensor data with these tools and components, see the following publications:

- *IMS High Performance Pointer Checker User's Guide*
- *IMS High Performance Image Copy User's Guide*
- *IMS Solution Packs: Data Sensor User's Guide*

Sensor data collection for DEDBs

The Analyze function of Fast Path Advanced Tool (FPA) and various online and offline utilities that can call the FPA Analyze function support sensor data collection. IMS HP Image Copy and FP Stand-alone DB Sensor, which are provided by IMS Fast Path Solution Pack, can also collect and store sensor data.

For details about the utilities that support sensor data collection, see the topic "Storing data in the Sensor Data repository of IMS Tools KB" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Reports that are generated by these utilities can be stored in the Output repository of IMS Tools KB for later analysis of database exceptions. Historical copies of utility reports are used to analyze database exceptions to determine appropriate tuning actions.

Each of these tools and utilities (except for the Smart Reorg utility) can notify Autonomics Director when sensor data is collected and stored in the Sensor Data repository. Autonomics Director uses this sensor data notification to schedule the policy evaluation for the subject database resources.

Tip: You can automate sensor data collection, in addition to policy evaluation, by using Autonomics Director. You can use the Autonomics Director ISPF user interface to set up the schedule.

For details about defining sensor data collection schedule by using Autonomics Director, see the *IMS Tools Base Autonomics Director User's Guide*.

Step 4: Receiving notifications about database exceptions

Autonomics Director can notify DBAs about exceptions that are detected during a policy evaluation.

Several notification methods are available. You can specify the notification method that you want to use in the policy. You can also create notification lists, which are lists of the users who will be notified when an exception occurs. Users who are included in the notification lists that are specified in the policy that was used will receive a message that summarizes the results of the policy evaluation. This message is called a *policy evaluation summary message* or simply a *summary message*.

You can use the summary message to identify the highest severity of all space management exceptions that were detected for a database resource at a specific point in time. The message also indicates where you can find relevant database space tuning information.

The following lists describe the summary messages for full-function database resources and DEDB areas:

Summary messages for full-function database resources

- BBE2902I *resource_name* IN RECONID=*recon_id* HAS WARNING EXCEPTIONS IN THE REORG DOMAIN.
- BBE2905I *resource_name* IN RECONID=*recon_id* HAS CRITICAL EXCEPTIONS IN THE REORG DOMAIN. USER ACTION IS REQUIRED.
- BBE2908I *resource_name* IN RECONID=*recon_id* HAS SEVERE EXCEPTIONS IN THE REORG DOMAIN.
- BBE2910I ACTION "REORG" IS RECOMMENDED FOR *resource_name* IN RECONID=*recon_id*.

where:

resource_name

Indicates the name of the resource that was processed. The resource can be a non-partitioned full-function database, a non-partitioned index, or a HALDB partition.

recon_id

Indicates the RECON ID of the RECON environment that the database belongs to. The RECON ID is defined through IMS Tools Knowledge Base.

For a more detailed explanation of these messages, see the *IMS Database Reorganization Expert User's Guide*.

Note: The Smart Reorg utility of IMS Database Reorganization Expert does exception detection and notification in each utility run by directly communicating with Policy Services. Therefore, Autonomics Director is not used by the Smart Reorg utility. The Smart Reorg utility attempts to reorganize the database to resolve free space fragmentation exceptions when such exceptions are detected by Policy Services. When the database is reorganized, the sensor data for the reorganized database is collected and evaluated with the same policy. Consequently, you might receive a summary message that is not listed in this topic.

For details about the Smart Reorg utility and other policy evaluation summary messages, see the topic "Reorganizing a database in one job step" in the *IMS Database Reorganization Expert User's Guide*.

Summary messages for DEDB areas

- HFP2902I *area_name* IN RECONID=*recon_id* HAS WARNING EXCEPTIONS IN THE REORG DOMAIN.
- HFP2905I *area_name* IN RECONID=*recon_id* HAS CRITICAL EXCEPTIONS IN THE REORG DOMAIN. USER ACTION IS REQUIRED.
- HFP2908I *area_name* IN RECONID=*recon_id* HAS SEVERE EXCEPTIONS IN THE REORG DOMAIN.
- HFP2910I ACTION "REORG" IS RECOMMENDED FOR *area_name* IN RECONID=*recon_id*.

where:

area_name

Indicates the name of the DEDB area.

recon_id

Indicates the RECON ID of the RECON environment that the DEDB area belongs to. The RECON ID is defined through IMS Tools Knowledge Base.

For a more detailed explanation of these messages, see the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Step 5: Analyzing and tuning databases

When exceptions are detected, Autonomics Director generates a recommendation list based on the exceptions that were detected during the policy evaluation, and stores this list in the repository of IMS Tools Knowledge Base. This recommendation list includes recommended actions for tuning the database, information about the detected exceptions, and a summary message. You can access the recommendation list by using the Autonomics Director ISPF user interface.

When you complete Steps 1 through 5, you will have a list of the database resources that need your attention and corresponding recommendations for your analysis.

The next step is to analyze and tune the databases. The following topics provide guidelines for analyzing the status of databases, determining appropriate actions, and tuning databases:

- To analyze and tune full-function databases, see [Part 2, "Analyzing and tuning IMS full-function databases,"](#) on page 15.
- To analyze and tune DEDBs, see [Part 3, "Analyzing and tuning IMS data entry databases,"](#) on page 93.

Tip: When you analyze the status of your databases, you can use the graphical web interface of IMS Administration Foundation or Management Console to view IMS Tools reports and the sensor data values that are relevant to the reported exceptions. IMS Administration Foundation and Management Console provide a consolidated view of all the information resources. For more information, see the *Unified Management Server User Guide* or the *Management Console User's Guide*.

Part 2. Analyzing and tuning IMS full-function databases

You can monitor, analyze, and tune IMS full-function databases by using the tools and components that are provided by IMS Database Solution Pack, IMS Database Utility Solution, IMS Tools Base, and IMS Administration Foundation or Management Console.

The following topics explain how to analyze and tune IMS full-function databases.

Topics:

- [Chapter 3, “Overview of full-function database analysis and tuning,” on page 17](#)
- [Chapter 4, “Analyzing space use exceptions for all full-function database types,” on page 29](#)
- [Chapter 5, “Analyzing and resolving fragmentation and optimization exceptions for HDAM and PHDAM databases,” on page 37](#)
- [Chapter 6, “Analyzing and resolving fragmentation and optimization exceptions for HIDAM and PHIDAM databases,” on page 63](#)
- [Chapter 7, “Analyzing and resolving fragmentation and optimization exceptions for HISAM and SHISAM databases,” on page 77](#)
- [Chapter 8, “Analyzing and resolving fragmentation and optimization exceptions for index databases,” on page 87](#)

Chapter 3. Overview of full-function database analysis and tuning

If you receive exceptions from Autonomics Director, you should analyze the state of the database and, if necessary, change the database configuration or tune the database.

The term *full-function database* includes the following database organizations: HDAM, HIDAM, HISAM, SHISAM, PHDAM, and PHIDAM. This term also pertains to the following indexes: HIDAM primary index, secondary index, PHIDAM primary index, and partitioned secondary indexes.

Each exception that is reported by Autonomics Director corresponds to one or more data elements. Therefore, when you receive exceptions from Autonomics Director, see the IMS Tools reports and charts in IMS Administration Foundation or Management Console by using its graphical user interface. These resources can help you understand the cause of the exceptions.

The topics in Part 2, “[Analyzing and tuning IMS full-function databases](#),” on page 15 describe how to resolve exceptions by pointing to the data elements that you should check.

Important characteristics to monitor

The typical key objectives of administering full-function databases are preventing out-of-space conditions in database data sets, preventing performance degradation that might be caused by excessive database I/Os, and, when such conditions occur, tuning the databases to recover from such conditions. To help achieve these objectives, exceptional states of database data sets are reported in the following three categories:

Space use

Monitor space use exceptions to prevent out-of-space conditions.

A full-function database consists of one or more data sets. The size of these data sets increase as the size of data increases. However, when the size of a data set reaches a limit, and if no space is available to store data, the condition might lead to a critical situation where IMS applications cannot run. This situation is referred to as an out-of-space condition.

To prevent out-of-space conditions, you must monitor how the space in the data sets is used, estimate when a data set runs out of space, and take appropriate actions to mitigate a problem before it becomes a significant problem.

Fragmentation

Monitor fragmentation exceptions to prevent and resolve redundant space and performance degradation that might be caused by excessive database I/Os. Fragmentation exceptions warn you about scattered segments, IMS free space fragmentation, VSAM control interval (CI) and control area (CA) splits, and other factors that might increase database I/Os.

Optimization

Monitor optimization exceptions also to prevent and resolve performance degradation that might be caused by excessive database I/Os. Database I/Os increase if the database definition does not match the characteristics of the records that are stored in the database.

For HDAM and PHDAM databases, the randomizing efficiency is also a key monitoring indicator.

By monitoring these three characteristics, you can understand and analyze the state of your databases. However, because exceptions tend to be interrelated, you might receive exceptions from multiple categories. When you are determining the actions to resolve exceptions, consider the exceptions not as separate exceptions, but rather as a group of interrelated exceptions.

Topics:

- “[Exception categories for full-function databases](#)” on page 18
- “[Scenario for using the analysis and resolution topics](#)” on page 19

- [“Understanding how IMS uses space in database data sets” on page 21](#)
- [“Information resources for analyzing exceptions” on page 25](#)

Exception categories for full-function databases

If you receive exceptions from Autonomics Director, you should identify the exception categories that the exceptions belong to.

The following table lists the major exceptions that might be identified as a result of a policy evaluation for full-function databases. The exceptions are organized by exception category. Use this table to identify the exception categories for the exceptions that were reported, and follow the instructions in the subsequent topics to analyze the state of the databases and to determine the appropriate actions to resolve the exceptions.

Tips:

- **IMS Administration Foundation:** The Exceptions tab of the DBD page shows a latest list of exceptions that were detected for the DBD. When you select a REORG domain exception, thumbnails of charts that are related to the exception are displayed. If you click one of those charts, all the related charts in the same exception categories are automatically selected and displayed in the Statistics tab. You can use various charts functions on the Statistics tab. For details, see the topic "Using IMS Administration Foundation" in the *Unified Management Server User Guide*.
- **Management Console:** The three exception categories correlate with the Space Use, Optimization, and Fragmentation sensor data widgets. In these widgets, you can locate the relevant charts for the following exceptions.
- The following table provides a summary for each exception. If you want to obtain a full explanation, see the content of the corresponding rule by using the Policy Services ISPF interface.

Table 3. Exception categories for full-function databases

Exception category	Exception and exception detection rule	Meaning of the exception
Space use	EXCESSIVE_SEGMENT_OCCURENCES Rule: IBM.SEGM_COUNT.10	The number of segments in a data set exceeded a threshold.
	DATA_SET_EXTENTS_AVAILABILITY Rule: IBM.DBDS_EXTENTS.10	The number of available extents for a data set fell below a threshold.
	DATA_SET_SIZE_GROWTH Rule: IBM.DBDS_GROWTH.10	The size of a data set is approaching its maximum size.
	GROWING_DBDS_WITH_DATA_FULL Rule: IBM.DBDS_GROWTH.30	The size of a data set is approaching its maximum size and the data set contains a relatively high amount of data.
	GROWING_DBDS_WITH_FREE_SPACES Rule: IBM.DBDS_GROWTH.20	The size of a data set is approaching its maximum size; however, the data set has a relatively high amount of free space.
	FREE_SPACE_AVAILABILITY Rule: IBM.FREE_SPACES.10	Total free space in one or more data sets is insufficient.
	INDEX_SIZE_GROWTH Rule: IBM.IX_GROWTH.10	The size of an index data set is approaching its maximum size.
	INDEX_EXTENTS_AVAILABILITY Rule: IBM.IX_EXTENTS.11	The number of available extents for the index data set fell below a threshold.
	GROWING_INDEX_WITH_DATA_FULL Rule: IBM.IX_GROWTH.20	The size of an index data set is approaching its maximum size and the data set contains a relatively high amount of data.

Table 3. Exception categories for full-function databases (continued)

Exception category	Exception and exception detection rule	Meaning of the exception
Fragmentation	EXCESSIVE_CI_OR_CA_SPLITS Rule: IBM.CICA_SPLITS.10	The percentage of VSAM CI or CA splits of the primary data set of a HISAM or SHISAM database exceeded a threshold.
	EXCESSIVE_HISAM_DELETE_SEGM Rule: IBM.HISAM_SEG_DEL.10	The number of deleted segment occurrences in the primary data set or the overflow data set of the HISAM database exceeded a threshold.
	FRAGMENTED_FREE_SPACES Rule: IBM.FRAGMENTAION.10	Excessive free space fragmentation was detected in a data set.
	EXCESSIVE_SEGMENT_SCATTERING (See Note “1” on page 19) Rule: IBM.SEGMENT_SPREAD.10	The number or percentage of physical pointers that point to different blocks or CIs in a data set exceeded a threshold.
	EXCESSIVE_SLACK_BYTES Rule: IBM.SLACK_BYTES.10	The number of slack bytes in a data set exceeded a threshold.
	EXCESSIVE_VL_SPLITS_SEGMENTS Rule: IBM.VL_SEGM_SPLIT.10	The number or percentage of variable-length segment splits in a data set exceeded a threshold.
	EXCESSIVE_INDEX_CI_OR_CA_SPLITS Rule: IBM.IX_CICA_SPLIT.10 and IBM.IX_CICA_SPLIT.11	The percentage of VSAM CI or CA splits in the index primary data set exceeded a threshold.
Optimization (See Note “2” on page 19)	AVERAGE_DB_RECORD_LENGTH Rule: IBM.AVG_DBREC_LEN.10	The average length of database records exceeded a threshold.
	IMBALANCED_RANDOMIZING Rule: IBM.RANDOMIZING.10	Even though an acceptable percentage of unused root anchor points (RAP) are available, the percentage of RAP synonyms exceeded a threshold.
	NUMBER_OF_DB_RECORDS Rule: IBM.NUM_DBRECORDS.10	The total number of database records exceeded a threshold.
	EXCESSIVE_HDAM_ROOTS_NOT_HOME Rule: IBM.ROOTS_NOHOME.10	The number or percentage of root segments that are not in the home block exceeded a threshold.
	EXCESSIVE_HDAM_ROOTS_OVERFLOW Rule: IBM.ROOTS_OVERFLOW.10	The number or percentage of root segment occurrences in the overflow area of an HDAM database or a PHDAM partition exceeded a threshold.
	EXCESSIVE_RAP_SYNONYMS Rule: IBM.RAP_SYNONYMS.10	The number or percentage of root segment occurrences involved in synonym chains of randomizing exceeded a threshold.
	NUMBER_OF_INDEX_POINTER_SEGMENTS Rule: IBM.IX_NUM_SEGM.10	The number of index pointer segments (IPs) exceeded a threshold.
	EXCESSIVE_INDEX_OVERFLOW Rule: IBM.IX_OVERFLOW.10	The number or percentage of IPs in the overflow data set exceeded a threshold.

Notes:

1. The EXCESSIVE_SEGMENT_SCATTERING exception can also be an optimization exception.
2. No optimization exceptions are detected for PHIDAM primary indexes.

Scenario for using the analysis and resolution topics

If you receive exceptions from Autonomics Director, you should analyze the state of the database and, if necessary, change the database configuration or tune the database.

The following figure shows an overview of the major steps to analyze and resolve exceptions. The organization of the topics in Part 2, “Analyzing and tuning IMS full-function databases,” on page 15 correspond to the steps of this scenario.

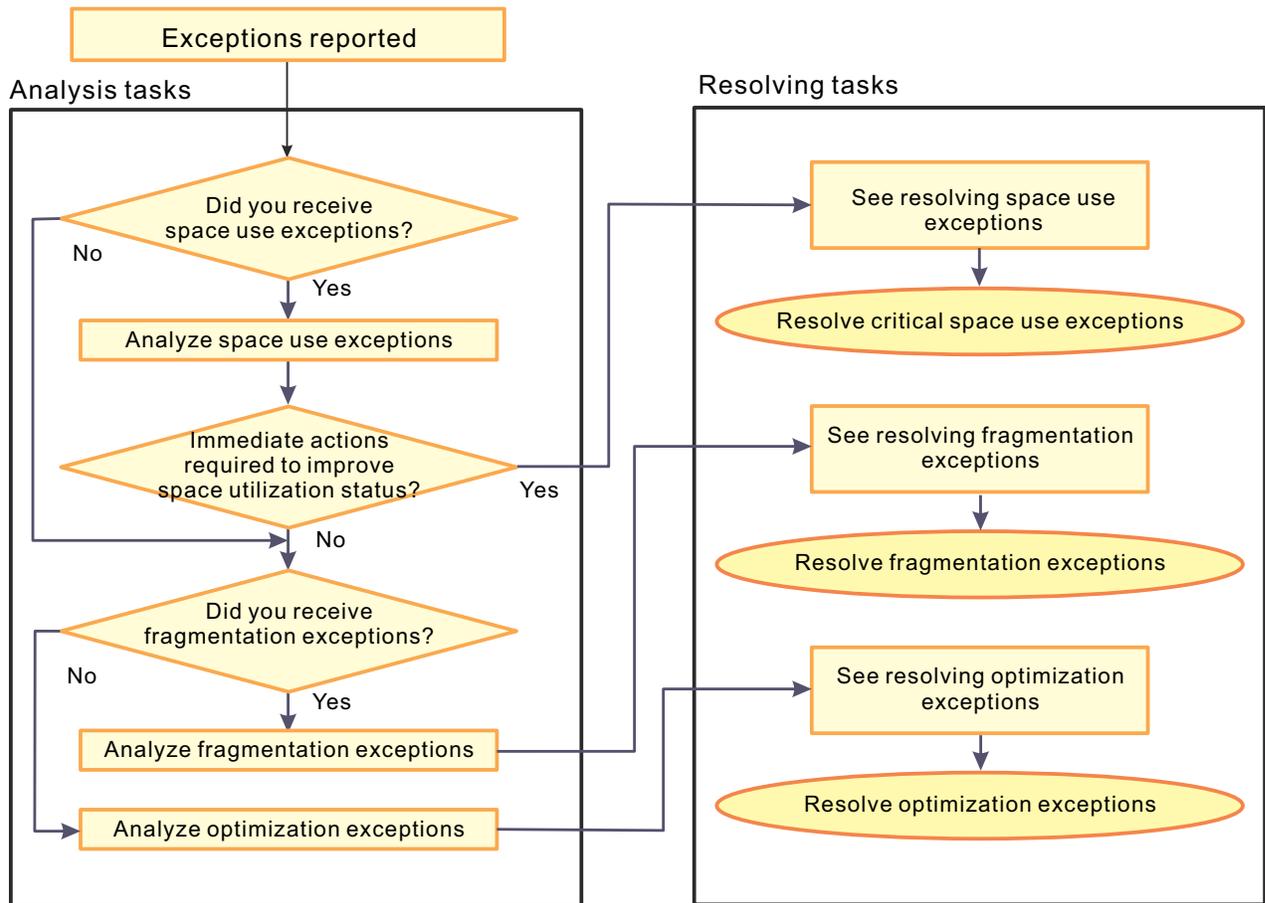


Figure 2. Overview of the major steps to analyze and resolve exceptions (full-function databases)

The following steps correspond to the flow that is presented in the previous figure:

1. If you received space use exceptions, see [Chapter 4, “Analyzing space use exceptions for all full-function database types,”](#) on page 29 and analyze the space use exceptions.

If you do not resolve space use exceptions that alert you about insufficient free space, database data sets can cause out-of-space conditions where new segments cannot be added. This situation might give negative impact to the IMS applications that use the database.

If you determine that immediate actions are required to improve the space utilization status, see the following topics and determine the actions to resolve the condition:

- [“Resolving space use exceptions for HDAM and PHDAM databases”](#) on page 45
- [“Resolving space use exceptions for HIDAM and PHIDAM databases”](#) on page 69
- [“Resolving space use exceptions for PHIDAM primary indexes”](#) on page 74
- [“Resolving space use exceptions for HISAM and SHISAM databases”](#) on page 82
- [“Resolving space use exceptions for index databases”](#) on page 89

2. If you determine that space utilization does not require immediate action, analyze fragmentation exceptions next, followed by optimization exceptions to understand the overall status of the database. A good understanding of the status helps you determine the best approach to resolve the exceptions. Then you should determine how you will resolve the exceptions based on your maintenance schedule.

- a. If you received fragmentation exceptions, analyze the fragmentation exceptions.

The fragmentation exceptions can cause redundant space and performance degradation that might be caused by excessive database I/Os.

See the following topics to analyze the fragmentation exceptions:

- [“Analyzing fragmentation exceptions for HDAM and PHDAM databases” on page 37](#)
- [“Analyzing fragmentation exceptions for HIDAM and PHIDAM databases” on page 63](#)
- [“Analyzing fragmentation exceptions for HISAM and SHISAM databases” on page 77](#)
- [“Analyzing fragmentation exceptions for index databases” on page 87](#)

After you have analyzed the fragmentation exceptions, see the following topics to determine the actions to resolve the condition:

- [“Resolving fragmentation exceptions for HDAM and PHDAM databases” on page 46](#)
- [“Resolving fragmentation exceptions for HIDAM and PHIDAM databases” on page 71](#)
- [“Resolving fragmentation exceptions for PHIDAM primary indexes” on page 75](#)
- [“Resolving fragmentation exceptions for HISAM and SHISAM databases” on page 83](#)
- [“Resolving fragmentation exceptions for index databases” on page 90](#)

- b. If you received optimization exceptions, analyze the optimization exceptions.

Optimization exceptions also can cause performance degradation. Performance degradation might be caused by excessive database I/Os when database record and root segment randomizing are not optimized.

See the following topics to analyze the optimization exceptions:

- [“Analyzing optimization exceptions for HDAM and PHDAM databases” on page 41](#)
- [“Analyzing optimization exceptions for HIDAM and PHIDAM databases” on page 67](#)
- [“Analyzing optimization exceptions for HISAM and SHISAM databases” on page 80](#)
- [“Analyzing optimization exceptions for index databases” on page 88](#)

Optimization exceptions typically cannot be resolved by a simple reorganization. Resolving optimization exceptions involves database tuning by changing the database definitions and other factors.

You can find guidelines for optimization tuning in the following topics:

- [“Resolving optimization exceptions for HDAM and PHDAM databases” on page 47](#)
- [“Resolving optimization exceptions for HIDAM and PHIDAM databases” on page 71](#)
- [“Resolving optimization exceptions for HISAM and SHISAM databases” on page 83](#)
- [“Resolving optimization exceptions for index databases” on page 91](#)

If you receive both fragmentation and optimization exceptions, consider working on the fragmentation exceptions first. Working on fragmentation exceptions before optimization exceptions will help you simplify your optimization tuning steps.

3. When you complete working on the exceptions, run DB Sensor to generate a Sensor Data Statistics report to confirm that the exceptions are resolved or that the status of the database has improved.

The scenario in this topic is an example of the flow for working on exceptions. Use this scenario as a starting point to develop your own approach that is best suited for your particular environment.

Understanding how IMS uses space in database data sets

Understanding how IMS uses space in full-function database data sets helps you manage your databases more efficiently.

A full-function database consists of one or more data sets. As IMS data increases in the data set, the free space in the data set decreases. When all the free space is used up, IMS applications cannot insert new segments or replace existing segments. To prevent this situation, you must take appropriate actions to mitigate the potential problems before they become significant issues. If you observe any space use exceptions, you must analyze the exceptions and establish action plans to resolve and prevent these exceptional states from occurring in the future.

This topic explains how the space in full-function database data sets is used by IMS. By understanding how IMS uses the space, you can address space use exceptions faster and maintain your data sets more efficiently.

Out-of-space problems can be caused by one of the following three conditions:

Primary space is used up

Primary allocation space is used up.

The data set is full

Data set size has reached the maximum limit that is defined by IMS.

Extent resources are used up

Data set cannot be expanded.

A primary-space-used-up condition occurs for the data sets that have only primary space allocated. Data-set-full and extent-resources-used-up conditions occur for the data sets that can allocate secondary space. When secondary space allocation is specified, data sets are expanded until a data-set-full or an extent-resources-used-up condition occurs.

The most appropriate method for resolving an out-of-space condition depends on the cause. Therefore, when you receive a space use exception, you must first identify the cause of the exception.

Subsections:

- [“How IMS uses space in data sets” on page 22](#)
- [“Primary-space-used-up condition” on page 23](#)
- [“Data-set-full condition” on page 23](#)
- [“Extent-resources-used-up condition” on page 23](#)

How IMS uses space in data sets

The following figure shows how IMS uses space in a full-function database data set.

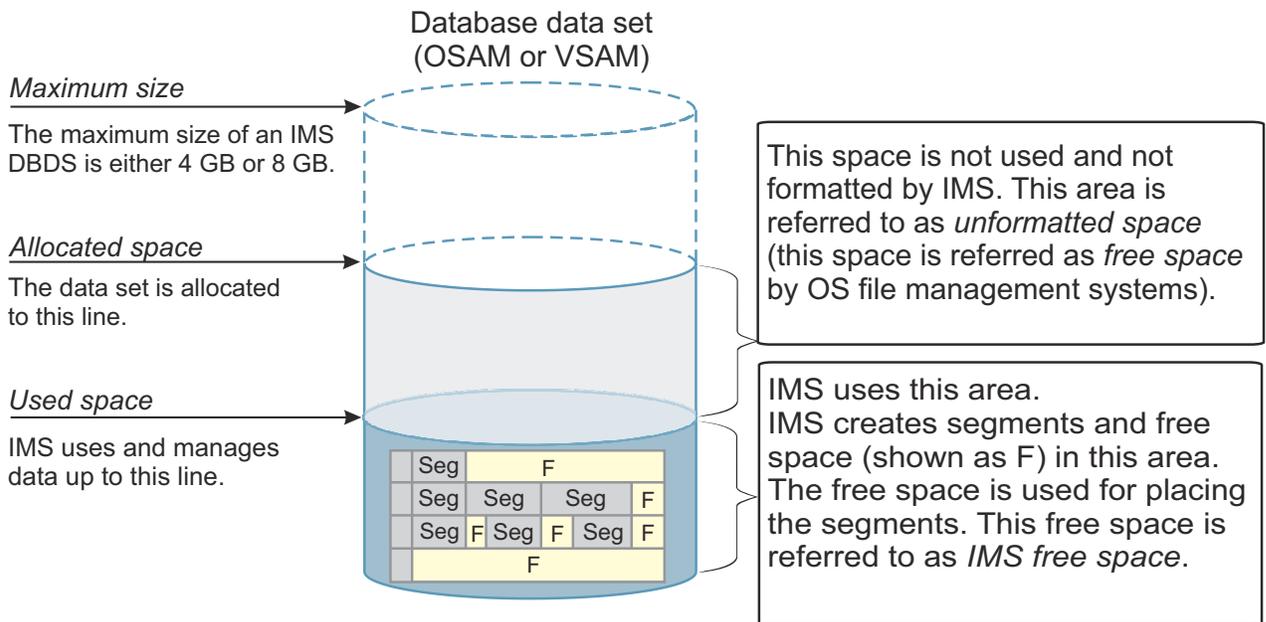


Figure 3. Space map of a full-function database data set

Locate *allocated space* and *used space* in the figure. The term *allocated space* refers to the space that is allocated by the file management system, and the term *used space* refers to the space that is used by IMS.

When IMS uses a data set for the first time, IMS formats the control intervals (CIs) or blocks that are inside the allocated space so that the data set can be used by IMS. The space that is formatted by IMS is the *used space*.

Immediately after the space is formatted, CIs or blocks contain no data, which means that CIs or blocks consist of only free space. When IMS applications run insert or replace operations, IMS stores the segment data in the free space. In these topics, this free space is referred to as *IMS free space*. However, the space in the allocated space (managed by the file management system) that is not used by IMS is generally known as free space. In these topics, to distinguish the terms, this free space is referred to as *unformatted space*.

Primary-space-used-up condition

When a database data set has only primary allocated space, and when the primary space is used up, IMS cannot add segments to the data set.

When the amount of IMS free space becomes too small to store new segment data in the remaining IMS free space, IMS formats additional CIs or blocks to expand the used space. When new CIs or blocks are formatted, IMS free space is increased and new segment data can be added. IMS repeats this, as necessary, until the size of the used space reaches the size of the allocated space. When the size of the used space reaches the primary allocated space and if the remaining IMS free space is not enough to store new segment data, IMS cannot insert or replace new segment data. This state is referred to as the *primary-space-used-up* condition.

Even if the primary allocated space is larger than the maximum size that is limited by IMS (4 GB or 8 GB), the used space cannot be larger than the size that is limited by IMS. When the amount of IMS free space is insufficient, IMS cannot insert or replace segment data.

The Policy Evaluation Services of Autonomics Director monitors these space resources, and when a primary-space-used-up condition is predicted, it sends a notification to the DBA.

Data-set-full condition

IMS limits the size of a database data set to either 4 GB or 8 GB. When the size of a data set reaches the maximum limit, that state is referred to as the *data-set-full* condition. To avoid data sets becoming full, DBAs must constantly identify the remaining free space in the data sets.

When the amount of IMS free space becomes too small to store new segment data in the remaining IMS free space, IMS formats additional CIs or blocks to expand the used space. When new CIs or blocks are formatted, IMS free space is increased, and new segment data can be added. IMS repeats this, as necessary, until the size of the used space reaches the size of the allocated space. Eventually, when the allocated space is used up, the file management system allocates an additional extent to make the allocated space larger. IMS continues to format additional CIs or blocks to expand the used space.

When the size of the used space reaches the maximum limit, and the remaining IMS free space is not enough to store new segment data, IMS cannot insert or replace new segment data, and a *data-set-full* condition occurs.

The Policy Evaluation Services of Autonomics Director monitors these space resources, and when a data-set-full condition is predicted, it sends a notification to the DBA.

Extent-resources-used-up condition

As mentioned in the explanation for the data-set-full condition, the file management system allocates a new extent to expand the data set. However, the file system cannot allocate a new extent if any one of the following conditions are met:

Insufficient available space on DASD volume

A data set can be allocated across multiple DASD volumes. The number of DASD volumes is specified when a data set is created. If the available space in the current volume is not enough, the file management system uses the next DASD volume to allocate the data set extent. However, when all

DASD volumes are used up, and if the last DASD volume does not have enough space for an extent, the file management system cannot expand the data set.

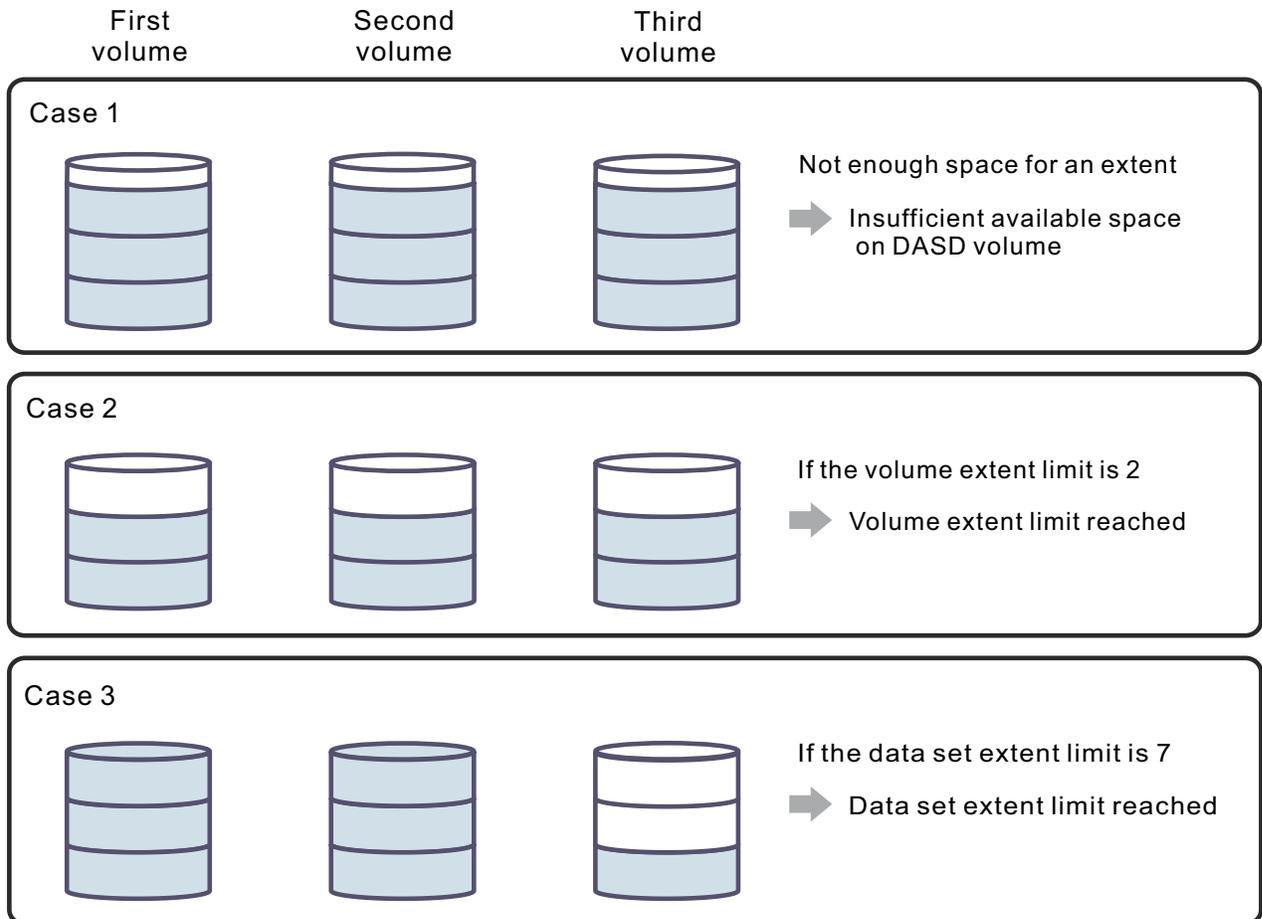
Volume extent limit reached

The number of data set extents on one DASD volume is limited by the file management system. When the data set uses up the last DASD volume, and if the volume extent limit is reached, the file management system cannot expand the data set.

Data set extent limit reached

The number of data set extents for a data set is limited by the file management system. When the data set reaches the data set extents limit, the file management system cannot expand the data set.

The following figure shows how DASD volumes are used and each condition that causes an extent-resources-used-up condition.



Note: This figure is simplified to illustrate the causes of extent-resources-used-up condition. The actual extent limits depend on your system environment and are typically larger than the values used in this figure.

Figure 4. How an extent-resources-used-up condition occurs

When the file management system cannot expand the data set, IMS cannot insert or replace a segment. To prevent this condition, you must monitor these space resources.

The Policy Evaluation Services of Autonomics Director monitors these space resources, and when an extent-resources-used-up condition is predicted, it sends a notification to the DBA.

By using Autonomics Director, you can detect potential exceptional states based on a predefined policy. When you receive exceptions from Autonomics Director, analyze the exceptions and establish an action plan before they become significant issues.

Information resources for analyzing exceptions

In the exception analysis procedures, you will be referring to a Sensor Data Statistics report, utility reports, and charts in IMS Administration Foundation or Management Console.

DB Sensor Statistics report

You can generate a Sensor Data Statistics report by running one of the following utilities or components:

- FF Stand-alone DB Sensor
- FF DB Sensor Printing utility
- FF Integrated DB Sensor that is called within the following jobs:
 - HD Pointer Checker utility of IMS HP Pointer Checker
 - IMS HP Image Copy

For information about generating a Sensor Data Statistics report by using FF Stand-alone DB Sensor or the FF DB Sensor Printing utility, see the *IMS Solution Packs: Data Sensor User's Guide*.

IMS HP Pointer Checker reports and IMS Database Reorganization Expert reports

IMS HP Pointer Checker and IMS Database Reorganization Expert reports provide more information to analyze the state of the database data sets.

The following table summarizes the IMS HP Pointer Checker reports that help you analyze the database data sets.

Table 4. IMS HP Pointer Checker reports that help you analyze database data sets

Report	Description	Topic in the <i>IMS High Performance Pointer Checker User's Guide</i>
DB Record Distribution Statistics report	By reviewing the summary of dependent segments distribution part and the distribution of dependent segments in root block part, you can determine how many dependent segments exist in the blocks that are different from the root segment block.	DB Record Distribution Statistics report
Free Space Map report	You can obtain information about the percentage of free space in each block or control interval (CI) in the database.	Free Space Map report
HD Data Set Statistics report	You can obtain various information about a database data set including the following information: <ul style="list-style-type: none">• Database and root segment definitions in DBD• Block or CI statistics in the database data set• Space use statistics, such as the total number of segment occurrences and free space in the data set• Statistics about segment occurrences and splits for each segment type	HD Data Set Statistics report

Table 4. IMS HP Pointer Checker reports that help you analyze database data sets (continued)

Report	Description	Topic in the <i>IMS High Performance Pointer Checker User's Guide</i>
HISAM Data Set Statistics report	<p>You can obtain various information about a database data set including the following information:</p> <ul style="list-style-type: none"> • Database and root segment definitions in DBD • Block or CI statistics in the database data set • Space use statistics, such as the total number of segment occurrences and free space in the data set • Statistics about segment occurrences and splits for each segment type 	HISAM Data Set Statistics report
Index Segment Counts Statistics report	<p>You can obtain information about index counts and indexed databases, such as index database and indexed database names, source and target segment names, number of index pointer and source segments, and the number of suppressed segments.</p> <p>This report is contained in the Evaluation of All Pointers to the Same Target report.</p>	Evaluation of All Pointers to the Same Target report
Interval Free Space Summary report	<p>You can obtain a summary of the free space for each processing interval that is defined by INTERVAL= keyword on the OPTION statement in histogram format. Each value in the report is the cumulative value of each interval that has been reported before, and will be accumulated to the next report.</p>	Interval Free Space Summary report
Maximum Free Space Distribution report	<p>You can obtain the following information for each block or CI in the database:</p> <ul style="list-style-type: none"> • The length of the longest free space element in the block or CI • The number of free space elements in the block or CI 	Maximum Free Space Distribution report
Partition Statistics report and Database Statistics report	<p>By reviewing the segment and pointer statistics part, the rate of segment I/O occurrence part, and the VL segment split statistics part, you can determine the number of pointers that point to different blocks for each segment.</p>	Partition Statistics report and Database Statistics report
Scan of Index Database report	<p>You can obtain the number of index pointer segments and the number of deleted index pointer segments in the primary data set and overflow data sets.</p>	Scan of Index Database report
Space Analysis by Data Set report	<p>You can obtain information for analyzing the space utilization of data sets for each data set. Entries are sorted in alphabetic order of the database name and data set ddname.</p>	Space Analysis by Data Set report
Space Monitor Graph report	<p>You can view historical trend of allocated space, used space, and IMS data (space used for IMS data).</p>	Space Monitor Graph report

Table 4. IMS HP Pointer Checker reports that help you analyze database data sets (continued)

Report	Description	Topic in the <i>IMS High Performance Pointer Checker User's Guide</i>
Summary of Data Sets by Volume report	You can obtain a list of data sets by volume. Entries are sorted in alphabetic order of the volume serial number, database name, and data set dname.	Summary of Data Sets by Volume report
Total DASD Utilization by Volume/Device-Type report	You can obtain information about DASD utilization by volume and device type.	Total DASD Utilization by Volume/Device-Type report

Notes:

- For information about generating these reports, see the *IMS High Performance Pointer Checker User's Guide*.
- If you store the reports in the IMS Tools KB Output repository, you can retrieve the reports by using the IMS Tools KB ISPF interface.

The following table summarizes the IMS Database Reorganization Expert reports that help you analyze the database data sets.

Table 5. IMS Database Reorganization Expert reports that help you analyze database data sets

Report	Description	Topic in the <i>IMS Database Reorganization Expert User's Guide</i>
Data Set Statistics report	You can check the DBD definition information and some statistics of each data set of the database. The DBD information includes the block size or CI size and the FRSPC parameter values. The statistics includes the largest and smallest segment lengths and the number blocks or CIs in the data set.	Data Set Statistics report
DB (Tuning) Statistics report	You can check the average number of I/Os that are required to randomly read all database segments of one database record. This report helps you determine whether a database is well organized.	Using Database Tuning Statistics Note: This topic can be found in the <i>IMS High Performance Unload User's Guide</i> .
Decoded DBD report	You can check the DBD source statements.	Decoded DBD report
Distribution of Database Record Sizes report	You can check the average length of database records and the distribution of the length of database records to determine the block size or the CI size of database data sets.	Distribution of Database Record Sizes report
HDAM RAP Chain I/O Summary report	You can use check the performance of the randomizer from the viewpoint of database I/O. The report also includes the packing density of root addressable area.	HDAM RAP Chain I/O Summary report
HDAM Root Addressable Area Statistics report	You can use this report to tune randomizing processes. The report includes the key indicators for measuring the quality of randomizing, the distribution of RAP chain lengths, and the distribution of root segment locations.	HDAM Root Addressable Area Statistics report
HISAM Statistics report	You can check the number of logical records in the data sets of a HISAM database and the chaining information of segments to the overflow data set.	HISAM Statistics report
Largest Database Records report	You can identify the longest database records in the database.	Largest Database Records report

Table 5. IMS Database Reorganization Expert reports that help you analyze database data sets (continued)

Report	Description	Topic in the IMS Database Reorganization Expert User's Guide
Segment Statistics report	You can check the average length per segment, the compression ratio of variable-length segment, and the status of splits.	Segment Statistics report

Notes:

- For information about generating these reports, see the *IMS Database Reorganization Expert User's Guide*.
 - If you store the reports in the IMS Tools KB Output repository, you can retrieve the reports by using the IMS Tools KB ISPF interface.
-

Chapter 4. Analyzing space use exceptions for all full-function database types

Analyze the space use exceptions that were detected for HDAM, PHDAM, HIDAM, PHIDAM, HISAM, SHISAM, and index databases.

Before you begin

Ensure that you can access the Sensor Data Statistics report that contains the latest sensor data. If you cannot access this report, run the FF DB Sensor Printing utility to generate the report.

About this task

In the following procedure, you will analyze each space use exception that you received.

For each exception that was reported, you will first identify the relevant data elements to understand the condition of the database. Then you will analyze the relevant IMS Tools reports to learn more about the database state. Finally, if you use IMS Administration Foundation or Management Console and if relevant charts are available, you will analyze historical trends of sensor data to understand the trend characteristics of the sensor data that was obtained from the database data sets.

By completing these steps, you will also identify the database data set for which the exceptions were reported and the condition that is causing the out-of-space problem, which can be one of the following conditions:

- Primary space is used up
- The data set is full
- Extent resources are used up

Procedure

In this section, instructions to analyze the common space use exceptions are provided. Follow the instructions for the exceptions that you received. The most common exceptions are listed first.

Exception class name: **EXCESSIVE_SEGMENT_OCCURENCES**

This exception indicates that the number of segments in one ore more data sets of the database exceeded a threshold value.

This exception does not always indicate a problem. When many segments exist and if the data set is about to meet an out-of-space condition, you must take appropriate actions to resolve this exception.

You can determine the number of segments in the database data set and the number of bytes that these segments occupy in the database data set by completing the following steps:

- a. Locate the following data elements in the Sensor Data Statistics report:

DB_NUM_SEG
DB_BYTES_SEG

The values of these data elements indicate the number of segments and the total bytes of segments.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

- b. Optional: To obtain more information about the segments, run the HD Pointer Checker utility of IMS HP Pointer Checker, or IMS HP Image Copy with the HDPC=YES option, and generate the following reports:

HISAM Data Set Statistics report

Use this report to determine the state of an HISAM or SHISAM database.

Database Statistics report

Use this report to determine the state of an HDAM or HIDAM database.

Partition Statistics report

Database Statistics report

Use either of these reports to determine the state of a PHDAM or PHIDAM database.

By reviewing these reports, you can determine how many segments exist for each segment type.

- c. Optional: To view the trend of data over time, run the Space Monitor utility of IMS HP Pointer Checker to generate the Space Monitor Graph report. You can run Space Monitor by running a stand-alone job, an HD Pointer Checker job with SPMN=YES, or an IMS HP Image Copy job with SPMN=YES.

By reviewing this report, you can identify the allocated space, used space, and the space that is occupied by IMS data.

- d. Optional: To view the trend of data over time, locate the following charts in IMS Administration Foundation or Management Console:

- Number of segments chart
- Total segment size chart

Understanding the trend over time can help you anticipate the future behavior of the data sets.

Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Steps [c](#) and [d](#), you also identified the report and the charts that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

If you want to obtain more information about the space utilization state, complete other analysis steps in this topic even if you did not receive other space use exceptions.

Exception class name: **FREE_SPACE_AVAILABILITY**

This exception indicates that the total bytes of segments have increased and that IMS free space has decreased in the used space.

This exception does not always indicate a problem. IMS free space decreases as segments are stored in the free space. If IMS free space is used up, IMS formats additional CIs or blocks to increase the IMS free space.

However, if the data set has only a primary space allocated, if the data set is using the last volume, and if the used space reaches the allocated space, the data set state might be close to a *primary-space-used-up* condition.

To determine the state of the data set, complete the following steps:

- a. Locate the following data elements in the Sensor Data Statistics report:

DB_BYTES_SEG

DB_PCT_BYTES_SEG

The values of these data elements indicate the total bytes of segments and the total bytes of segments compared to the size of used space.

DB_BYTES_FREE_SPACE

DB_PCT_BYTES_FREE_SPACE

The values of these data elements indicate the total bytes of IMS free space and the total bytes of IMS free space compared to the size of used space.

Locate the following data elements to identify whether the primary data set is used up:

DB_NUM_SEC_SPACE

The value of this data element indicates the secondary space allocation size of the data set. A value of zero indicates that the data set has only primary space.

DB_NUM_UNUSED_VOL

The value of this data element indicates the number of remaining volumes that can be allocated for the data set. A value of zero indicates that the data set is using the last DASD volume.

DB_RBA_HIGH_USED**DB_RBA_HIGH_ALLOC**

The values of these data elements indicate the size of used space and the size of the allocated space. Determine whether the values are the same.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

- b. Optional: To obtain more information about the segments and IMS free space, run the HD Pointer Checker utility of IMS HP Pointer Checker, or IMS HP Image Copy with the HDPC=YES option, and generate the following reports:

HISAM Data Set Statistics report

Use this report to determine the state of an HISAM or SHISAM database.

Database Statistics report

Use this report to determine the state of an HDAM or HIDAM database.

Partition Statistics report**Database Statistics report**

Use either of these reports to determine the state of a PHDAM or PHIDAM database.

Free Space Map report**Maximum Free Space Distribution report****Interval Free Space Summary report**

Use these reports to review the IMS free space in an HDAM, HIDAM, PHDAM, or PHIDAM database.

- c. Optional: To view the trend of data over time, run the Space Monitor utility of IMS HP Pointer Checker to generate the Space Monitor Graph report. You can run Space Monitor by running a stand-alone job, an HD Pointer Checker job with SPMN=YES, or an IMS HP Image Copy job with SPMN=YES.

By reviewing this report, you can identify the allocated space, used space, and the space that is occupied by IMS data.

- d. Optional: To view the trend of data over time, locate the following charts in IMS Administration Foundation or Management Console:

- Number of segments chart
- Total segment size chart
- IMS free space chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Steps [c](#) and [d](#), you also identified the report and the charts that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

Exception class name: DATA_SET_SIZE_GROWTH

Exception class name: GROWING_DBDS_WITH_DATA_FULL

Exception class name: GROWING_DBDS_WITH_FREE_SPACES

Exception class name: INDEX_SIZE_GROWTH

Exception class name: GROWING_INDEX_WITH_DATA_FULL

One or more of these exceptions indicate that the database data set size has increased and that the state is close to a *data-set-full* condition.

DATA_SET_SIZE_GROWTH

INDEX_SIZE_GROWTH

These exceptions indicate that the allocated size and the used size are growing.

GROWING_DBDS_WITH_DATA_FULL

GROWING_INDEX_WITH_DATA_FULL

These exceptions indicate that the allocated size is growing and both the unformatted free space and IMS free space have decreased. These exceptions indicate that the state is close to a data-set-full condition.

GROWING_DBDS_WITH_FREE_SPACES

This exception indicates that the allocated space has grown and the total size of IMS free space is large.

Even if the total size of IMS free space is large, IMS cannot add new segments when the IMS free space is fragmented and the fragmented space is not large enough to store the segments. If you receive this exception, also complete the analysis steps for fragmentation exceptions to analyze the state of IMS free space fragmentation.

If you do not address these exceptions, the database data set cannot be expanded any further, and new segments cannot be inserted.

Complete these steps to analyze these exceptions:

- a. Locate the following data elements in the Sensor Data Statistics report:

Note: For database data sets, locate the data elements that begin with DB_. For index database data sets and for PHIDAM primary indexes, locate the data elements that begin with DBX_.

DB_MAX_DS_SIZE

DBX_MAX_DS_SIZE

The values of these data elements indicate the maximum data set size that is specified by IMS, which is 4 GB or 8 GB.

DB_PCT_OF_MAX_DS_SIZE

DBX_PCT_OF_MAX_DS_SIZE

The values of these data elements indicate the allocated size compared to the maximum data set size.

DB_RBA_HIGH_ALLOC

DBX_RBA_HIGH_ALLOC

DB_RBA_HIGH_USED

DBX_RBA_HIGH_USED

The values of these data elements indicate the allocated space size in bytes and the used space size in bytes.

DB_UNUSED_BYTES

DBX_UNUSED_BYTES

DB_PCT_UNUSED_BYTES

DBX_PCT_UNUSED_BYTES

The values of these data elements indicate the unformatted space size in bytes and the unformatted space compared to the allocated size. If the DB_PCT_UNUSED_BYTES or DBX_PCT_UNUSED_BYTES value is small, the file management system will attempt to expand the data set soon.

DB_BYTES_SEG

DB_PCT_BYTES_SEG

The values of these data elements indicate the total bytes of segments and the total bytes of segments compared to the used space. If the DB_PCT_BYTES_SEG value is large, the file management system will attempt to expand the data set soon.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

The following figure illustrates how these data elements relate to space in IMS database data sets.

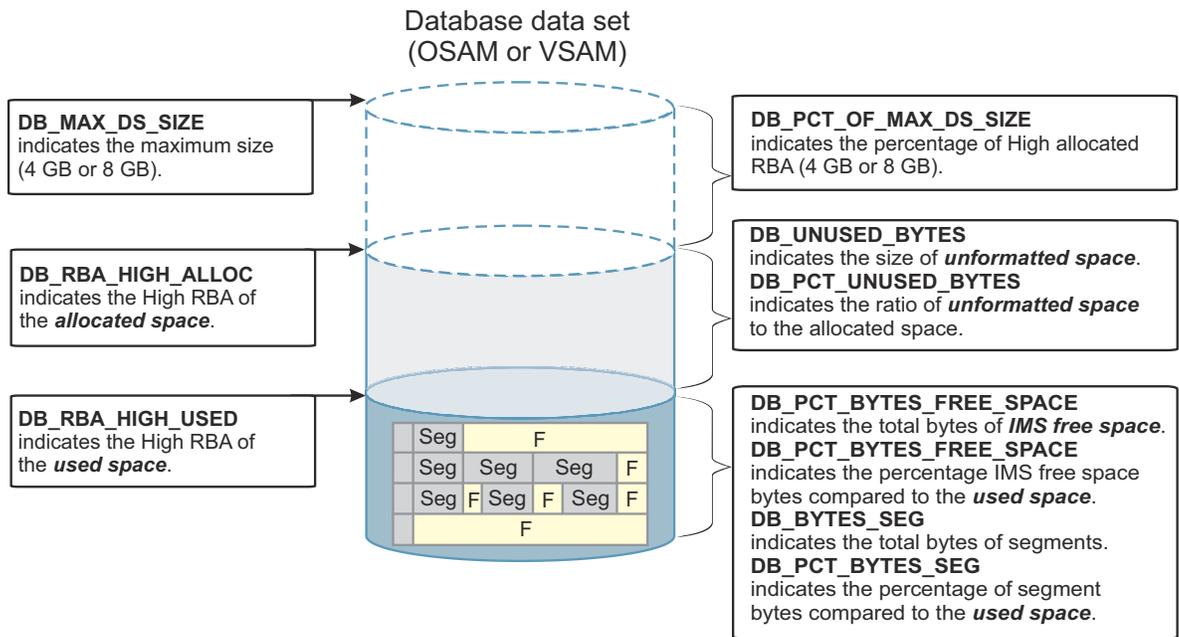


Figure 5. IMS database data set space and data elements

- b. Optional: To view the trend of data over time, run the Space Monitor utility of IMS HP Pointer Checker to generate the Space Monitor Graph report. You can run Space Monitor by running a stand-alone job, an HD Pointer Checker job with SPMN=YES, or an IMS HP Image Copy job with SPMN=YES.

By reviewing this report, you can identify the allocated space, used space, and the space that is occupied by IMS data.

- c. Optional: To view the trend of data over time, locate the following charts in IMS Administration Foundation or Management Console:

- Allocated space chart
- Unformatted space chart
- Highest used and highest allocated RBA chart
- IMS free space chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Steps [b](#) and [c](#), you also identified the report and the charts that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

Exception class name: DATA_SET_EXTENTS_AVAILABILITY

Exception class name: INDEX_EXTENTS_AVAILABILITY

This exception indicates that the available number of extents for the data set is inadequate. The data set state is close to an *extent-resources-used-up* condition.

If you do not address this exception, the database data set cannot be expanded any further, and new segments cannot be added.

Complete these steps to analyze these exceptions:

- a. Locate the following data elements in the Sensor Data Statistics report:

DB_NUM_EXT

The value of this data element indicates the number of extents of the data set.

DB_NUM_AVAIL_EXT

The value of this data element indicates the estimated number of extents that the data set will be able to acquire.

The following factors are evaluated, and if the minimum value of these factors is less than 100, that value is printed for this data element:

- Available space on DASD volumes
- The maximum extent count for the DASD volume
- The maximum extent count for the data set

If the value is zero, locate the DB_PCT_UNUSED_BYTES data element. If

DB_PCT_UNUSED_BYTES is small, the value indicates that the file management system will attempt to expand the data set soon. Schedule an action early, such as preparing more resources or reallocating the data set with more space.

Considerations for Extent Constraint Removal

When Extent Constraint Removal is active on your VSAM data set, the maximum extent count for the VSAM data set is removed and only the maximum extent count for the DASD volume is in effect for the data set. However, DB Sensor cannot determine whether Extent Constraint Removal is active. Consequently, DB Sensor assumes extent limits for all data sets, and sets the data element values.

If your VSAM data set has the extent constraint removal flag set to ON, and if the value for the DB_AVAIL_EXT_LIMIT data element is VSAM_MAXIMUM, the available extent count might be larger than the value that is indicated by the DB_NUM_AVAIL_EXT data element. In this case, run the Space Monitor utility of IMS HP Pointer Checker to generate a Space Monitor report, and analyze the report to determine the available number of DASD extents and the amount of free space in the DASD volume.

DB_AVAIL_EXT_LIMIT

The value is one of VSAM_MAXIMUM, OSAM_MAXIMUM, or VOL_FREE_EXTENTS.

VSAM_MAXIMUM**OSAM_MAXIMUM**

These values indicate that the number of available extents is close to the allowable number of VSAM or OSAM data set extents.

VOL_FREE_EXTENTS

This value indicates that available free space on the DASD volume is inadequate.

DB_MAX_EXT_DS**DB_MAX_EXT_VOL**

The values of these data elements indicate the maximum extent count for the data set, and the maximum extent count for a DASD volume.

DB_NUM_VOL

The value of this data element indicates the number of DASD volumes that the data set allocates.

DB_NUM_UNUSED_VOL

The value of this data element indicates the number of DASD volumes that the data set has not allocated yet. The value indicates that the data set can use the indicated number of DASD volumes.

DB_NUM_UNUSED_VOL_SER

The value of this data element indicates the number of unused DASD volumes that the volume serial number has been assigned.

DB_NUM_UNUSED_VOL_CAND

The value of this data element indicates the number of unused DASD volumes that the volume serial number has not been assigned.

When this value is more than zero, if you add DASD volumes to the SMS management resource group, the data set can allocate them.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

- b. Optional: To obtain more information about the DASD volumes and data set extents, run the Space Monitor utility of IMS HP Pointer Checker to generate the following reports. You can run the Space Monitor by running a stand-alone job, an HD Pointer Checker job with SPMN=YES, or an IMS HP Image Copy job with SPMN=YES.

Space Analysis by Data Set report

Use this report to analyze the space utilization of the data set.

Summary of Data Sets by Volume report

Use this report to identify the data sets in each volume.

Total DASD Utilization by Volume/Device-Type report

Use this report to understand DASD space utilization for each volume and for each device type.

- c. Optional: To view the trend of data over time, locate the following charts in IMS Administration Foundation or Management Console:

- Remaining extents chart
- DASD volume use chart
- Data sets extents chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Step c, you also identified the charts that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

What to do next

If you received other exceptions that belong to other exception categories, see the following topics to determine your next step:

- [“Exception categories for full-function databases” on page 18](#)
- [“Scenario for using the analysis and resolution topics” on page 19](#)

If the only exceptions that you received are space use exceptions, continue with the following topics to resolve the exceptions:

- [“Resolving space use exceptions for HDAM and PHDAM databases” on page 45](#)
- [“Resolving space use exceptions for HIDAM and PHIDAM databases” on page 69](#)
- [“Resolving space use exceptions for PHIDAM primary indexes” on page 74](#)
- [“Resolving space use exceptions for HISAM and SHISAM databases” on page 82](#)
- [“Resolving space use exceptions for index databases” on page 89](#)

Chapter 5. Analyzing and resolving fragmentation and optimization exceptions for HDAM and PHDAM databases

If you receive fragmentation and optimization exceptions for HDAM or PHDAM databases, analyze the state of the database and resolve the exceptions.

The following topics explain how to analyze the database states based on the exceptions that you received and how to recover from the exceptional states.

Topics:

- [“Analyzing fragmentation exceptions for HDAM and PHDAM databases” on page 37](#)
- [“Analyzing optimization exceptions for HDAM and PHDAM databases” on page 41](#)
- [“Resolving space use exceptions for HDAM and PHDAM databases” on page 45](#)
- [“Resolving fragmentation exceptions for HDAM and PHDAM databases” on page 46](#)
- [“Resolving optimization exceptions for HDAM and PHDAM databases” on page 47](#)

Analyzing fragmentation exceptions for HDAM and PHDAM databases

Analyze the fragmentation exceptions that were detected for HDAM and PHDAM databases.

Before you begin

Ensure that you can access the Sensor Data Statistics report that contains the latest sensor data. If you cannot access this report, run the FF DB Sensor Printing utility to generate the report.

About this task

In the following procedure, you will analyze each fragmentation exception that you received for HDAM databases or PHDAM partitions.

For each exception that was reported, you will first identify the relevant data elements to understand the condition of the database. Then you will analyze the relevant IMS Tools reports to learn more about the database state. Finally, if you use IMS Administration Foundation or Management Console and if relevant charts are available, you will analyze historical trends of sensor data to understand the trend characteristics of the sensor data that was obtained from the database data sets.

By completing these steps, you will identify the type of fragmentation symptoms that occurred and the database data set for which the exceptions were reported.

Procedure

In this section, instructions to analyze the common fragmentation exceptions are provided. Follow the instructions for the exceptions that you received. The most common exceptions are listed first.

Exception class name: **EXCESSIVE_SEGMENT_SCATTERING**

This exception indicates that too many physical pointers point to another block or CI.

When a parent segment and its dependent segment exist in the same block or CI, or when twin segments are in the same block or CI, extra I/O operations are not needed. However, if they are in different blocks or CIs, the number of I/O operations might increase.

Complete these steps to analyze this exception:

- a. Locate the following data elements in the Sensor Data Statistics report:

DB_PCT_NUM_PTR_DIFF_BLK

DB_NUM_PTR_DIFF_BLK

These data elements indicate the number of physical pointers that point to another block or CI, and the percentage of physical pointers that point to another block or CI compared to the physical pointers.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

- b. Optional: To obtain more information about the physical pointers, run the HD Pointer Checker utility of IMS HP Pointer Checker, or run IMS HP Image Copy with the HDPC=YES option, and generate the following reports:

DB Record Distribution Statistics report

By reviewing the summary of dependent segments distribution part and the distribution of dependent segments in root block part, you can determine how many dependent segments exist in the blocks that are different from the root segment block.

Partition Statistics report

Database Statistics report

By reviewing the segment and pointer statistics part, the rate of segment I/O occurrence part, and the VL segment split statistics part, you can determine the number of pointers that point to different blocks for each segment.

- c. Optional: To view the trend of data over time, locate the following chart in IMS Administration Foundation or Management Console:

- Pointers to other blocks or CIs chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Step c, you also identified the chart that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

Exception class name: EXCESSIVE_VL_SPLIT_SEGMENTS

This exception indicates that too many variable-length segment splits exist.

When a variable-length segment is replaced with a segment that has a longer length, and the amount of free space in the block or CI is inadequate, the segment is split into the prefix part and the data part. These segment splits can increase I/O operations.

Complete these steps to analyze this exception:

- a. Locate the following data elements in the Sensor Data Statistics report:

DB_NUM_VLSEG_SPLIT

DB_PCT_NUM_VLSEG_SPLIT

The values for these data elements indicate the number of segment splits and the percentage of segment splits compared to the number of variable segments.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

- b. Optional: To obtain more information about the variable-length segment splits, run the HD Pointer Checker utility of IMS HP Pointer Checker, or IMS HP Image Copy with the HDPC=YES option, and generate the HD Data Set Statistics report. In this report, review the Segment occurrences/split part.

- c. Optional: To view the trend of data over time, locate the following chart in IMS Administration Foundation or Management Console:

- Variable-length segment splits chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Step c, you also identified the chart that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

Exception class name: FRAGMENTED_FREE_SPACE

Exception class name: EXCESSIVE_SLACK_BYTES

These exceptions indicate that an excessive amount of unusable space exists in the data set. An excessive amount of unusable space wastes space and causes segments and database records to be fragmented.

FRAGMENTED_FREE_SPACE

This exception indicates an excessive amount of *fragmented unusable IMS free space*. Fragmented unusable IMS free space is fragmented IMS free space that is too small to store the shortest segment that is defined in the DBD definition.

EXCESSIVE_SLACK_BYTES

This exception indicates an excessive number of *slack bytes*. A slack byte is unusable space that is not used by a segment or that cannot be counted as IMS free space.

To learn more about slack bytes, see the topic "How IMS reclaims space" in the *IMS High Performance Pointer Checker User's Guide*.

These two exceptions indicate the same state in terms of space that cannot be reused by IMS. Even if you receive only one of these exceptions, analyze the state of the data sets from both perspectives to gain an overall understanding of the space that cannot be reused by IMS.

Complete these steps to analyze these exceptions:

- a. Locate the data elements that are related to IMS free space in the Sensor Data Statistics report:

DB_AVG_NUM_FSE

The value of this data element indicates the average number of IMS free space fragments in a CI or block.

DB_AVG_NUM_NOREUSE_FSE

The value of this data element indicates the average number of unusable IMS free space fragments in a CI or block. Unusable IMS free space refers to free space that is too small to store the shortest segment that is defined in the DBD definition. If the value of this data element is large, the database contains excessive amount of unusable space.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

Also, locate the following data elements. These data elements provide more information about the amount of remaining free space that IMS can use.

DB_NUM_FSE_MIN

The value of this data element indicates the number of IMS free space fragments in which the shortest segment that is defined in the DBD definition can be stored.

DB_NUM_FSE_MAX

The value of this data element indicates the number of IMS free space fragments in which the longest segment that is defined in the DBD definition can be stored.

DB_NUM_FSE

The value of this data element indicates the number of IMS free space fragments.

DB_BYTES_FREE_SPACE

The value of this data element indicates the total bytes of free space.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

Also, in some cases, determining if the length of fragmented IMS free space is large enough to store the segment of the segment type that has the most occurrences can help you analyze the condition of the IMS free space. You can do so by comparing the average length of IMS free space fragments with the segment length of the segment type that has the most occurrences. These values can be obtained as follows:

- i) Calculate the average length of IMS free space fragments by dividing the `DB_BYTES_FREE_SPACE` value by the `DB_NUM_FSE` value.
- ii) Determine the segment type that has the most occurrences in the data set by generating an IMS HP Pointer Checker report as explained in Step [c](#).

If the segment length of that segment type is shorter than the average length, you can assume that an excessive amount of unusable IMS free space exists.

- b. Locate the data elements that are related to slack bytes in the Sensor Data Statistics report:

DB_NUM_UNIDENTIFIED

The value of this data element indicates the number of slack bytes.

DB_BYTES_UNIDENTIFIED

The value of this data element indicates the total slack bytes.

DB_AVG_NUM_UNIDENTIFIED

The value of this data element indicates the average number of slack bytes in a CI or block.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

- c. Optional: To obtain more information, run the HD Pointer Checker utility of IMS HP Pointer Checker, or IMS HP Image Copy with the `HDPC=YES` option, and generate the HD Data Set Statistics report. In this report, review the Database record statistics part.

In the Database record statistics part, determine the segment type that has the most segment occurrences and the segment length for that segment type. Then compare the length of the segment with the average length of IMS free space fragments that you already calculated in Step [a](#). In an ideal data set, the average length of IMS free space fragments is typically longer than the segment length.

- d. Optional: To view the trend of data over time, locate the following charts in IMS Administration Foundation or Management Console:

- Unusable IMS free space chart
- Usable IMS free space chart
- Slack bytes areas chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Step [d](#), you also identified the charts that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

What to do next

If you received other exceptions that belong to other exception categories, see the following topics to determine your next step:

- [“Exception categories for full-function databases” on page 18](#)
- [“Scenario for using the analysis and resolution topics” on page 19](#)

If the only exceptions that you received are fragmentation exceptions, continue with [“Resolving fragmentation exceptions for HDAM and PHDAM databases” on page 46](#).

Analyzing optimization exceptions for HDAM and PHDAM databases

Analyze the optimization exceptions that were detected for HDAM and PHDAM databases.

Before you begin

Ensure that you can access the Sensor Data Statistics report that contains the latest sensor data. If you cannot access this report, run the FF DB Sensor Printing utility to generate the report.

About this task

In the following procedure, you will analyze each optimization exception that you received for HDAM databases or PHDAM partitions.

For each exception that was reported, you will first identify the relevant data elements to understand the condition of the database. Then you will analyze the relevant IMS Tools reports to learn more about the database state. Finally, if you use IMS Administration Foundation or Management Console and if relevant charts are available, you will analyze historical trends of sensor data to understand the trend characteristics of the sensor data that was obtained from the database or partition.

By completing these steps, you will determine the possible causes of the optimization exceptions. The causes can be the block or CI size not matching the average database record length, the randomizing routine, the existence of long database records, or combination of these factors.

Procedure

In this section, instructions to analyze the common optimization exceptions are provided. Follow the instructions for the exceptions that you received. The most common exceptions are listed first.

Exception class name: **NUMBER_OF_DB_RECORDS**

This exception indicates that the number of database records, which is the number of root segments, exceeded a threshold value.

This exception does not always indicate a problem in the database. When many root segments exist in the database, the number of synonym roots or root segments that are not in the home block tends to increase. Follow the analysis steps to determine if the data sets need to be worked on.

Complete these steps to analyze this exception:

- a. Locate the following data element in the Sensor Data Statistics report:

DB_NUM_ROOTS

The value of this data element indicates the number of root segments. The number of root segments is also the number of database records.

- b. Optional: To view the trend of data over time, locate the following chart in IMS Administration Foundation or Management Console:

- Number of database records chart

Understanding the trend over time helps you predict the timing to maintain the data sets. Based on this chart, determine the timing to take preventative actions and establish maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Step b, you also identified the chart that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

Exception class name: **EXCESSIVE_HDAM_ROOTS_NOT_HOME**

Exception class name: **EXCESSIVE_HDAM_OVERFLOW**

These exceptions indicate that too many root segments are not in the optimal CI or block.

When the root segment is not in the home block, extra I/O operations are required to read the root segments.

In HDAM or PHDAM databases, root segments are placed in the CI or block that is assigned by a randomizer. The most desirable location for these root segments is referred to as the *home block*. When all root segments are placed in their home blocks, the distribution of root segments is considered to be optimal.

However, when the space in the home block is insufficient, IMS places the root segment in other blocks, which might be outside of the root addressable area (RAA).

EXCESSIVE_HDAM_ROOTS_NOT_HOME

This exception indicates that too many root segments reside on the block that is not the home block.

EXCESSIVE_HDAM_OVERFLOW

This exception indicates that too many segments are in the overflow area. The overflow area is outside of the RAA.

Complete these steps to analyze these exceptions:

- a. Locate the following data elements in the Sensor Data Statistics report:

DB_NUM_ROOT_NOHOME

DB_PCT_NUM_ROOT_NOHOME

The values for these data elements indicate the number of root segments that are not in the home block, and the percentage of these root segments compared to the total number of root segments.

DB_NUM_ROOT_OVFL

DB_PCT_NUM_ROOT_OVFL

The values for these data elements indicate the number of root segments in the overflow area, and the percentage of these root segments compared to the total number of root segments.

- b. Optional: To obtain more information about the root segments, run the HD Pointer Checker utility of IMS HP Pointer Checker, or IMS HP Image Copy with the HDPC=YES option, and generate the DB Record Distribution Statistics report. In this report, review the distribution of root segments part.

By analyzing this report, you can determine how far the root segments are from the home block. Typically, most of the root segments are preferred to be in HOMEBLOCK-1, HOMEBLOCK, and HOMEBLOCK+1.

- c. Optional: To view the trend of data over time, locate the following charts in IMS Administration Foundation or Management Console:

- Root outside home block chart
- Roots in overflow area chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Step [c](#), you also identified the charts that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

Exception class name: EXCESSIVE_RAP_SYNONYMS

Exception class name: IMBALANCED_RANDOMIZING

These exceptions indicate an exceptional state of the synonym roots.

When synonym roots occur frequently, you must determine the state of the database. The state can be either the number of root anchor points (RAPs) are insufficient with respect to the number of roots, or many synonym chains exist even though unused RAPs exist.

EXCESSIVE_RAP_SYNONYMS

This exception indicates that many synonym roots exist.

IMBALANCED_RANDOMIZING

This exception indicates that many synonym roots exist even though unused RAPs exist.

Complete these steps to analyze these exceptions:

- a. Locate the following data elements in the Sensor Data Statistics report to determine the current database state that caused this exception:

DB_NUM_SYNONYM

DB_PCT_NUM

The values for these data elements indicate the number of synonym roots, and the percentage of synonym roots compared to the total number of roots.

DB_PCT_NUM_UNUSED_RAP

DB_PCT_NUM_SYNONYM

The values for these data elements indicate the percentage of unused RAPs compared to the total number of RAPs, and the percentage of synonyms compared to the total number of root segment occurrences.

DB_ESTIMATED_ROOT_IO

The value for this data element indicates the estimated number of I/O operations that are required to read a root segment (that is, the average number of reading different CIs or blocks to read from the RAP to a root segment by tracking the synonym chain).

Note: The value is an estimated value; the actual I/O count might be different.

Even when many synonyms exist, if the synonym roots on the same synonym chain are in the same CI or block, no extra I/O operations are required to read the root segment. In this case, you do not necessarily need to resolve this exception. If the DB_ESTIMATED_ROOT_IO value is high, extra I/O operations are likely needed; therefore, you should consider taking actions to resolve this status.

- b. Optional: To obtain more information about the used RAPs and the synonym roots, run the HD Pointer Checker utility of IMS HP Pointer Checker, or IMS HP Image Copy with the HDPC=YES option, and generate the DB Record Distribution Statistics report. Review the distribution of RAP chain lengths part.

- c. Optional: To view the trend of data over time, locate the following chart in IMS Administration Foundation or Management Console:

- Synonym roots chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Step c, you also identified the chart that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

Exception class name: AVERAGE_DB_RECORD_LENGTH

This exception indicates that the average length of database records has exceeded the threshold.

This exception does not always indicate a problem in the database. When the database record length is long, the segments that belong to a same database record might be spread across multiple CIs or blocks, and might be causing extra I/O operations.

You can identify whether extra I/O operations are required by reviewing the DB_ESTIMATED_DBREC_IO data element value. When the value is small, you do not need to analyze this exception. The smallest value, which typically is the optimal value, is the number of data sets. If this value is large, you can improve the condition by reorganizing the database; however, you might be able to improve the condition even further by performing optimization tuning (determining optimal DBD definitions and randomizing parameters, and applying the changes).

The method that you use to analyze the database depends on the number of database data sets:

Single data set

When a database has one data set, all segments reside in that data set. In the most optimized condition, all the segments that belong to a same database record reside in the same CI or block. However, when fragmented IMS free space is shorter than the database record, which means that all segments that belong to the database record cannot be stored in a single CI or block, IMS places some of the segments in other CIs or blocks. This state can cause extra I/O operations.

To determine if extra I/O operations are occurring, locate the following data elements in the Sensor Data Statistics report:

DB_AVG_DBREC_LENGTH

The value for this data element indicates the average length of database records.

DB_NUM_ROOT

The value for this data element indicates the number of root segments (that is, the number of database records).

DB_BLOCK_SIZE

The value for this data element indicates the CI or block size.

DB_ESTIMATED_DBREC_IO

The value for this data element indicates the estimated number of I/O operations that are required to read a database record (that is, the average number of reading different CIs or blocks to read the root segment and all dependent segments by tracking the hierarchical path).

Note: The value is an estimated value; the actual I/O count might be different.

To obtain more information about the segment distribution, run the HD Pointer Checker utility of IMS HP Pointer Checker, or IMS HP Image Copy with the HDPC=YES option, and generate the DB Record Distribution Statistics report.

By reviewing the Distribution of dependent segments by segment code part, you can determine, for each segment type, the number of segments that are not in the block or CI where its root segment is stored, and the distribution of these segments across different CIs or blocks.

The information in this report can be a good indicator for tuning the CI or block size to prevent database records being spread across different CIs or blocks.

Multiple data sets

When a database has multiple data sets, the segments that belong to the same database record can be stored across multiple data sets. Therefore, reviewing the average database record length will not help you identify the data sets in which the segments that increased the database record length reside. To identify the data set that has the most occurrences of scattered segments, follow the analysis steps for the [EXCESSIVE_SEGMENT_SCATTERING](#) exception (a fragmentation exception).

If you identify excessive segment scattering in data set group 1 (DSG1), obtain more information about the segment distribution by running the HD Pointer Checker utility of IMS HP Pointer Checker, or IMS HP Image Copy with the HDPC=YES option, and generate the DB Record Distribution Statistics report.

By reviewing the Distribution of dependent segments by segment code part, you can determine, for each segment type, the number of segments that are not in the block or CI where its root segment is stored, and the distribution of these segments across different CIs or blocks within DSG1.

Together with the CI or block size, this information can be a good indicator for tuning the CI or block size to prevent segments being spread across different CIs or blocks.

If excessive segment scattering was not observed in DSG1, you do not need to perform optimization tuning. Reorganizing the database will resolve the exception.

The average database record length can be a trigger to start database analysis. In the following charts of IMS Administration Foundation or Management Console, you can review the trend of the average database record length and the number of database records:

- Average database record length chart

- Number of database records chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

You also identified the charts that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

What to do next

If you received other exceptions that belong to other exception categories, see the following topics to determine your next step:

- [“Exception categories for full-function databases” on page 18](#)
- [“Scenario for using the analysis and resolution topics” on page 19](#)

If the only exceptions that you received are optimization exceptions, continue with [“Resolving optimization exceptions for HDAM and PHDAM databases” on page 47](#).

Resolving space use exceptions for HDAM and PHDAM databases

Resolve the space use exceptions that you received for your HDAM and PHDAM databases.

Before you begin

Ensure that you have completed the analysis steps for HDAM and PHDAM exceptions and that you have identified the possible causes of the database exceptions. If you have not yet analyzed the space use exceptions, see [Chapter 4, “Analyzing space use exceptions for all full-function database types,” on page 29](#).

Procedure

If you receive space use exceptions, and if you identify an out-of-space condition in database data sets as a result of analyzing the space use exceptions, consider resolving the exception condition by completing the following actions:

- If you also receive fragmentation exceptions, and if you identify excessive fragmentation of IMS free space or the existence of many unusable IMS free space fragments, consider reorganizing the database. Reorganization will eliminate the fragmentation and increase usable IMS free space in the data sets; subsequently, new segments can be added to the database.
- After you analyze the exceptions that you received, if you determine that reorganizing the database will not create sufficient IMS free space, and if you have not applied a Segment Edit/Compression exit routine, consider applying the routine. For information about applying the routine, see the topic "Segment Edit/Compression exit routines" in *IMS Database Administration*.

If the out-of-space conditions are not resolved by these actions, consider changing the database structure or increasing the DASD resources. The method you apply depends on the results of the analysis steps as follows:

Tip: Depending on the status of the data set, applying several of the following methods can be an effective approach to resolving the exceptions. Before making any changes to your environment, evaluate all the methods in the following list and determine the best approach to resolve the condition.

- If the out-of-space condition is caused by a *primary-space-used-up* condition (that is, the data set has only primary space, and the size of used space is reaching the size of the primary allocated space), and if the data sets allocation size is smaller than the maximum size (4 GB or 8 GB), increase the size of the allocated space. For instructions for increasing the size of allocated space, see the topic "Changing the amount of space allocated" in *IMS Database Administration*.

- If the out-of-space condition is caused by a *data-set-full* condition (that is, the size of the data set is approaching its maximum size of 4 GB or 8 GB), typically, changing the DBD definition and then restructuring the database, migrating the database, or both will resolve the condition.

Consider the following methods to resolve a data-set-full condition:

- If the database data set is VSAM, consider changing the data set type to OSAM. By changing the data set type to OSAM, you can increase the maximum size of the data set from 4 GB to 8 GB.
- Split the database into several smaller databases.
- Increase the number of database data sets. The maximum number of data sets is 10. When you increase the number of data sets, be aware that I/O counts for accessing database records also increase.
- If the database is an HDAM database, migrate the database from HDAM to PHDAM.
- If the database is PHDAM, increase the number of partitions.

For more information about these tasks, see the following topics in *IMS Database Administration*:

- "Changing operating system access methods"
 - "Using OSAM as the access method"
 - "Multiple data set groups"
 - "Changing the number of data set groups"
 - "Converting database types"
 - "Adding partitions to an existing HALDB database"
- If the out-of-space condition is caused by an *extent-resources-used-up* condition (that is, when the data set cannot be expanded), typically, changing the data set allocation definitions and reorganizing the database will resolve the condition.

Consider the following methods to resolve an extent-resources-used-up condition:

- If the data set has reached the maximum number of data set extent counts, and if the DASD volumes have enough space, increase the primary and the secondary extent sizes for data set allocation.
- If the DASD volume does not have enough space or if the data set has reached the maximum number of extents that is defined for that volume, increase the number of DASD volumes.

Generally, as the number of extents or the number of DASD volumes increases, I/O operations also increase. Therefore, consider using a DASD volume that has sufficient size, and specify an extent size that is large enough for data set allocation.

Resolving fragmentation exceptions for HDAM and PHDAM databases

Resolve the fragmentation exceptions that you received for your HDAM and PHDAM databases.

Before you begin

Ensure that you have completed the analysis steps for HDAM and PHDAM exceptions and that you have identified the possible causes of the database exceptions. If you have not yet analyzed the fragmentation exceptions, see [“Analyzing fragmentation exceptions for HDAM and PHDAM databases” on page 37](#).

Procedure

If the only exceptions that you received are fragmentation exceptions, reorganize the database. After you reorganize the database, rerun a policy evaluation to ensure that the exceptions are resolved.

After the reorganization, the EXCESSIVE_SEGMENT_SCATTERING exception might remain, or the values of the data elements that are related to this exception might continue to be high (for example, if this exception was reported as a critical level exception, the exception might be reported as a severe or

warning level exception after the reorganization.) If you receive this exception again, follow the steps for resolving optimization exceptions.

Resolving optimization exceptions for HDAM and PHDAM databases

Resolve the optimization exceptions that you received for your HDAM and PHDAM databases.

Before you begin

Ensure that you have completed the analysis steps for HDAM and PHDAM exceptions and that you have identified the possible causes of the database exceptions. If you have not yet analyzed the optimization exceptions, see [“Analyzing optimization exceptions for HDAM and PHDAM databases” on page 41](#).

Procedure

If you receive optimization exceptions, you must collect more information about the database data sets to evaluate the requirements for the database definition, randomizer, randomizing parameters, and other factors.

If you determine that you must change these database resources, plan a database reorganization to apply the changes.

Tip: You can use the HD Tuning Aid utility of IMS HP Pointer Checker to simulate DBD changes and randomizer changes. By using this utility, you can determine the optimal randomizing parameters without making changes to the DBD definitions or the database.

To collect more information and to evaluate the requirements for your HDAM and PHDAM databases, see the following topics:

- [“Average number of I/O operations per database record” on page 47](#)
- [“Tuning a randomizer” on page 48](#)
- [“Tuning HDAM database that have long database records” on page 59](#)

Average number of I/O operations per database record

HDAM randomizing is considered to be efficient if the average number of I/O operations that are required to randomly read all database segments of one database record is low. The average number of I/Os required to read at random all database segments of one database record is one of the most important indicators of the quality of the randomizing.

The average number of I/O operations is reported in the DB Tuning Statistics report. By looking at this average number, the database administrator can determine if a database is randomized efficiently without analyzing the entire report. For information about how to generate this report, see [“Information resources for analyzing exceptions” on page 25](#).

For an ideal database that consists of a single data set group, this average number is 1.0. However, in a real production database, this ideal value is typically difficult to achieve and the average number of database I/Os per database record is higher. Achieving a good randomizing value is easier when the block size, divided by average database record length, is large.

As a general guideline, a database is considered as well randomized if the average number of I/Os per database record meets the following criteria:

- Is below 1.20 for databases with an average database record length below one tenth of the block or CI size
- Is below 1.30 for databases with larger average database record lengths

Numbers above 1.20 and 1.30 might indicate that a database is poorly randomized (unless the database record length is large). In this case, the database administrator can use other numbers that are provided by the reports listed in [“Information resources for analyzing exceptions” on page 25](#) to determine the

reason for the poor randomizing. See [“Tuning a randomizer” on page 48](#) for information about how to use those numbers to tune a randomizer.

For HDAM databases with a high percentage of very long database records, achieving a low value for the average number of I/O per database record is difficult. See [“Tuning HDAM database that have long database records” on page 59](#) for information about using reports that are produced by the IPR Unload utility to tune HDAM databases that have long records.

Tuning a randomizer

This topic discusses how to fit as many root segments as possible in the home block when you find out that the randomizing process is not functioning well.

If, in an HDAM database, there are many root segments that do not fit in the home block (the home block is the block the root segment must be placed in, based on the randomizing routine), the randomizing parameters might not be functioning well.

Whether the randomizing process is functioning well or not can be checked by the steps in [“Example to assess the effectiveness of the randomizing module” on page 48](#).

If there are more root segments than can be fit in the home block, check the factors discussed in the following topics:

- [“Example to assess the effectiveness of the randomizing module” on page 48](#)
- [“Is the number of RAPs appropriate?” on page 50](#)
- [“Is the size of the Root Addressable Area appropriate?” on page 52](#)
- [“Is the HDAM bytes limit appropriate for the database record lengths in this database?” on page 54](#)
- [“Is the block size or CI size appropriate for the database record lengths of this database?” on page 56](#)
- [“Is the amount of free space specified during DBDGEN equal to zero?” on page 56](#)
- [“Specifying randomizing parameters of an HDAM database” on page 57](#)

Example to assess the effectiveness of the randomizing module

To check to which home block the root segment belong, see the HDAM Root Placement section of the HDAM Root Addressable Area Statistics report.

This section shows the distribution of the locations—the position before or after the home block—of root segments. You can also check the root segments that do not fit in the home block from NUMBER OF ROOTS NOT IN THE RANDOMIZED BLOCK in the Key Indicators for Quality of Randomizing section of the HDAM Root Addressable Area Statistics report. [Figure 6 on page 49](#) and [Figure 7 on page 49](#) show cases where problems exist.

The following figure shows that there are many root segments that do not fit in the home block; the value of NUMBER OF ROOTS NOT IN THE RANDOMIZED BLOCK is high.

"HDAM ROOT ADDRESSABLE AREA STATISTICS"

RANDOMIZER=DFSHDC40 NBR OF RAPS=2 MAX RBN=200 MAX BYTES=4,000

<<< KEY INDICATORS FOR QUALITY OF RANDOMIZING >>>

NUMBER OF ROOT SEGMENTS.....	2,200
NUMBER OF ROOT SYNONYM CHAINS.....	383
NUMBER OF ROOTS PER SYNONYM CHAIN.....	9.4
NUMBER OF ROOTS NOT ON A SYNONYM CHAIN.....	1,406
NUMBER OF ROOTS ON A SYNONYM CHAIN.....	3,606
NUMBER OF ROOTS IN LONGEST SYNONYM CHAIN.....	16
NUMBER OF ROOTS NOT IN THE RANDOMIZED BLOCK...	1,610
NUMBER OF ROOTS PER ROOT ANCHOR POINT.....	5.5
NUMBER OF ROOT ANCHOR POINTS PER ROOT.....	0.2
NUMBER OF ROOT ANCHOR POINTS USED.....	397
NUMBER OF ROOT ANCHOR POINTS NOT USED.....	3
PERCENT OF ROOT ANCHOR POINTS USED.....	99.3%

Figure 6. Database tuning of the primary data set group in an HDAM database: sample 1

The following figure shows that there are more segments that reside on blocks other than the home block than those that reside on the home block.

"HDAM ROOT ADDRESSABLE AREA STATISTICS"

<<< HDAM ROOT PLACEMENT >>>

LOCATION	PCT	QUANTITY
- (11-)	1.8%	40
- 10	0.1%	3
- 9	0.0%	0
- 8	0.0%	1
- 7	0.1%	2
- 6	0.0%	0
- 5	0.0%	0
- 4	0.0%	1
- 3	0.1%	2
- 2	0.1%	3
- 1	0.1%	2
HOME BLOCK	26.8%	590
+ 1	0.1%	3
+ 2	0.1%	3
+ 3	0.2%	4
+ 4	0.1%	2
+ 5	0.0%	1
+ 6	0.1%	2
+ 7	0.1%	3
+ 8	0.1%	2
+ 9	0.0%	1
+ 10	0.0%	0
+ (11-)	4.1%	91
OVERFLOW	65.6%	1,444

Figure 7. Database tuning of the primary data set group in an HDAM database: sample 2

Figure 8 on page 50 and Figure 9 on page 50 show cases where the problems are solved by reloading the database.

The following figure shows that the number of root segments that do not fit in the home block has decreased; the value of NUMBER OF ROOTS NOT IN THE RANDOMIZED BLOCK has decreased.

"HDAM ROOT ADDRESSABLE AREA STATISTICS"

RANDOMIZER=DFSHDC40 NBR OF RAPS=7 MAX RBN=385 MAX BYTES=1,000

<<< KEY INDICATORS FOR QUALITY OF RANDOMIZING >>>

NUMBER OF ROOT SEGMENTS.....	2,200
NUMBER OF ROOT SYNONYM CHAINS.....	526
NUMBER OF ROOTS PER SYNONYM CHAIN.....	2.3
NUMBER OF ROOTS NOT ON A SYNONYM CHAIN.....	985
NUMBER OF ROOTS ON A SYNONYM CHAIN.....	1,215
NUMBER OF ROOTS IN LONGEST SYNONYM CHAIN.....	N/A
NUMBER OF ROOTS NOT IN THE RANDOMIZED BLOCK...	51
NUMBER OF ROOTS PER ROOT ANCHOR POINT.....	1.5
NUMBER OF ROOT ANCHOR POINTS PER ROOT.....	1.2
NUMBER OF ROOT ANCHOR POINTS USED.....	1,511
NUMBER OF ROOT ANCHOR POINTS NOT USED.....	1,184
PERCENT OF ROOT ANCHOR POINTS USED.....	56.1%

Figure 8. Database tuning of the primary data set group in an HDAM database: sample 3

The following figure shows that the number of root segments in the home block has increased.

"HDAM ROOT ADDRESSABLE AREA STATISTICS"

<<< HDAM ROOT PLACEMENT >>>

LOCATION	PCT	QUANTITY
- 3	0.3%	6
- 2	0.5%	11
- 1	2.7%	59
HOME BLOCK	96.4%	2,121
+ 1	0.1%	3
+ 2	0.0%	0
+ 3	0.0%	0
OVERFLOW	0.0%	0

Figure 9. Database tuning of the primary data set group in an HDAM database: sample 4

"Tuning a randomizer" on page 48 describes the details on how to tune a database.

Is the number of RAPs appropriate?

Use the values in the following reports to determine whether the number of RAPs is appropriate.

HDAM Root Addressable Area Statistics report

- NUMBER OF ROOTS PER ROOT ANCHOR POINT in the Key Indicators for Quality of Randomizing section

This value is obtained by dividing the number of root segments (NUMBER OF ROOT SEGMENTS) by the number of RAPs that have at least one root segment assigned (NUMBER OF ROOT ANCHOR POINTS USED). When there are many synonyms, this value is large.

- NUMBER OF ROOT ANCHOR POINTS PER ROOT in the Key Indicators for Quality of Randomizing section

This value is obtained by dividing the number of total RAPs by the number of root segments (NUMBER OF ROOT SEGMENTS). You can check the ratio of the number of RAPs specified to the root segments. A recommended value for this field is 1.2.

Generally, the number of RAPs must be approximately 1.2 times the number of database records. When in doubt, it is better to specify too many rather than too few RAPs (because a RAP is only 4 bytes and hence inexpensive in terms of DASD space)^{Note}. For details, see *IMS Database Administration*.

Note: An exception to this recommendation is an Intersystem block-level sharing environment with sequential database processing. In such an environment, the number of RAPs must not be too large because access to each RAP might create intersystem block-level sharing delays.

Discussion of the example:

In [Figure 6 on page 49](#), the value of NUMBER OF ROOT ANCHOR POINTS PER ROOT is small, which can mean that the number of RAPs is not enough. This might be the cause of the value NUMBER OF ROOTS PER ROOT ANCHOR POINT being so large.

The following figure shows a case where the database was reloaded after increasing the value of MAX RBN (the maximum RBN value, one of the randomizing parameters).

```

IMS HIGH PERFORMANCE LOAD                                PAGE:          1
5655-M26                                                10/01/2021  13.10.14

                "HDAM ROOT ADDRESSABLE AREA STATISTICS"

RANDOMIZER=DFSHDC40  NBR OF RAPS=2      MAX RBN=1,320      MAX BYTES=4,000

<<< KEY INDICATORS FOR QUALITY OF RANDOMIZING >>>

NUMBER OF ROOT SEGMENTS.....                2,200
NUMBER OF ROOT SYNONYM CHAINS.....           531
NUMBER OF ROOTS PER SYNONYM CHAIN.....        2.3
NUMBER OF ROOTS NOT ON A SYNONYM CHAIN.....   961
NUMBER OF ROOTS ON A SYNONYM CHAIN.....      1,239
NUMBER OF ROOTS IN LONGEST SYNONYM CHAIN..... N/A
NUMBER OF ROOTS NOT IN THE RANDOMIZED BLOCK... 79
NUMBER OF ROOTS PER ROOT ANCHOR POINT.....    1.5
NUMBER OF ROOT ANCHOR POINTS PER ROOT.....    1.2
NUMBER OF ROOT ANCHOR POINTS USED.....       1,492
NUMBER OF ROOT ANCHOR POINTS NOT USED.....    1,148
PERCENT OF ROOT ANCHOR POINTS USED.....       56.5%

```

Figure 10. Database tuning of the primary data set group in an HDAM database: sample 5

Because MAX RBN has increased, NUMBER OF ROOT ANCHOR POINTS PER ROOT has also increased. This means that the problem of insufficient number of RAPs has been solved. Also, NUMBER OF ROOTS PER ROOT ANCHOR POINT has decreased, which means the number of synonyms has also decreased.

Use the values in the following reports to determine the appropriate number of RAPs assigned for a block or a CI.

Data Set Statistics report

You can check the block size or the CI size and the number of blocks or the number of CIs that specify about the data set.

Distribution of Database Record Sizes report

CUM PCT

You can check what percent of the database records can fit on the length that you want to specify.

You can use the value to determine the appropriate number of RAPs for that block. For multiple data set groups, however, you cannot use the value as it is.

Discussion of the example:

The following description is true only when there is only one data set group and when the bytes limit—the maximum number of bytes of a database record that can be inserted into this database's root segment addressable area; one of the randomizing parameters—is not specified.

The following figure shows that the block size or the CI size is 8192 bytes. [Figure 12 on page 52](#) shows that the size of 18.0% of database records is 2000 bytes or less, and 92.7% is 4000 bytes or less.

"DATA SET STATISTICS"

DSG	DDNAME	BLOCK SIZE	RECORD SIZE	DATA SET NAME
1	DDHDV10	8,192	8,185	TESTDS.NTFC5201.DBHDV10.DDHDV10

DSG	DDNAME	FBFF	FSPF	FSPF BYTES	SMALL SEGSZ	LARGE SEGSZ	TOTAL BLOCKS
1	DDHDV10	0	0	0	86	1,006	629

NUMBER OF BLOCKS IN OVERFLOW = 429

Figure 11. Database tuning of the primary data set group in an HDAM database: sample 6

"DISTRIBUTION OF DATABASE RECORD SIZES"

RANGE	QUANTITY	PCT	CUM PCT
- 100	0	0.0%	0.0%
101 - 200	0	0.0%	0.0%
201 - 300	0	0.0%	0.0%
301 - 400	20	0.9%	0.9%
401 - 500	0	0.0%	0.9%
501 - 600	0	0.0%	0.9%
601 - 700	0	0.0%	0.9%
701 - 800	0	0.0%	0.9%
801 - 900	0	0.0%	0.9%
901 - 1000	0	0.0%	0.9%
1001 - 2000	375	17.0%	18.0%
2001 - 3000	1,549	70.4%	88.4%
3001 - 4000	95	4.3%	92.7%
4001 - 5000	154	7.0%	99.7%
5001 - 6000	7	0.3%	100.0%
6001 - 7000	0	0.0%	100.0%
7001 - 8000	0	0.0%	100.0%
8001 - 9000	0	0.0%	100.0%
9001 - 10000	0	0.0%	100.0%
10001 -	0	0.0%	100.0%

AVERAGE DATABASE RECORD SIZE = 2,317
 SHORTEST DATABASE RECORD SIZE = 318
 LONGEST DATABASE RECORD SIZE = 5,808

Figure 12. Database tuning of the primary data set group in an HDAM database: sample 7

If, for instance, the number of RAPs per block or CI is 4, 18.0% of database records fit both the root and the dependent segments on one block or on one CI. If, for instance, the number of RAPs per a block or a CI is 2, 92.7% of database records fit both the root and the dependent segments on one block or on one CI.

When the dependent segments reside on the same block or CI as the root segment, the access time is shorter (than when the segments reside on different blocks or CIs). However, if the frequency of access to the dependent segments is low, where these two types of segments reside is not a significant factor for performance. Therefore, trade-off between the access speed and the size of the database data set must be considered according to the operation.

Is the size of the Root Addressable Area appropriate?

This topic discusses whether the size of the Root Addressable Area is appropriate.

As a reasonable rule of thumb, if you are using a standard DFSHDC40 randomizer or the Sequential Subset Randomizer, the percentage of space used for segments in an RAA (shown as OTHER KEY INDICATORS in the HDAM RAP Chain I/O Summary report of the IPR Unload utility, and referred to as *packing density*) must be as follows:

75% (for databases with an average database record length smaller than one tenth of the CI/block size)

70% (for databases with a larger average database record length)

Notes:

1. Aiming for higher packing densities saves DASD space, but often decreases the performance of HDAM database accesses. Each installation will have probably its own idea of what is the ideal trade-off between DASD space and performance and regarding the ideal packing density. For example, some installations might want packing densities of 75% and 80% (instead of the target values of 70% and 75%).
2. For small or medium-sized HDAM databases (when saving DASD space is not important), it is often reasonable to aim for a packing density below 70%.
3. If the total size of all database records will eventually grow, it is then reasonable to oversize the root addressable area at database reload time, in order to provide enough space for the future data growth.

You can change the packing density of the root addressable area by varying the number of blocks or CIs in the root addressable area and by varying the size of a block or CI. Increasing the number of blocks or CIs or increasing the block or CI size increases the size of the root addressable area and lower its packing density; this typically increases the performance of random accesses to the database.

Varying the bytes limit might also change the packing density of the root addressable area (reducing the bytes limit will tend to store more information in the overflow area and less information in the root addressable area; this will hence tend to reduce the packing density in the root addressable area).

Discussion of the example

Packing Density is printed under OTHER KEY INDICATORS in the HDAM RAP Chain I/O Summary report of the IPR Unload utility.

In the following figure, the value of PACKING DENSITY OF ROOT ADDRESSABLE AREA is too high. To lower the value of packing density, you can increase the number of blocks or the number of CIs in the RAA, or you can decrease the total bytes of the segments in the RAA, and so on. Make appropriate specifications, taking into account the pros and cons previously discussed.

"HDAM RAP CHAIN I/O SUMMARY"

KEY INDICATORS FOR QUALITY OF RANDOMIZING FROM A VIEWPOINT OF DATABASE I/O

AVG NBR OF I/O PER DB RECORD.....	2.91
AVG NBR OF I/O ON RAP CHAIN PER ROOT.....	1.66
AVG NBR OF I/O ON RAP CHAIN PER RAP CHAIN.....	2.00
AVG NBR OF I/O IN ROOT ADDRESSABLE AREA PER DB RECORD...	1.50
AVG NBR OF I/O IN OVERFLOW AREA PER DB RECORD.....	1.41
AVG POSITION OF ROOT SEGMENTS ON RAP CHAINS.....	1.33

OTHER KEY INDICATORS

PACKING DENSITY OF ROOT ADDRESSABLE AREA.....	46.09
---	-------

NBR OF I/O'S REQUIRED TO READ A ROOT SEGMENT RANDOMLY
(SIMULATED WITH 4 PCB-BUFFERS)

NBR I/O	NBR ROOTS	PCT	CUM PCT
1	4	33.33%	33.33%
2	8	66.66%	100.00%

POSITION OF ROOT SEGMENTS ON RAP CHAINS

POSITION	NBR ROOTS	PCT	CUM PCT
1	8	66.66%	66.66%
2	4	33.33%	100.00%

Figure 13. Database tuning of the primary data set group in an HDAM database: sample 8

You can check the block size or the CI size and the number of blocks or the number of CIs of the current database data sets in the Data Set Statistics report. You can check the average size of database records in AVERAGE DATABASE RECORD SIZE of the Distribution of Database Record Sizes report.

The value of the bytes limit (MAX BYTES) that is specified is printed in the top part of the HDAM Root Addressable Area Statistics report (and also in the DBD Summary report for a non-HALDB).

Figure 6 on page 49 shows that MAX BYTES= is blank, which means the bytes limit is not specified. For details, see [“Is the HDAM bytes limit appropriate for the database record lengths in this database?”](#) on page 54.

Is the HDAM bytes limit appropriate for the database record lengths in this database?

The bytes limit is the maximum number of bytes of a database record that can be stored into the root addressable area in a series of insert calls unbroken by a call to another database record.

The bytes limit is shown in the top part of the HDAM Root Addressable Area Statistics report.

The database administrator needs to perform a trade-off when specifying a bytes limit. Specifying a bytes limit that is too small results in the storing of too many database segments into the overflow area. Access to these many database segments in the overflow area typically triggers additional I/Os (since these database segments are not stored in the same block or CI as the root segment in the root addressable area); these additional I/Os adversely affect the performance of access to the HDAM database.

Specifying a bytes limit that is too high (for example, 10 times the average database record length) might create a situation where one individual database record occupies too much space in the root addressable area. Other database records of the same block or CI or of neighboring blocks or CIs will then have an insufficient amount of DASD space available for the storing of their own database segments in the same block or CI as the RAP. Access to the database segments of these other database records requires additional I/Os. This adversely affects the performance of access to the HDAM database.

Note: Specification of a bytes limit that is too high often creates a *cascade* effect. If, for example, one database record randomizing to block n is allowed to store as many as five block sizes worth of data in the root addressable area, then the database records that are randomized to neighboring blocks (for example, block n, n+1, n+2, n+3, and n+4) cannot be stored in the same block as their RAPs. The segments of these database records will spill into blocks n+5, n+6, and so on, and will create problems for the database records that are randomized in block n+5, n+6, and so on. Note that the PSSR utility of IMS HP Load assists in preventing such cascading effects.

Discussion of the example

See AVG SEGM LENGTH and AVG OCCURRENCES PER RECORD in the Segment Statistics report. The sum of the products of AVG SEGM LENGTH and AVG OCCURRENCES PER RECORD of the segments that are used frequently is compared with the specified bytes limit. You can check whether the bytes limit is appropriate.

In [Figure 14](#) on [page 55](#), the bytes limit is specified as 1000. [Figure 15](#) on [page 56](#) shows that you can put segments from ROOTLEV1 to DEP2LEVE2 in RAA with no problem, but if you want to put all segments in RAA, you might have to specify a value larger than 1000 for the bytes limit.

```

IMS HIGH PERFORMANCE LOAD                                PAGE:          1
5655-M26                                                10/01/2021  13.10.14

                "HDAM ROOT ADDRESSABLE AREA STATISTICS"

RANDOMIZER=DFSHDC40  NBR OF RAPS=7    MAX RBN=385    MAX BYTES=1,000

<<< KEY INDICATORS FOR QUALITY OF RANDOMIZING >>>

NUMBER OF ROOT SEGMENTS.....                2,200
NUMBER OF ROOT SYNONYM CHAINS.....           526
NUMBER OF ROOTS PER SYNONYM CHAIN.....        2.3
NUMBER OF ROOTS NOT ON A SYNONYM CHAIN.....   985
NUMBER OF ROOTS ON A SYNONYM CHAIN.....      1,215
NUMBER OF ROOTS IN LONGEST SYNONYM CHAIN..... N/A
NUMBER OF ROOTS NOT IN THE RANDOMIZED BLOCK... 51
NUMBER OF ROOTS PER ROOT ANCHOR POINT.....    1.5
NUMBER OF ROOT ANCHOR POINTS PER ROOT.....    1.2
NUMBER OF ROOT ANCHOR POINTS USED.....        1,511
NUMBER OF ROOT ANCHOR POINTS NOT USED.....    1,184
PERCENT OF ROOT ANCHOR POINTS USED.....       56.1%

```

Figure 14. Database tuning of the primary data set group in an HDAM database: sample 9

"SEGMENT STATISTICS"

SEGMENT STATISTICS

SEGMENT	TOTAL NBR OF OCCURRENCES	<-----AVG OCCURRENCES-----> PER PARENT	PER RECORD	AVG SEGM LENGTH	<--CONTRIBUTION TO DB RECORD SIZE--> PCT	CUM SIZE
ROOTLEV1	2,200	1.0	1.0	318	20.0%	318
DEP1LEV2	314	0.1	0.1	786	7.1%	430
DEP2LEV2	14,217	6.5	6.5	50	20.3%	753
DEP3LEV3	8,547	0.6	3.9	46	11.3%	932
DEP4LEV2	28,290	12.9	12.9	51	41.3%	1,588

SEGMENT IN OVERFLOW STATISTICS

SEGMENT	<-----OVERFLOW-----> NUMBER	PCT
ROOTLEV1	0	0.0%
DEP1LEV2	298	94.9%
DEP2LEV2	3,001	21.1%
DEP3LEV3	5,019	58.7%
DEP4LEV2	17,849	63.1%
TOTAL SEGMENTS =		53,568

Figure 15. Database tuning of the primary data set group in an HDAM database: sample 10

Is the block size or CI size appropriate for the database record lengths of this database?

This topic discusses the block size or CI size for the database record lengths.

A general rule of thumb is a 4 KB CI/block size for HDAM. However, if the average database record length is higher than 800 bytes (that is, larger than 1/5th of the block or CI size), then an increase of the block or CI size from 4 KB to 8 KB might improve the performance of HDAM database accesses (especially if the database administrator wants a high packing density (more than 70%) of the root addressable area). In this case, an increase of the CI size/block size from 4 KB to 8 KB can often reduce the number of I/O operations (but will increase slightly the time of an I/O operation). CI/block sizes larger than 8 KB should be an exception (for example, for very large average database record sizes).

Discussion of the example

You can check the current block size or the CI size, and the number of blocks and the number of CIs in the Data Set Statistics report. You can also check the average length of database records in AVERAGE DATABASE RECORD SIZE of the Distribution of Database Record Sizes report. You need to make specifications correctly, taking into account the pros and cons discussed.

Is the amount of free space specified during DBDGEN equal to zero?

This topic discusses whether the amount of free space that is specified during DBDGEN is equal to zero or not.

Free space within each block/CI

Specification on the DATASET macro of DBDGEN of free space within each block or CI has both advantages and disadvantages.

Free space within each block or CI is often advantageous, if the database will have many insert activities after database load/reload in the overflow area. In this case, free space specifications often reduce the scattering of database segments of the same database record into a large number of different blocks or CIs during subsequent insert activities.

The percentage of specified free space must be taken into account when determining the size of the RAA. For example, when specifying 10% of free space, some database administrators increase the size of the root addressable area by 10%, since the free space is not available at database load time or database reload time.

Free space within each block or CI is disadvantageous if few database segments are inserted after database load/reload or if nearly all database segments can be stored in the root addressable area.

Free block frequency factor

As explained in *IMS Database Administration*, specifying a free block frequency factor on the DATASET macro of DBDGEN for the first data set group of HDAM is self-defeating. For data set groups other than the first HDAM data set group, a free block frequency specification is often useful.

Discussion of the example:

You can check the definition information of free space in the Dataset Statistics report. You need to specify appropriate values taking into considerations the pros and cons discussed.

Specifying randomizing parameters of an HDAM database

This topic is provided to assist database administrators who do not have experience with the specifications of efficient HDAM randomizing parameters.

This topic contains general rules of thumb for using a "first guess" method to determine the values of the following parameters, which have an impact on the randomizing performance of an HDAM database:

- The block or CI size
- The bytes limit
- The number of blocks or CIs in the root addressable area
- The number of RAPs per block or CI

The following discussion assumes that the values of these parameters are determined in the sequence listed here. After loading or reloading the database with the randomizing parameters determined by the first-guess method, the database administrator can obtain the HDAM Root Addressable Area Statistics report and the Distribution of Database Record Sizes report in order to verify the efficiency of the first-guess randomizing parameters. Then, if necessary, the randomizing parameters could be adjusted with an iterative, experimental "try and see" process (that is, change the value of a randomizing parameter, reload the database, and observe the effect of the changed parameter in the HDAM Root Addressable Area Statistics report and the Distribution of Database Record Sizes report).

Determining the block or CI size

There are several factors for determining the block size or the CI size.

- A recommended general rule of thumb is a 4 KB block or CI size. However, if the average database record length reported in the Distribution of Database Record Sizes report is larger than 800 bytes, then a larger block or CI size could be tried.
- If the average database record length is in the range of 800-1600 bytes, an 8 KB block or CI size is often a better bet.
- If the average database record length is in the range of 1600-2400 bytes, a 12 KB block or CI size should be investigated.
- For average database record lengths substantially longer than 2400 bytes, the first-guess method described in this topic often does not apply.

Block or CI sizes larger than 12 KB should be an exception.

Block or CI sizes smaller than 4 KB are seldom reasonable.

Determining the bytes limit

To determine the bytes limit, you must check the distribution of the database length records in the Distribution of Database Record Sizes report.

As explained in [“Is the HDAM bytes limit appropriate for the database record lengths in this database?”](#) on page 54, the database administrator needs to perform trade-offs when specifying a bytes limit.

To define a reasonable bytes limit, the database administrator should check the Distribution of Database Record Sizes report. The table with the distribution of the database length records should be used in order to determine the bytes limit in accordance with the following rules of thumb:

- The bytes limit is large enough to allow for a high percentage of database records to have all their database segments stored in the root addressable area.
- The bytes limit is small enough to prevent the cascading effect described in [“Is the HDAM bytes limit appropriate for the database record lengths in this database?”](#) on page 54. If the average database record length is around 1/5th of the block or CI size, then (as a first guess, which could be revised by experiments by the database administrator) the bytes limit should not be larger than twice the database record length. For smaller average database record lengths, the ratio between bytes limit and average database record length can be larger than 2. For larger average database record lengths, the ratio between bytes limit and average database record length should be smaller than 2.
- The bytes limit should not be larger than the block or CI size.

Determining the number of blocks or CIs in the root addressable area

In order to determine the number of blocks or CIs in the root addressable area, the database administrator must have previously determined the block or CI size and the bytes limit.

These tasks are described in [“Determining the block or CI size”](#) on page 57 and [“Determining the bytes limit”](#) on page 58).

Then the database administrator should:

1. Select a target packing density for the root addressable area. Is the size of the root addressable area appropriate as described in [“Is the size of the Root Addressable Area appropriate?”](#) on page 52. A good bet would be a packing density of 75 for databases with an average database record length smaller than one tenth of the block or the CI size—or 70 for databases with a larger average database record length.
2. Estimate the total length (expressed in bytes) of all database segments that should be stored in the root addressable area. You can estimate the length by using:
 - a. The selected bytes limit
 - b. The distribution of the database record lengths reported by the Distribution of Database Record Sizes report

Example:

a.

The HDAM Root Addressable Area Statistics report shows the bytes limit specified in the randomizing parameters. Use the Distribution of Database Record Sizes report to see how much (in percentage) database records can fit on the specified bytes limit.

3. If you want to include all segments in the RAA, you can obtain the size of the RAA (in bytes) from the following formula:

$$\text{RAA size (in bytes)} = \frac{\text{AVERAGE DATABASE RECORD SIZE} \times \text{NUMBER OF ROOT SEGMENTS}}{\text{target packing density}}$$

(total length of database segments in the root addressable area)

Where:

AVERAGE DATABASE RECORD SIZE is a field in the Distribution of Database Record Sizes report and NUMBER OF ROOT SEGMENTS is a field in the HDAM Root Addressable Area Statistics report.

If you do not want to include all segments in the RAA, you can obtain the size of the RAA (in bytes) from the following formula:

$$\text{RAA size (in bytes)} = \frac{\text{Sum of ASLEN} \times \text{NUMBER OF ROOT SEGMENTS}}{\text{target packing density}}$$

(total length of database segments in the root addressable area)

Where:

Sum of ASLEN is the sum of the products of AVG SEGM LENGTH and AVG OCCURRENCES PER RECORD of the segments in the Segment Statistics report, of the segments that you want to include in the RAA and NUMBER OF ROOT SEGMENTS is a field in the HDAM Root Addressable Area Statistics report.

4. To obtain the number of blocks or CIs in the root addressable area, divide the size (in bytes) of the root addressable area by the block or CI size.

Determining the number of RAPs per block or CI

This topic discusses the factors to determine the number of RAPs per block or CI.

As a rule of thumb, the total number of RAPs should be around 1.2 times the number of database records. When in doubt, it is better to specify too many rather than too few RAPs. For details, read *IMS Database Administration*. Because the number of database records is printed in NUMBER OF ROOT SEGMENTS in the HDAM Root Addressable Area Statistics report, the database administrator can multiply this number by 1.2 in order to obtain the desired total number of RAPs. Then, in order to obtain the number of RAPs per block or CI, the total number of RAPs must be divided by the number of blocks or CIs in the root addressable area (and rounded up to the next integer).

Tuning HDAM database that have long database records

For HDAM databases with a high percentage of very long database records (for example, 8 KB or more), it is difficult to achieve a low value for the average number of I/O per database record.

Whether the database record length is high can be determined by looking at the Distribution of Database Record Sizes report; this provides the average database record length and a table with the distribution of the database record length. Also, you can see which database records are long by looking at the Longest Database Records report. Some of the actions that can be taken, in order to limit the performance problems of long database records, are:

- Define the bytes limit in such a way that:
 - The root segment and a reasonable amount of dependent segments can be stored in the same block of the root addressable area as the root segment.
 - The remaining dependent segments are stored in the overflow area.

A practice sometimes used is to load the most often referred-to database segments in the root addressable area at database initial load or reload time; the less frequently referred-to database segments are then inserted after the initial load/reload. Sometimes less frequently accessed segment types can be stored in a separate data set group.

Bytes limits larger than the block or CI size should not be used if the database administrator intends to store database ROOTS in most of the blocks or CIs of the root addressable area. The bytes limit of the current database is shown in the top part of the HDAM Root Addressable Area Statistics report, and for an HDAM database, also in the DBD SUMMARY report.

- Use a large block or CI size (for example, 12 KB).
- Evaluate usage of database segment compression in order to reduce the average size of database records.

The following discusses the points to be careful of when using compressed segments:

Using a compression exit routine in order to store the segments on DASD in a compressed format can be useful, especially if the compression can significantly reduce the average database record size in such a way that the average number of database I/Os required to read at random all database segments of

a database record can be lowered. When using compression routines, beware of segment splits created by replace calls that extend the compressed size of a database segment. Some protection against such replace calls can be achieved by avoiding high packing densities in the root addressable area and by defining enough free space for non-primary data set groups.

- Evaluate changes to the database design in order to reduce the average size of database records.
- Evaluate usage of multiple data set groups.

Discussion of the example

You can check the average database record size in AVERAGE DATABASE RECORD SIZE in the Distribution of Database Record Sizes report. In the following figure, AVERAGE DATABASE RECORD SIZE shows 27,999 bytes. Because this is long, the method previously described might be effective to improve performance.

RANGE		QUANTITY	PCT	CUM PCT
-	1000	4	0.2%	0.2%
1001	- 2000	0	0.0%	0.2%
2001	- 3000	19	1.0%	1.2%
3001	- 4000	0	0.0%	1.2%
4001	- 5000	0	0.0%	1.2%
5001	- 6000	0	0.0%	1.2%
6001	- 7000	0	0.0%	1.2%
7001	- 8000	0	0.0%	1.2%
8001	- 9000	0	0.0%	1.2%
9001	- 10000	0	0.0%	1.2%
10001	- 20000	405	20.3%	21.4%
20001	- 30000	0	0.0%	21.4%
30001	- 40000	1,572	78.6%	100.0%
40001	-	0	0.0%	100.0%
AVERAGE DATABASE RECORD SIZE =			27,999	
SHORTEST DATABASE RECORD SIZE =			678	
LONGEST DATABASE RECORD SIZE =			31,498	

Figure 16. Database tuning of the primary data set group in an HDAM database: sample 11

You can check the value of the bytes limit (MAX BYTES) that is specified in the top part of the HDAM Root Addressable Area Statistics report, and for an HDAM, in the DBD Summary report.

In [Figure 6 on page 49](#), the bytes limit is not specified. If there are dependent segments that are referenced at low frequency, specifying the bytes limit is effective for reducing the I/O response time of the database. You should, however, make appropriate specifications considering factors such as the use of the database and the system environment.

In [Figure 11 on page 52](#), you can check the block size or the CI size, and the number of blocks and the number of CIs of the current database in the Data Set Statistics report. In [Figure 11 on page 52](#), BLOCK SIZE shows 8,192 bytes. This is too short for AVERAGE DATABASE RECORD SIZE 27,999 bytes in [Figure 16 on page 60](#). Determine which is better: to decrease the size of the database records that fit in the block or the CI—that is, to compress or to group them into multiple data set groups, and so on—or to increase the size of the block or the CI size.

In [Figure 15 on page 56](#), if you are to use multiple data set groups, you can calculate the average record length of the database records from the root segment to the segment type that you want to include in the primary data set by using the products of AVG SEGM LENGTH and AVG OCCURRENCES PER RECORD of the Segment Statistics report.

You can check the status of segment compression in the Variable Length Segment Statistics section of the Segment Statistics report. The following figure shows that COMPRESSION FACTOR is -8.3%, which means the segment has become longer after the compression.

<<< VARIABLE LENGTH SEGMENT STATISTICS >>>

SEGMENT	<-----LENGTHS----->				<-----SEGMENT COUNT----->		<--COMPRESSION-->	
	MIN	MAX	PFX	DATA	<MIN	=>MIN	FACTOR	NAME
DEP1LEV2	20	120	6	96	0	24,010		
DEP2LEV2	50	200	10	139	0	6,000		
DEP3LEV3	4	40	6	33	0	15,747	- 8.3%	DFSCMPX0

Figure 17. Database tuning of the primary data set group in an HDAM database: sample 12

Chapter 6. Analyzing and resolving fragmentation and optimization exceptions for HIDAM and PHIDAM databases

If you receive fragmentation and optimization exceptions for HIDAM or PHIDAM databases, analyze the state of the database and resolve the exceptions.

The following topics explain how to analyze the database states based on the exceptions that you received and how to recover from the exceptional states.

Notes:

- If you receive exceptions for the primary index of HIDAM database, follow the instructions in [Chapter 8, “Analyzing and resolving fragmentation and optimization exceptions for index databases,” on page 87](#).
- If you receive exceptions for the primary index of PHIDAM database, follow the instructions in the following topics.

Topics:

- [“Analyzing fragmentation exceptions for HIDAM and PHIDAM databases” on page 63](#)
- [“Analyzing optimization exceptions for HIDAM and PHIDAM databases” on page 67](#)
- [“Resolving space use exceptions for HIDAM and PHIDAM databases” on page 69](#)
- [“Resolving fragmentation exceptions for HIDAM and PHIDAM databases” on page 71](#)
- [“Resolving optimization exceptions for HIDAM and PHIDAM databases” on page 71](#)
- [“Resolving space use exceptions for PHIDAM primary indexes” on page 74](#)
- [“Resolving fragmentation exceptions for PHIDAM primary indexes” on page 75](#)

Analyzing fragmentation exceptions for HIDAM and PHIDAM databases

Analyze the fragmentation exceptions that were detected for HIDAM and PHIDAM databases.

Before you begin

Ensure that you can access the Sensor Data Statistics report that contains the latest sensor data. If you cannot access this report, run the FF DB Sensor Printing utility to generate the report.

About this task

In the following procedure, you will analyze each fragmentation exception that you received for HIDAM databases or PHIDAM partitions. For a PHIDAM partition, you might also receive exceptions that were detected in its primary index.

For each exception that was reported, you will first identify the relevant data elements to understand the condition of the database. Then you will analyze the relevant IMS Tools reports to learn more about the database state. Finally, if you use IMS Administration Foundation or Management Console and if relevant charts are available, you will analyze historical trends of sensor data to understand the trend characteristics of the sensor data that was obtained from the database data sets.

By completing these steps, you will identify the type of fragmentation symptoms that occurred and the database data set for which the exceptions were reported.

Procedure

In this section, instructions to analyze the common fragmentation exceptions are provided. Follow the instructions for the exceptions that you received. The most common exceptions are listed first.

Exception class name: **EXCESSIVE_SEGMENT_SCATTERING**

This exception indicates that too many physical pointers point to another block or CI.

When a parent segment and its dependent segment exist in the same block or CI, or when twin segments are in the same block or CI, extra I/O operations are not needed. However, if they are in different blocks or CIs, I/O operations might increase.

Complete these steps to analyze this exception:

- a. Locate the following data elements in the Sensor Data Statistics report:

DB_PCT_NUM_PTR_DIFF_BLK

DB_NUM_PTR_DIFF_BLK

These data elements indicate the number of physical pointers that point to another block or CI, and the percentage of physical pointers that point to another block or CI compared to the physical pointers.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

- b. Optional: To obtain more information about the physical pointers, run the HD Pointer Checker utility of IMS HP Pointer Checker, or run IMS HP Image Copy with the HDPC=YES option, and generate the following reports:

DB Record Distribution Statistics report

By reviewing the summary of dependent segments distribution part and the distribution of dependent segments in root block part, you can determine how many dependent segments exist in the blocks that are different from the root segment block.

Partition Statistics report

Database Statistics report

By reviewing the segment and pointer statistics part, the rate of segment I/O occurrence part, and the VL segment split statistics part, you can determine the number of pointers that point to different blocks for each segment.

- c. Optional: To view the trend of data over time, locate the following chart in IMS Administration Foundation or Management Console:

- Pointers to other blocks or CIs chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Step [c](#), you also identified the chart that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

Exception class name: **EXCESSIVE_VL_SPLIT_SEGMENTS**

This exception indicates that too many variable-length segment splits exist.

When a variable-length segment is replaced with a segment that has a longer length, and if the amount of free space in the block or CI is inadequate, the segment is split into the prefix part and the data part. These segment splits can increase I/O operations.

Complete these steps to analyze this exception:

- a. Locate the following data elements in the Sensor Data Statistics report:

DB_NUM_VLSEG_SPLIT
DB_PCT_NUM_VLSEG_SPLIT

The values for these data elements indicate the number of segment splits, and the percentage of segment splits compared to the number of variable segments.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

- b. Optional: To obtain more information about the variable-length segment splits, run the HD Pointer Checker utility of IMS HP Pointer Checker, or run IMS HP Image Copy with the HDPC=YES option, and generate the HD Data Set Statistics report. In this report, review the Segment occurrences/ split part.
- c. Optional: To view the trend of data over time, locate the following chart in IMS Administration Foundation or Management Console:
 - Variable-length segment splits chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Step c, you also identified the chart that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

Exception class name: FRAGMENTED_FREE_SPACE

Exception class name: EXCESSIVE_SLACK_BYTES

These exceptions indicate that an excessive amount of unusable space exists in the data set. An excessive amount of unusable space wastes space and causes segments and database records to be fragmented.

FRAGMENTED_FREE_SPACE

This exception indicates an excessive amount of *fragmented unusable IMS free space*. Fragmented unusable IMS free space is fragmented IMS free space that is not large enough to store the shortest segment that is defined in the DBD definition.

EXCESSIVE_SLACK_BYTES

This exception indicates an excessive number of *slack bytes*. A slack byte is unusable space that is not used by a segment or that cannot be counted as IMS free space.

To learn more about slack bytes, see the topic "How IMS reclaims space" in the *IMS High Performance Pointer Checker User's Guide*.

These two exceptions indicate the same state in terms of space that cannot be reused by IMS. Even if you receive only one of these exceptions, analyze the state of the data sets from both perspectives to gain an overall understanding of the space that cannot be reused by IMS.

Complete these steps to analyze these exceptions:

- a. Locate the data elements that are related to IMS free space in the Sensor Data Statistics report:

DB_AVG_NUM_FSE

The value of this data element indicates the average number of IMS free space fragments in a CI or block.

DB_AVG_NUM_NOREUSE_FSE

The value of this data element indicates the average number of unusable IMS free space fragments in a CI or block. Unusable IMS free space refers to free space that is too small to store the shortest segment that is defined in the DBD definition. If the value of this data element is large, the database contains an excessive amount of unusable space.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

Also, locate the following data elements. These data elements provide more information about the amount of remaining free space that IMS can use.

DB_NUM_FSE_MIN

The value of this data element indicates the number of IMS free space fragments in which the shortest segment that is defined in the DBD definition can be stored.

DB_NUM_FSE_MAX

The value of this data element indicates the number of IMS free space fragments in which the longest segment that is defined in the DBD definition can be stored.

DB_NUM_FSE

The value of this data element indicates the number of IMS free space fragments.

DB_BYTES_FREE_SPACE

The value of this data element indicates the total bytes of free space.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

Also, in some cases, determining if the length of fragmented IMS free space is large enough to store the segment of the segment type that has the most occurrences can help you analyze the condition of the IMS free space. You can do so by comparing the average length of IMS free space fragments with the segment length of the segment type that has the most occurrences. These values can be obtained as follows:

- i) Calculate the average length of IMS free space fragments by dividing the DB_BYTES_FREE_SPACE value by the DB_NUM_FSE value.
- ii) Determine the segment type that has the most occurrences in the data set by generating an IMS HP Pointer Checker report as explained in Step [c](#).

If the segment length of that segment type is shorter than the average length, you can assume that an excessive amount of unusable IMS free space exists.

- b. Locate the data elements that are related to slack bytes in the Sensor Data Statistics report:

DB_NUM_UNIDENTIFIED

The value of this data element indicates the number of slack bytes.

DB_BYTES_UNIDENTIFIED

The value of this data element indicates the total slack bytes.

DB_AVG_NUM_UNIDENTIFIED

The value of this data element indicates the average number of slack bytes in a CI or block.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

- c. Optional: To obtain more information, run the HD Pointer Checker utility of IMS HP Pointer Checker, or run IMS HP Image Copy with the HDPC=YES option, and generate the HD Data Set Statistics report. In this report, review the Database record statistics part.

In the Database record statistics part, determine the segment type that has the most segment occurrences and the segment length for that segment type. Then compare the length of the segment with the average length of IMS free space fragments that you already calculated in Step [a](#). In an ideal data set, the average length of IMS free space fragments is typically longer than the segment length.

- d. Optional: To view the trend of data over time, locate the following charts in IMS Administration Foundation or Management Console:

- Unusable IMS free space chart
- Usable IMS free space chart
- Slack bytes areas chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Step [d](#), you also identified the charts that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

Exception class name: EXCESSIVE_INDEX_CI_OR_CA_SPLITS

This exception applies only to primary indexes of PHIDAM database.

This exception indicates that the number of control interval (CI) splits, control area (CA) splits, or both are high, and that the VSAM KSDS file is fragmented.

An excessive number of CI splits and CA splits might affect the I/O performance efficiency of applications that perform a massive amount of sequential reading. For information about the effects of an excessive number of CI splits or CA splits, see the topic "Optimizing Free Space Distribution" in *z/OS DFSMS Using Data Sets*.

Locate the following data elements in the Sensor Data Statistics report:

DBX_NUM_CI_SPLIT

DBX_PCT_NUM_CI_SPLIT

The values for these data elements indicate the number of split CIs, and the percentage of split CIs compared to the total number of CIs.

DBX_NUM_CA_SPLIT

DBX_PCT_NUM_CA_SPLIT

The values for these data elements indicate the number of split CAs, and the percentage of split CAs compared to the total number of CAs.

Analyze these values to determine why you received this exception.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

What to do next

If you received other exceptions that belong to other exception categories, see the following topics to determine your next step:

- [“Exception categories for full-function databases” on page 18](#)
- [“Scenario for using the analysis and resolution topics” on page 19](#)

If the only exceptions that you received are fragmentation exceptions, continue with [“Resolving fragmentation exceptions for HIDAM and PHIDAM databases” on page 71](#).

Analyzing optimization exceptions for HIDAM and PHIDAM databases

Analyze the optimization exceptions that were detected for HIDAM and PHIDAM databases.

Before you begin

Ensure that you can access the Sensor Data Statistics report that contains the latest sensor data. If you cannot access this report, run the FF DB Sensor Printing utility to generate the report.

About this task

In the following procedure, you will analyze each optimization exception that you received for HIDAM databases or PHIDAM partitions.

For each exception that was reported, you will first identify the relevant data elements to understand the condition of the database. Then you will analyze the relevant IMS Tools reports to learn more about the database state. Finally, if you use IMS Administration Foundation or Management Console and if relevant charts are available, you will analyze historical trends of sensor data to understand the trend characteristics of the sensor data that was obtained from the database or partition.

By completing these steps, you will determine the possible causes of the optimization exception. The causes can be the block or CI size not matching the average database record length, the existence of long database records, or both.

Procedure

Follow the instruction to analyze the AVERAGE_DB_RECORD_LENGTH exception.

Exception class name: AVERAGE_DB_RECORD_LENGTH

This exception indicates that the average length of database records has exceeded the threshold.

This exception does not always indicate a problem in the database. When the database record length is long, the segments that belong to a same database record might be spread across multiple CIs or blocks, and might be causing extra I/O operations.

You can identify whether extra I/O operations are required by reviewing the DB_ESTIMATED_DBREC_IO data element value. When the value is small, you do not need to analyze this exception. The smallest value, which typically is the optimal value, is the number of data sets. If this value is large, you can improve the condition by reorganizing the database; however, you might be able to improve the condition even further by performing optimization tuning (that is, determining optimal DBD definitions and applying the changes).

The method that you use to analyze the database depends on the number of database data sets:

Single data set

When a database has one data set, all segments reside in that data set. In the most optimized condition, all the segments that belong to a same database record reside in the same CI or block. However, when fragmented IMS free space is shorter than the database record, which means that all segments that belong to the database record cannot be stored in a single CI or block, IMS places some of the segments in other CIs or blocks. This state can cause extra I/O operations.

To determine if extra I/O operations are occurring, locate the following data elements in the Sensor Data Statistics report:

DB_AVG_DBREC_LENGTH

The value for this data element indicates the average length of database records.

DB_NUM_ROOT

The value for this data element indicates the number of root segments (that is, the number of database records).

DB_BLOCK_SIZE

The value for this data element indicates the CI or block size.

DB_ESTIMATED_DBREC_IO

The value for this data element indicates the estimated number of I/O operations that are required to read a database record (that is, the average number of reading different CIs or blocks to read the root segment and all dependent segments by tracking the hierarchical path).

Note: The value is an estimated value; the actual I/O count might be different.

To obtain more information about the segment distribution, run the HD Pointer Checker utility of IMS HP Pointer Checker, or IMS HP Image Copy with the HDPC=YES option, and generate the DB Record Distribution Statistics report.

By reviewing the Distribution of dependent segments by segment code part, you can determine, for each segment type, the number of segments that are not in the block or CI where its root segment is stored, and the distribution of these segments across different CIs or blocks.

The information in this report can be a good indicator for tuning the CI or block size to prevent database records being spread across different CIs or blocks.

Multiple data sets

When a database has multiple data sets, the segments that belong to a same database record can be stored across multiple data sets. Therefore, reviewing the average database record length will not help you identify the data sets in which the segments that increased the database record length reside. To identify the data set that has the most occurrences of scattered segments, follow the analysis steps for the [EXCESSIVE_SEGMENT_SCATTERING](#) exception (a fragmentation exception).

If you identify excessive segment scattering in data set group 1 (DSG1), obtain more information about the segment distribution by running the HD Pointer Checker utility of IMS HP Pointer Checker, or IMS HP Image Copy with the HDPC=YES option, and generate the DB Record Distribution Statistics report.

By reviewing the Distribution of dependent segments by segment code part, you can determine, for each segment type, the number of segments that are not in the block or CI where its root segment is stored, and the distribution of these segments across different CIs or blocks within DSG1.

Together with the CI or block size, this information can be a good indicator for tuning the CI or block size to prevent segments being spread across different CIs or blocks.

If excessive segment scattering was not observed in DSG1, you do not need to perform optimization tuning. Reorganizing the database will resolve the exception.

The average database record length can be a trigger to start database analysis. In the following charts of IMS Administration Foundation or Management Console, you can review the trend of the average database record length and the number of database records:

- Average database record length chart
- Number of database records chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

You also identified the charts that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

What to do next

If you received other exceptions that belong to other exception categories, see the following topics to determine your next step:

- [“Exception categories for full-function databases” on page 18](#)
- [“Scenario for using the analysis and resolution topics” on page 19](#)

If the only exceptions that you received are optimization exceptions, continue with [“Resolving optimization exceptions for HIDAM and PHIDAM databases” on page 71](#).

Resolving space use exceptions for HIDAM and PHIDAM databases

Resolve the space use exceptions that you received for your HIDAM and PHIDAM databases.

Before you begin

Ensure that you have completed the analysis steps for HIDAM and PHIDAM exceptions and that you have identified the possible causes of the database exceptions. If you have not yet analyzed the space use

exceptions, see Chapter 4, “Analyzing space use exceptions for all full-function database types,” on page 29.

About this task

The instructions in this topic cannot be applied to resolve exceptions of primary indexes of PHIDAM databases. To resolve exceptions for primary indexes of PHIDAM databases, see “Resolving space use exceptions for PHIDAM primary indexes” on page 74.

Procedure

If you receive space use exceptions, and if you identify an out-of-space condition in database data sets as a result of analyzing the space use exceptions, consider resolving the exception condition by taking the following actions.

- If you also receive fragmentation exceptions, and if you identify excessive fragmentation of IMS free space or the existence of many unusable IMS free space fragments, consider reorganizing the database. Reorganization will eliminate the fragmentation and increase usable IMS free space in the data sets; subsequently, new segments can be added to the database.
- After you analyze the exceptions that you received, if you determine that reorganizing the database will not create sufficient IMS free space, and if you have not applied a Segment Edit/Compression exit routine, consider applying the routine. For information about applying the routine, see the topic "Segment Edit/Compression exit routines" in *IMS Database Administration*.

If the out-of-space conditions are not resolved by these actions, consider changing the database structure or increasing the DASD resources. The method you apply depends on the results of the analysis steps as follows:

Tip: Depending on the status of the data set, applying several of the following methods can be an effective approach to resolving the exceptions. Before making any changes to your environment, evaluate all the methods in the following list and determine the best approach to resolve the condition.

- If the out-of-space condition is caused by a *primary-space-used-up* condition (that is, the data set has only primary space, and the size of used space is reaching the size of the primary allocated space), and if the data sets allocation size is smaller than the maximum size (4 GB or 8 GB), increase the size of the allocated space. For instructions for increasing the size of allocated space, see the topic "Changing the amount of space allocated" in *IMS Database Administration*.
- If the out-of-space condition is caused by a *data-set-full* condition (that is, the size of the data set approaching its maximum size of 4 GB or 8 GB), typically, changing the DBD definition and then restructuring the database, migrating the database, or both will resolve the condition.

Consider the following methods to resolve a data-set-full condition:

- If the database data set is HIDAM and VSAM, consider changing the data set type to OSAM. By changing the data set type to OSAM, you can increase the maximum size of the data set from 4 GB to 8 GB.
- Split the database into several smaller databases.
- Increase the number of database data sets. The maximum number of data sets is 10. When you increase the number of data sets, be aware that I/O counts for accessing database records also increase.
- If the database is an HIDAM database, migrate the database from HDAM to PHDAM.
- If the database is PHIDAM, increase the number of partitions.

For more information about these tasks, see the following topics in *IMS Database Administration*:

- "Changing operating system access methods"
- "Using OSAM as the access method"
- "Multiple data set groups"
- "Changing the number of data set groups"

- "Converting database types"
- "Adding partitions to an existing HALDB database"
- If the out-of-space condition is caused by an *extent-resources-used-up* condition (that is, when the data set cannot be expanded), typically, changing the data set allocation definitions and reorganizing the database will resolve the condition.

Consider the following methods to resolve an extent-resources-used-up condition:

- If the data set has reached the maximum number of data set extent counts, and if the DASD volumes have enough space, increase the primary and the secondary extent sizes for data set allocation.
- If the DASD volume does not have enough space, or if the data set has reached the maximum number of extents that is defined for that volume, increase the number of DASD volumes.

Generally, as the number of extents or the number of DASD volumes increases, I/O operations also increase. Therefore, consider using a DASD volume that has sufficient size, and specify an extent size that is large enough for data set allocation.

Resolving fragmentation exceptions for HIDAM and PHIDAM databases

Resolve the fragmentation exceptions that you received for your HIDAM and PHIDAM databases.

Before you begin

Ensure that you have completed the analysis steps for HIDAM and PHIDAM exceptions and that you have identified the possible causes of the database exceptions. If you have not yet analyzed the fragmentation exceptions, see [“Analyzing fragmentation exceptions for HIDAM and PHIDAM databases”](#) on page 63.

About this task

The instructions in this topic cannot be applied to resolve exceptions of primary indexes of PHIDAM databases. To resolve exceptions for primary indexes of PHIDAM databases, see [“Resolving fragmentation exceptions for PHIDAM primary indexes”](#) on page 75.

Procedure

If the only exceptions that you received are fragmentation exceptions, reorganize the database. After you reorganize the database, rerun a policy evaluation to ensure that the exceptions are resolved.

After the reorganization, the EXCESSIVE_SEGMENT_SCATTERING exception might remain, or the values of the data elements that are related to this exception might continue to be high (for example, if this exception was reported as a critical level exception, the exception might be reported as a severe or warning level exception after the reorganization.) If you receive this exception again, follow the steps for resolving optimization exceptions.

Resolving optimization exceptions for HIDAM and PHIDAM databases

Resolve the optimization exceptions that you received for your HIDAM and PHIDAM databases.

Before you begin

Ensure that you have completed the analysis steps for HIDAM and PHIDAM exceptions and that you have identified the possible causes of the database exceptions. If you have not yet analyzed the optimization exceptions, see [“Analyzing optimization exceptions for HIDAM and PHIDAM databases”](#) on page 67.

Procedure

If you receive optimization exceptions, you must collect more information about the database data sets to evaluate the requirements for the database definition and other factors that are described in the following topics.

If you determine that you must change the DBD definition, plan a database reorganization to apply the changes.

To collect more information and to evaluate the requirements for your HIDAM and PHIDAM databases, see the following topics:

- [“Average number of I/O operations per database record” on page 72](#)
- [“Is there enough free space defined in DBDGEN?” on page 72](#)
- [“Is the block size or CI size appropriate?” on page 73](#)
- [“Is the problem with long database records solved appropriately?” on page 73](#)

Average number of I/O operations per database record

A database is considered to be efficiently organized if the average number of I/O operations that are required to randomly read all database segments of one database record is low. The average number of I/Os required to read at random all database segments of one database record is one of the most important indicators of the quality of the physical organization of the HIDAM database.

The average number of I/O operations is reported in the DB Tuning Statistics report. For information about generating this report, see [“Information resources for analyzing exceptions” on page 25](#). Note that the HSSR Engine ignores I/Os that are required to access the index of the HIDAM database record. By looking at this average number, the database administrator can determine if the database records and their segment occurrences are stored efficiently without analyzing the entire report.

For an ideal database that consists of a single data set group, this average number is 1.0. However, in a real production database, this ideal value is typically difficult to achieve and the average number of database I/Os per database record is higher.

As a general guideline, the database is considered to be fairly well organized if the average number of I/Os per database record meets the following criteria:

- Is below 1.10 for databases with an average database record length below one tenth of the block or CI size
- Is below 1.20 for databases with larger database record lengths

Numbers above 1.10 or 1.20 might indicate that a database is poorly organized. In this case, the database administrator should ask the following questions to determine the reason for a poor organization:

1. Is the amount of free space that was specified during DBDGEN sufficient? See [“Is there enough free space defined in DBDGEN?” on page 72](#).
2. Is the block size or CI size appropriate for the database record lengths of this database? See [“Is the block size or CI size appropriate?” on page 73](#).
3. Is the database record length excessive? See [“Is the problem with long database records solved appropriately?” on page 73](#).

Is there enough free space defined in DBDGEN?

Based on an algorithm of IMS, you need to define free space according to the database uses.

To check the current free space when defining free space, see the Dataset Statistics report. For the detail of the algorithm of IMS, see *IMS Database Administration*.

Discussion of the example

You can check the information about the definition of free space in the Dataset Statistics report of the IPR Unload utility and the IPR Reload utility. [Figure 11 on page 52](#) shows that the value of FSPF is 0, which means that no free space is defined. You need to specify necessary free space taking into account the database uses based on the IMS algorithm.

The following figure is an example of the Distribution of Free Space section of the Dataset Statistics report produced by the IPR Reload utility. AVG SIZE OF FREE SPACE in the Distribution of Free Space section is printed only in the IPR Reload utility, and it shows the average free space of each data set group. This value can be used to check whether there is sufficient free space in the database after the database is reloaded.

<<< DISTRIBUTION OF FREE SPACE >>>

DSG	FREE SPACE RANGE	NUMBER OF BLOCKS	PCT OF TOTAL BLKS	<----CUMULATIVE---->		AVG SIZE OF FREE SPACE
				NUMBER OF BLOCKS	PCT OF TOTAL BLKS	
1	322 - 512	0	0.0%	0	0.0%	
1	310 - 321	0	0.0%	0	0.0%	
1	0 - 309	21	100.0%	21	100.0%	
1					114.3
2	314 - 512	0	0.0%	0	0.0%	
2	310 - 313	0	0.0%	0	0.0%	
2	0 - 309	21	100.0%	21	100.0%	
2					186.5
3	318 - 512	0	0.0%	0	0.0%	
3	306 - 317	0	0.0%	0	0.0%	
3	0 - 305	32	100.0%	32	100.0%	
3					190.2

Figure 18. Tuning a HIDAM database: sample 13

Is the block size or CI size appropriate?

As a rule of thumb, use a 4 KB CI/block size for the data portion of a HIDAM. However, if the average database record length is more than 1000 bytes (that is, larger than 1/4th of the block or CI size), then an increase of the block or CI size from 4 KB to 8 KB might improve the performance of HIDAM database accesses.

In these cases, an increase of the CI size/block size from 4 KB to 8 KB can often reduce the number of I/O operations (but will increase slightly the time of an I/O operation). CI/block sizes larger than 8 KB should be an exception (for example, for very large average database record sizes).

Discussion of the example

You can check the current block size or CI size, the number of blocks or the number of CIs in the Data Set Statistics report. You can check the average database record size in AVERAGE DATABASE RECORD SIZE in the Distribution of Database Record Sizes report. In [Figure 12 on page 52](#), AVERAGE DATABASE RECORD SIZE shows 2,317 bytes. Because this is over 1000 bytes, you can expect performance to improve by increasing the block size from 4 KB to 8 KB.

Is the problem with long database records solved appropriately?

For HIDAM databases with a high percentage of very long database records (for example, 32 KB or more), it is difficult to achieve a low value for the average number of I/Os per database record.

Some of the actions that can be taken to limit the performance problems of long database records are:

- Use a large block or CI size (for example, 12 KB)
- Evaluate usage of database segment compression in order to reduce the average size of database records

- The following discusses the points to be careful of when using compressed segments:

Using a compression exit routine in order to store the segments on DASD in a compressed format can be useful, especially if the compression can significantly reduce the average database record size in such a way that the average number of database I/Os required to read at random all database segments of a database record can be lowered. When using compression routines, beware of segment splits created by replace calls that extend the compressed size of a database segment. Some protection against such replace calls can be achieved by defining enough free space for non-primary data set groups.

- Evaluate storage of the most often referred-to database segments in the same block or CI as the root. The less frequently referred-to database segments can be inserted after the initial load/reload or can be stored in another data set group.
- Evaluate changes to the database design in order to reduce the average size of database records.

Discussion of the example

You can check the average size of database records in AVERAGE DATABASE RECORD SIZE of the Distribution of Database Record Sizes report. Figure 16 on page 60 shows that the average size of database records is 27,999, which is too long, and the method previously described is effective.

You can check the status of segment compression in the Variable Length Segment Statistics section of the Segment Statistics report. The following figure shows that COMPRESSION FACTOR is 0.0%, which means the segment is not compressed at all. You should review the compression routine to raise the compression ratio, which might shorten the average length of database records.

<<< VARIABLE LENGTH SEGMENT STATISTICS >>>

SEGMENT	<-----LENGTHS----->				<-----SEGMENT COUNT----->		<--COMPRESSION-->	
	MIN	MAX	PFX	DATA	<MIN	=>MIN	FACTOR	NAME
DEP1LEV2	220	700	6	544	0	10,290		
DEP2LEV2	50	2,500	10	1,454	0	12,000		
DEP3LEV3	4	41	6	31	0	57,600	0.0%	DFSCMPX0

Figure 19. Tuning a HIDAM database: sample 14

You can calculate the average record length of the database records from the root segment to the segment type that you want to include in the primary data set by summing up the values of AVG SEGM LENGTH in the Segment Statistics report. For example, in Figure 15 on page 56, if you want to include segments ROOTLEV1 and DEP1LEV2 in the primary data set group, add the sum of the products of AVG SEGM LENGTH and AVG OCCURRENCES PER RECORD for segments ROOTLEV1 and DEP1LEV2—that is, the value of CONTRIBUTION TO DB RECORD SIZE CUM SIZE, which is 430.

You can check the block size or the CI size of the current database data sets in the Data Set Statistics report. See Figure 11 on page 52.

Resolving space use exceptions for PHIDAM primary indexes

Resolve the space use exceptions that you received for your PHIDAM primary index.

Before you begin

Ensure that you have completed the analysis steps for PHIDAM primary index exceptions and that you have identified the possible causes of the exceptions. If you have not yet analyzed the space use exceptions, see Chapter 4, “Analyzing space use exceptions for all full-function database types,” on page 29.

Procedure

If you receive space use exceptions for the PHIDAM primary index, the method you apply to resolve the exceptions depends on the results of the analysis steps as follows:

Tip: Depending on the status of the data set, applying several of the following methods can be an effective approach to resolving the exceptions. Before making any changes to your environment, evaluate all the methods in the following list and determine the best approach to resolve the condition.

- If the out-of-space condition is caused by a *primary-space-used-up* condition (that is, the data set has only primary space, and the size of used space is reaching the size of the primary allocated space), and if the data sets allocation size is smaller than the maximum size (4 GB), increase the size of the allocated space. For instructions for increasing the size of allocated space, see the topic "Changing the amount of space allocated" in *IMS Database Administration*.
- If the out-of-space condition is caused by a *data-set-full* condition (that is, the size of the data set approaching its maximum size of 4 GB based on the value of the DBX_RBA_HIGH_USED data element), new index segments cannot be created in the primary index. To resolve this condition, decrease the number of root segments in the PHIDAM partition by splitting the partition into smaller partitions.
- If the out-of-space condition is caused by an *extent-resources-used-up* condition (that is, when the data set cannot be expanded), typically, changing the data set allocation definitions and reorganizing the database will resolve the condition.

Consider the following methods to resolve an extent-resources-used-up condition:

- If the data set has reached the maximum number of data set extent counts, and if the DASD volumes have enough space, increase the primary and the secondary extent sizes for data set allocation.
- If the DASD volume does not have enough space, or if the data set has reached the maximum number of extents that is defined for that volume, increase the number of DASD volumes.

Generally, as the number of extents or the number of DASD volumes increases, I/O operations also increase. Therefore, consider using a DASD volume that has sufficient size, and specify an extent size that is large enough for data set allocation.

Resolving fragmentation exceptions for PHIDAM primary indexes

Resolve the fragmentation exceptions that you received for your PHIDAM primary index.

Before you begin

Ensure that you have completed the analysis steps for PHIDAM primary index exceptions and that you have identified the possible causes of the exceptions. If you have not yet analyzed the fragmentation exceptions, see [“Analyzing fragmentation exceptions for HIDAM and PHIDAM databases”](#) on page 63.

Procedure

The only fragmentation exception for PHIDAM primary indexes is the EXCESSIVE_INDEX_CI_OR_CA_SPLITS exception.

If you observe no performance degradation in IMS applications that use the PHIDAM partition, you can ignore this exception.

If you observe performance degradation, you can resolve this exception by removing the CI splits and CA splits by re-creating the VSAM KSDS. However, performance degradation can be caused by the PHIDAM partition rather than the primary index. Therefore, consider analyzing the fragmentation and optimization exceptions for the PHIDAM partition. If you reorganize the PHIDAM partition, CI splits and CA splits in the primary index are also resolved.

Chapter 7. Analyzing and resolving fragmentation and optimization exceptions for HISAM and SHISAM databases

If you receive fragmentation and optimization exceptions for HISAM or SHISAM databases, analyze the state of the database and resolve the exceptions.

The following topics explain how to analyze the database states based on the exceptions that you received and how to recover from the exceptional states.

Topics:

- [“Analyzing fragmentation exceptions for HISAM and SHISAM databases” on page 77](#)
- [“Analyzing optimization exceptions for HISAM and SHISAM databases” on page 80](#)
- [“Resolving space use exceptions for HISAM and SHISAM databases” on page 82](#)
- [“Resolving fragmentation exceptions for HISAM and SHISAM databases” on page 83](#)
- [“Resolving optimization exceptions for HISAM and SHISAM databases” on page 83](#)

Analyzing fragmentation exceptions for HISAM and SHISAM databases

Analyze the fragmentation exceptions that were detected for HISAM and SHISAM databases.

Before you begin

Ensure that you can access the Sensor Data Statistics report that contains the latest sensor data. If you cannot access this report, run the FF DB Sensor Printing utility to generate the report.

About this task

In the following procedure, you will analyze each fragmentation exception that you received for HISAM and SHISAM databases.

For each exception that was reported, you will first identify the relevant data elements to understand the condition of the database. Then, for HISAM databases, you will analyze the relevant IMS Tools reports to learn more about the database state. Finally, if you use IMS Administration Foundation or Management Console and if relevant charts are available, you will analyze historical trends of sensor data to understand the trend characteristics of the sensor data that was obtained from the database data sets.

By completing these steps, you will identify the type of fragmentation symptoms that occurred and the database data set for which the exceptions were reported.

Procedure

In this section, instructions to analyze the common fragmentation exceptions are provided. Follow the instructions for the exceptions that you received. The most common exceptions are listed first.

Exception class name: **EXCESSIVE_CI_OR_CA_SPLITS**

This exception indicates that the number of CI splits, CA splits, or both are high, and that the VSAM KSDS file is fragmented.

The data set of SHISAM databases and the primary data set of HISAM databases are VSAM KSDSs. As data is updated in VSAM KSDS files, control interval (CI) and control area (CA) splits occur. A condition where excessive CI and CA splits exist indicates that the data set is fragmented. CI and CA splits do not occur in overflow data sets because overflow data sets are VSAM ESDSs.

An excessive number of CI splits and CA splits might affect the I/O performance efficiency of applications that perform a massive amount of sequential reading. For information about the effects of an excessive number of CI splits or CA splits, see the topic "Optimizing Free Space Distribution" in *z/OS DFSMS Using Data Sets*.

Complete these steps to analyze this exception:

- a. Locate the following data elements in the Sensor Data Statistics report:

DB_NUM_CI_SPLIT

The value for this data element indicates the number of split CIs.

DB_PCT_NUM_CI_SPLIT

The value for this data element indicates the percentage of split CIs compared to the total number CIs.

DB_NUM_CA_SPLIT

The value for this data element indicates the number of split CAs.

DB_PCT_NUM_CA_SPLIT

The value for this data element indicates the percentage of split CAs compared to the total number of CAs.

- b. Optional: To view the trend of data over time, locate the following charts in IMS Administration Foundation or Management Console:

- CA splits chart
- CI splits chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Step b, you also identified the charts that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

Exception class name: EXCESSIVE_HISAM_DELETE_SEGM

This exception indicates that too many segments whose delete flag is set to ON exist. These segments have already been deleted, but they still exist in the database. These segments waste space and increase the number of I/O operations.

For information about the effects of delete segments, see the topic "Deleting segments" in *IMS Database Administration*.

Complete these steps to analyze this exception:

- a. Locate the following data elements in the Sensor Data Statistics report:

DB_NUM_DELSEG

DB_PCT_NUM_DELSEG

The values for these data elements indicate the number of segments whose delete flag is set to ON and the percentage of these segments compared to the total bytes of used blocks in the data set.

- b. Optional: To obtain more information about these segments, run the HD Pointer Checker utility of IMS HP Pointer Checker, or IMS HP Image Copy with the HDPC=YES option, and generate the HISAM Data Set Statistics report.

By reviewing this report, you can determine how many deleted segments exist for each segment type. You can also identify the length of each segment, which helps you identify the amount of space that is occupied by the deleted segments. In addition, you can identify the segment level for each segment type. If segment types whose segment level is high have many deleted segments, I/O performance can be affected.

- c. Optional: To view the trend of data over time, locate the following chart in IMS Administration Foundation or Management Console:

- Deleted segment chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Step c, you also identified the chart that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

Exception class name: FRAGMENTED_FREE_SPACE

This exception indicates an excessive amount of *fragmented unusable IMS free space*. Fragmented unusable IMS free space is fragmented space that is too small to store the shortest segment that is defined in the DBD definition.

Complete these steps to analyze these exceptions:

- Locate the following data element in the Sensor Data Statistics report:

DB_AVG_NUM_FSE

The value of this data element indicates the average number of IMS free space fragments in a CI.

This data element is reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

Also, locate the following data elements. These data elements help you understand more about the amount of remaining free space that IMS can use.

DB_NUM_FSE_MIN

The value of this data element indicates the number of IMS free space fragments in which the shortest segment that is defined in the DBD definition can be stored.

DB_NUM_FSE_MAX

The value of this data element indicates the number of IMS free space fragments in which the longest segment that is defined in the DBD definition can be stored.

DB_NUM_FSE

The value of this data element indicates the number of IMS free space fragments.

DB_BYTES_FREE_SPACE

The value of this data element indicates the total bytes of free space.

These data elements are reported for each data set. Locate the data element values for each data set, and identify the data sets for which this exception was reported.

- Optional: To view the trend of data over time, locate the following charts in IMS Administration Foundation or Management Console:

- Unusable IMS free space chart
- Usable IMS free space chart

Understanding the trend over time can help you anticipate the future behavior of the data sets. Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Step b, you also identified the charts that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

What to do next

If you received other exceptions that belong to other exception categories, see the following topics to determine your next step:

- [“Exception categories for full-function databases” on page 18](#)
- [“Scenario for using the analysis and resolution topics” on page 19](#)

If the only exceptions that you received are fragmentation exceptions, continue with [“Resolving fragmentation exceptions for HISAM and SHISAM databases” on page 83](#).

Analyzing optimization exceptions for HISAM and SHISAM databases

Analyze the optimization exceptions that were detected for HISAM databases. For SHISAM databases, no optimization exceptions are issued.

Before you begin

Ensure that you can access the Sensor Data Statistics report that contains the latest sensor data. If you cannot access this report, run the FF DB Sensor Printing utility to generate the report.

About this task

In the following procedure, you will analyze each optimization exception that you received for HISAM and SHISAM databases.

For each exception that was reported, you will first identify the relevant data elements to understand the condition of the database. Then you will analyze the relevant IMS Tools reports to learn more about the database state. Finally, if you use IMS Administration Foundation or Management Console and if relevant charts are available, you will analyze historical trends of sensor data to understand the trend characteristics of the sensor data that was obtained from the database.

By completing these steps, you will determine the possible causes of the optimization exceptions. The causes can be the that LRECL size of the primary data set, the CI size of the overflow data set for HISAM, or both do not match the length of database records or segments.

Procedure

Follow the instructions to analyze the AVERAGE_DB_RECORD_LENGTH exception.

Exception class name: AVERAGE_DB_RECORD_LENGTH

This exception indicates that the average database record length has exceeded the threshold value.

The average record length can be used to determine the logical record length of the primary data set of HISAM databases.

A HISAM database is considered to be in its optimal state when root segments and dependent segments that belong to the same database record exist in the same logical record. However, when IMS applications add dependent segments, and if the space in the logical records is insufficient to store dependent segments, the dependent segments will be stored in a logical record in the overflow data set.

If the length of a database record is longer than the logical record length (KSDS LRECL), the database record is divided and stored in the primary data set and the overflow data set, which results in increased I/O operations.

Optimizing HISAM databases typically includes tuning the logical record length and the CI size of the primary database data set to decrease the number of segments in the overflow data set.

Complete these steps to analyze this exception:

- a. Locate the following data elements in the Sensor Data Statistics report:

DB_AVG_DBREC_LENGTH

The value for this data element indicates the average length of database records.

DB_NUM_ROOT

The value for this data element indicates the number of root segments (that is, the number of database records).

DB_NUM_SEG

The value for this data element indicates the number of segments in a data set.

By comparing the numbers of segments in the primary data set and in the overflow data set, you can determine the ratio of segments that exist in the overflow.

DB_BLOCK_SIZE**DB_LRECL_SIZE**

The values for these data elements indicate the CI size and logical record length of data set.

By comparing the CI size and the logical record length of the primary data set with the average database length, you can determine if the CI size and the logical record length of the primary data set are long enough.

DB_ESTIMATED_DBREC_IO

The value for this data element indicates the estimated number of I/O operations that are required to read a database record (that is, the average number of reading CIs to read the root segment and all dependent segments by tracking the hierarchical path). This value is used later in optimization tuning.

Note: The value is an estimated value; the actual I/O count might be different.

When this value is small, most of the segments that belong to the same database record are in the same CI. Even if the average length of database records is long, you do not necessarily need to resolve this exception. If this value is large, consider taking actions to resolve this exception.

- b. Optional: To obtain more information about the segment distribution, run the HD Pointer Checker utility of IMS HP Pointer Checker, or IMS HP Image Copy with the HDPC=YES option, and generate the HISAM Data Set Statistics report.

By analyzing this report, you can determine the number of occurrences for each segment type in the primary and the overflow data sets.

- c. Optional: To view the trend of data over time, locate the following charts in IMS Administration Foundation or Management Console:

- Average database record length chart
- Number of database records chart

Understanding the trend over time can help you anticipate the future behavior of the data sets.

Use this knowledge to determine when you might need to take preventative action and to establish efficient maintenance plans.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

In Step c, you also identified the chart that can help you establish maintenance plans to prevent this exceptional state from occurring in the future.

What to do next

If you received other exceptions that belong to other exception categories, see the following topics to determine your next step:

- [“Exception categories for full-function databases” on page 18](#)
- [“Scenario for using the analysis and resolution topics” on page 19](#)

If the only exceptions that you received are optimization exceptions, continue with [“Resolving optimization exceptions for HISAM and SHISAM databases” on page 83](#).

Resolving space use exceptions for HISAM and SHISAM databases

Resolve the space use exceptions that you received for your HISAM and SHISAM databases.

Before you begin

Ensure that you have completed the analysis steps for HISAM and SHISAM exceptions and that you have identified the possible causes of the database exceptions. If you have not yet analyzed the space use exceptions, see [Chapter 4, “Analyzing space use exceptions for all full-function database types,”](#) on page 29.

Procedure

If you receive space use exceptions, and if you identify an out-of-space condition in database data sets as a result of analyzing the space use exceptions, consider resolving the exception condition by taking the following actions:

- If you also receive fragmentation exceptions, and if you identify excessive fragmentation of IMS free space or the existence of many deleted segments, consider reorganizing the database. Reorganization will resolve the fragmentation and increase usable IMS free space in the data sets; subsequently, new segments can be added to the database.
- After you analyze the exceptions that you received, if you determine that reorganizing the database will not create sufficient IMS free space, and if you have not applied a Segment Edit/Compression exit routine, consider applying the routine. For information about applying the routine, see the topic "Segment Edit/Compression exit routines" in *IMS Database Administration*.

If the out-of-space conditions are not resolved by these actions, consider changing the database structure or increasing the DASD resources. The method you apply depends on the results of the analysis steps as follows:

Tip: Depending on the status of the data set, applying several of the following methods can be an effective approach to resolving the exceptions. Before making any changes to your environment, evaluate all the methods in the following list and determine the best approach to resolve the condition.

- If the out-of-space condition is caused by a *primary-space-used-up* condition (that is, the data set has only primary space, and the size of used space is reaching the size of the primary allocated space), and if the data sets allocation size is smaller than the maximum size (4 GB or 8 GB), increase the size of the allocated space. For instructions for increasing the size of allocated space, see the topic "Changing the amount of space allocated" in *IMS Database Administration*.
- If the out-of-space condition is caused by a *data-set-full* condition (that is, the size of the data set is approaching its maximum size of 4 GB), changing the DBD definition and then restructuring the database will typically resolve the condition.

Consider the following methods to resolve a data-set-full condition:

- If your HISAM database has only a primary data set, add an overflow data set. When you add an overflow data set, be aware that I/O counts for accessing database records also increase.
- Split the database into several smaller databases.
- If the out-of-space condition is caused by an *extent-resources-used-up* condition (that is, when the data set cannot be expanded), typically, changing the data set allocation definitions and reorganizing the database will resolve the condition.

Consider the following methods to resolve an extent-resources-used-up condition:

- If the data set has reached the data set extent limit, and if the DASD volumes have enough space, increase the primary and the secondary extent sizes for data set allocation.
- If the DASD volume does not have enough space, or if the data set has reached the maximum number of extents that is allowed for that volume, increase the number of DASD volumes.

Generally, as the number of extents or the number of DASD volumes increases, I/O operations also increase. Therefore, consider using a DASD volume that has sufficient size, and specify an extent size that is large enough for data set allocation.

Resolving fragmentation exceptions for HISAM and SHISAM databases

Resolve the fragmentation exceptions that you received for your HISAM and SHISAM databases.

Before you begin

Ensure that you have completed the analysis steps for HISAM and SHISAM exceptions and that you have identified the possible causes of the database exceptions. If you have not yet analyzed the fragmentation exceptions, see [“Analyzing fragmentation exceptions for HISAM and SHISAM databases” on page 77](#).

Procedure

If the only the exception that you received for fragmentation is EXCESSIVE_CI_OR_CA_SPLITS, and if you observe no performance degradation in IMS applications that use the database, you can ignore this exception.

If you observe performance degradation, you can resolve this exception by removing the CI splits and CA splits by reorganizing the database.

If you received other exceptions, which can be EXCESSIVE_HISAM_DELETE_SEGM and FRAGMENTED_FREE_SPACE, reorganize the database.

Resolving optimization exceptions for HISAM and SHISAM databases

Resolve the optimization exceptions that you received for your HISAM and SHISAM databases.

Before you begin

Ensure that you have completed the analysis steps for HISAM and SHISAM exceptions and that you have identified the possible causes of the database exceptions. If you have not yet analyzed the optimization exceptions, see [“Analyzing optimization exceptions for HISAM and SHISAM databases” on page 80](#).

Procedure

If you receive optimization exceptions, you must collect more information about the database to evaluate the requirements for the database definition.

If you determine that you must change the DBD, plan a database reorganization to apply the changes.

To collect more information and to evaluate the requirements for your HISAM and SHISAM databases, see the following topics:

- [“Average number of I/O operations per database record” on page 83](#)
- [“Is the length of the KSDS record appropriate?” on page 84](#)
- [“Is the ESDS CI size appropriate for the record length of this database?” on page 85](#)
- [“Does the segment fit on KSDS?” on page 86](#)

Average number of I/O operations per database record

An HISAM database is considered to be efficiently organized if the average number of I/O operations that are required to randomly read all database segments of one database record is low. The average number

of I/Os required to read at random all database segments of one database record is one of the most important indicators for the quality of the physical organization of the HISAM database.

The average number of I/O operations is reported in the DB Tuning Statistics report. For information about generating this report, see [“Information resources for analyzing exceptions”](#) on page 25. Note that the HSSR Engine counts the reading of a KSDS record as a single I/O operation and ignores I/O operations that are required to access the VSAM index CIs. By looking at this average number, the database administrator can determine if the database records and their segment occurrences are stored efficiently without analyzing the entire report.

For an ideal HISAM database, this average number is 1.0. However, in a real production database, this ideal value is typically difficult to achieve and the average number of database I/Os per database record will be higher.

Numbers above 1.30 might indicate that the HISAM database is poorly organized. In this case, database administrator should ask the following questions to determine the reason for the poor organization:

- Is the KSDS record length appropriate? See [“Is the length of the KSDS record appropriate?”](#) on page 84.
- Is the ESDS CI size appropriate for the database record lengths of this DB? See [“Is the ESDS CI size appropriate for the record length of this database?”](#) on page 85.

The guideline for the KSDS record length applies to a SHISAM database and a HISAM database without an overflow data set. For HISAM database that has the overflow data set, see also [“Does the segment fit on KSDS?”](#) on page 86.

Is the length of the KSDS record appropriate?

The database administrator needs to perform trade-offs when determining the KSDS record length for a HISAM database.

Selecting a large KSDS LRECL allows reduction of the number of database records that cannot be stored entirely in the KSDS record. This approach reduces the average number of I/Os per database record. However, selection of a KSDS LRECL that is too large might represent a waste of DASD space if a high percentage of database record does not reasonably fill the KSDS record.

Discussion of the example

You can check the following in the Data Set Statistics report: the block size or the CI size of a database data set that is defined, the number of blocks or the number of CIs that are defined, and the logical record length. See the following figure.

```

IMS HIGH PERFORMANCE UNLOAD                                PAGE: 1
5655-E06                                                    10/01/2021 14.00.18

                                "DATA SET STATISTICS"

TYPE DDNAME      BLOCK RECORD
-----
PRIM DDHS0010    4,096   370
OVFL DDH00010    8,192   500
                                TESTDS.NTFC5106.DBHS0010.DDHS0010
                                TESTDS.NTFC5106.DBHS0010.DDH00010

TYPE DDNAME      SMALL  LARGE
-----
PRIM DDHS0010     22    202
OVFL DDH00010     22    202

```

Figure 20. Tuning a HISAM database: sample 15

You can check the number of logical records in NUMBER OF LOGICAL RECORDS in the HISAM Statistics report. See the following figure.

"HISAM STATISTICS"

PRIMARY DD = DDHS0010

NUMBER OF LOGICAL RECORDS.....	10
NUMBER OF RECORDS WITH OVERFLOW CHAINS.....	4
NUMBER OF RECORDS WITH NO OVERFLOW CHAINS...	6
MAX NUMBER OF OVERFLOW CHAINS.....	2
MIN NUMBER OF OVERFLOW CHAINS.....	1
AVERAGE NUMBER OF OVERFLOW CHAINS.....	1.3

OVERFLOW DD = DDH00010

NUMBER OF LOGICAL RECORDS.....	5
--------------------------------	---

Figure 21. Tuning a HISAM database: sample 16

You can check the average length of the database records and distribution of database record length in the Distribution of Database Record Sizes report.

Is the ESDS CI size appropriate for the record length of this database?

As a rule of thumb, use a 4 KB CI size for the ESDS portion of a HISAM database. However, for long database records, an increase of the ESDS CI size from 4 KB to 8 KB might improve the performance of HISAM database accesses.

Discussion of the example

In the Data set statistics report, you can check the block size or the CI size of the database data set and the number of blocks or the number of CIs that are defined—see [Figure 20 on page 84](#). You can also check the number of logical records in the HISAM Statistics report—see [Figure 21 on page 85](#).

You can check the average size of database records in AVERAGE DATABASE RECORD SIZE of the Distribution of Database Record Sizes report—see [Figure 12 on page 52](#).

You can calculate the average record length of the database records from the root segment to the segment type that you want to include in the primary data set by summing up the values of AVG SEGM LENGTH in the Segment Statistics report. For example, in the following figure, if you want to include segment ROOTLEV1 and segment DEP1LEV2 in the primary data set, you can compare a logical record length for the primary data set (value specified in the RECORD= operand of the DATASET statement in the DBD) and the sum of the products of AVG SEGM LENGTH and AVG OCCURRENCES PER RECORD values—that is, the value of CONTRIBUTION TO DB RECORD SIZE CUM SIZE, which is 45—of segment ROOTLEV1 and segment DEP1LEV2.

"SEGMENT STATISTICS"

SEGMENT STATISTICS

SEGMENT	TOTAL NBR OF OCCURRENCES	<-----AVG OCCURRENCES-----> PER PARENT	PER RECORD	AVG SEGM LENGTH	<--CONTRIBUTION TO DB RECORD SIZE--> PCT	CUM SIZE
ROOTLEV1	10	1.0	1.0	22	5.5%	22
DEP1LEV2	10	1.0	1.0	23	5.7%	45
DEP2LEV2	11	1.1	1.1	25	6.8%	73
DEP3LEV3	9	0.8	0.9	32	7.1%	101
DEP4LEV2	11	1.1	1.1	79	21.6%	189
DEP5LEV3	9	0.8	0.9	121	27.0%	297
DEP6LEV3	11	1.0	1.1	96	26.2%	403

SEGMENT IN OVERFLOW STATISTICS

SEGMENT	<-----OVERFLOW-----> NUMBER	PCT
ROOTLEV1	N/A	N/A
DEP1LEV2	0	0.0%
DEP2LEV2	0	0.0%
DEP3LEV3	0	0.0%
DEP4LEV2	2	18.2%
DEP5LEV3	4	44.4%
DEP6LEV3	8	72.7%

TOTAL SEGMENTS = 71

<<< VARIABLE LENGTH SEGMENT STATISTICS >>>

SEGMENT	<-----LENGTHS----->				<-----SEGMENT COUNT----->		<-COMPRESSION-->	
	MIN	MAX	PFX	DATA	<MIN	=>MIN	FACTOR	NAME
DEP2LEV2	4	40	2	23	0	11	13.6%	DFSCMPX0
DEP4LEV2	50	100	2	77	0	11		
DEP5LEV3	100	200	2	119	4	5		
DEP6LEV3	70	140	2	94	0	11		

Figure 22. Tuning a HISAM database: sample 17

Does the segment fit on KSDS?

The similar sizes allow you to pick logical record lengths and CI sizes so most database records fit on the primary data set.

You want most database records to fit on the primary data set, because additional read and seek operations are required to access those parts of a database record on the overflow data set. Additional reads and seeks degrade performance. If, however, most of the processing you do against a database record occurs on segments in the primary data set (in other words, your high-use segments fit on the primary data set), these considerations might not be as important.

Discussion of the example

The number of segments in the primary data set (KSDS) and the overflow data set (ESDS) can be checked in the Segment Statistics report. See [Figure 22 on page 86](#).

Chapter 8. Analyzing and resolving fragmentation and optimization exceptions for index databases

Analyzing and tuning index databases such as HIDAM primary indexes, secondary indexes, and HALDB partitioned secondary indexes includes four major tasks: analyzing space use exceptions, analyzing fragmentation exceptions, analyzing optimization exceptions, and resolving these exceptions.

The following topics explain how to analyze the database states based on the exceptions that you received and how to recover from the exceptional states.

Topics:

- [“Analyzing fragmentation exceptions for index databases” on page 87](#)
- [“Analyzing optimization exceptions for index databases” on page 88](#)
- [“Resolving space use exceptions for index databases” on page 89](#)
- [“Resolving fragmentation exceptions for index databases” on page 90](#)
- [“Resolving optimization exceptions for index databases” on page 91](#)

Analyzing fragmentation exceptions for index databases

Analyze the fragmentation exceptions that were detected for index databases.

Before you begin

Ensure that you can access the Sensor Data Statistics report that contains the latest sensor data. If you cannot access this report, run the FF DB Sensor Printing utility to generate the report.

About this task

In the following procedure, you will analyze the fragmentation exception that you received for HIDAM primary indexes, secondary indexes, or HALDB partitioned secondary indexes.

The only fragmentation exception for index databases is EXCESSIVE_INDEX_CI_OR_CA_SPLITS. You will identify the relevant data elements to understand the condition of the index database.

By completing this task, you will identify the type of fragmentation symptoms that occurred and the index data set for which the exception was reported.

Procedure

Follow the instruction to analyze the EXCESSIVE_INDEX_CI_OR_CA_SPLITS exception.

Exception class name: EXCESSIVE_INDEX_CI_OR_CA_SPLITS

This exception applies to VSAM KSDSs such as HIDAM primary index databases, the primary data set of secondary index databases, and partitioned secondary index databases. This exception does not apply to VSAM ESDSs such as the overflow data set of secondary index databases.

This exception indicates that the number of control interval (CI) splits, control area (CA) splits, or both are high, and that the VSAM KSDS file is fragmented.

An excessive number of CI splits and CA splits might affect the I/O performance efficiency of applications that perform a massive amount of sequential reading. For information about the effects of an excessive number of CI splits or CA splits, see the topic "Optimizing Free Space Distribution" in *z/OS DFSMS Using Data Sets*.

Locate the following data elements in the Sensor Data Statistics report:

DBX_NUM_CI_SPLIT
DBX_PCT_NUM_CI_SPLIT

The values for these data elements indicate the number of split CIs and the percentage of split CIs compared to the total number of CIs.

DBX_NUM_CA_SPLIT
DBX_PCT_NUM_CA_SPLIT

The values for these data elements indicate the number of split CAs and the percentage of split CAs compared to the total number of CAs.

You have now obtained information about the current database state that caused this exception. You will use this information later to determine the appropriate action to resolve the exception.

What to do next

If you received other exceptions that belong to other exception categories, see the following topics to determine your next step:

- [“Exception categories for full-function databases” on page 18](#)
- [“Scenario for using the analysis and resolution topics” on page 19](#)

If EXCESSIVE_INDEX_CI_OR_CA_SPLITS is the only exceptions that you received, [“Resolving fragmentation exceptions for index databases” on page 90](#).

Analyzing optimization exceptions for index databases

Analyze the optimization exceptions that were detected for index databases.

Before you begin

Ensure that you can access the Sensor Data Statistics report that contains the latest sensor data. If you cannot access this report, run the FF DB Sensor Printing utility to generate the report.

About this task

If a secondary index database has an overflow data set defined, duplicated index pointer segments are located in the overflow data set. Accessing the index pointer segments in the overflow data set requires extra I/O operations. Consequently, when an excessive number of index pointer segments exist in the overflow data set, the condition might cause performance degradation.

Optimizing secondary index databases typically includes tuning the database definitions to decrease the number of index pointer segments in the overflow data set.

In this task, you will identify the number of index pointer segments in the overflow data set, and determine whether you can reduce these segments.

For HIDAM primary indexes, HALDB partitioned secondary indexes, and secondary index databases without an overflow data set, you do not need to analyze duplicated keys.

Procedure

In this section, instructions to analyze the common optimization exceptions are provided. Follow the instructions for the exceptions that you received.

Exception class name: NUMBER_OF_INDEX_POINTER_SEGMENTS

This exception indicates that the number of index pointer segments (IPS) is high.

This exception does not always indicate a problem. When many index pointer segments exist in the overflow data set, determine whether you need to take an action to resolve this exception. Follow the analysis steps for the EXCESSIVE_INDEX_OVERFLOW exception to determine if the secondary index database needs to be worked on.

Exception class name: EXCESSIVE_INDEX_OVERFLOW

This exception indicates that many index keys are duplicated. This exception is reported for secondary indexes that have an overflow data set.

When index keys are duplicated, only the index pointer segment (IPS) that is generated first is placed in the primary data set. Other IPSs that have the same key are placed in the overflow data set. IPSs that are placed in overflow data sets require extra I/O operations to be read.

Complete these steps to analyze this exception:

- a. Locate the following data elements in the Sensor Data Statistics report:
 - DBX_NUM_IPS
 - DBX_NUM_IPS_OVFL
 - DBX_PCT_IPS_OVFL

The values for these data elements indicate the numbers of IPSs, IPSs in the overflow data set, and the percentage of IPSs in the overflow data set compared to the total number of IPSs. Analyze these values to determine why you received this exception.

- b. Optional: To obtain further information about the index database or the indexed database, run the HD Pointer Checker utility of IMS HP Pointer Checker, and generate the following reports:

Scan of Index Database report

You can obtain various information about index databases such as a summary description of data sets (VSAM KSDS and ESDS) of the index database and the total number of records (of various types) that were involved in the processing of the index database.

Index Segment Counts Statistics report

You can obtain information about index counts and indexed databases such as the index database and indexed database names, source and target segment names, number of index pointer and source segments, and the number of suppressed segments.

This report is contained in the Evaluation of All Pointers to the Same Target report.

What to do next

If you received other exceptions that belong to other exception categories, see the following topics to determine your next step:

- [“Exception categories for full-function databases” on page 18](#)
- [“Scenario for using the analysis and resolution topics” on page 19](#)

If the only exceptions that you received are optimization exceptions, continue with [“Resolving optimization exceptions for index databases” on page 91](#).

Resolving space use exceptions for index databases

Resolve the space use exceptions that you received for your index databases.

Before you begin

Ensure that you have completed the analysis steps for index database exceptions and that you have identified the possible causes of the database exceptions. If you have not yet analyzed the space use exceptions, see [Chapter 4, “Analyzing space use exceptions for all full-function database types,” on page 29](#).

Procedure

If you receive space use exceptions for index databases, the method you apply to resolve the exceptions depends on the results of the analysis steps as follows:

Tip: Depending on the status of the data set, applying several of the following methods can be an effective approach to resolving the exceptions. Before making any changes to your environment, evaluate all the methods in the following list and determine the best approach to resolve the condition.

- If the out-of-space condition is caused by a *primary-space-used-up* condition (that is, the data set has only primary space, and the size of used space is reaching the size of the primary allocated space), and if the data sets allocation size is smaller than the maximum size (4 GB or 8 GB), increase the size of the allocated space. For instructions for increasing the size of allocated space, see the topic "Changing the amount of space allocated" in *IMS Database Administration*.
- If the out-of-space condition is caused by a *data-set-full* condition (that is, the data set approaching its maximum size of 4 GB based on the value of the DBX_RBA_HIGH_USED data element), new index segments cannot be created in the index database data set.

Consider the following methods to resolve a data-set-full condition:

- If your HIDAM primary index database is approaching its maximum size, you must reduce the number of root segments in the HIDAM database. To reduce the number of root segments, split the HIDAM database to several smaller databases, or migrate to a PHIDAM database.
- If your secondary index primary data set, overflow data set, or both are approaching their maximum size, you must reduce the number of index pointer segments. To reduce the number of index pointer segments, split the database to several smaller databases, or migrate to a HALDB.

If you do not need certain pointer segments to be built, consider suppressing index entries (sparse indexing). For more information about sparse indexing, see the topic "How sparse indexing suppresses index entries" in *IMS Database Administration*.

- If your partitioned secondary index database data set is approaching its maximum size, split the partition of partitioned secondary index into several smaller partitions.
- If the out-of-space condition is caused by an *extent-resources-used-up* condition (that is, when the data set cannot be expanded), typically, changing the data set allocation definitions and reorganizing the database will resolve the condition.

Consider the following methods to resolve an extent-resources-used-up condition:

- If the data set has reached the maximum number of data set extent counts, and if the DASD volumes have enough space, increase the primary and the secondary extent sizes for data set allocation.
- If the DASD volume does not have enough space, or if the data set has reached the maximum number of extents that are defined for that volume, increase the number of DASD volumes.

Generally, as the number of extents or the number of DASD volumes increases, I/O operations also increase. Therefore, consider using a DASD volume that has sufficient size, and specify an extent size that is large enough for data set allocation.

Resolving fragmentation exceptions for index databases

Resolve the fragmentation exceptions that you received for your index databases.

Before you begin

Ensure that you have completed the analysis steps for index database exceptions and that you have identified the possible causes of the database exceptions. If you have not yet analyzed the fragmentation exceptions, see [“Analyzing fragmentation exceptions for index databases”](#) on page 87.

Procedure

The only fragmentation exception for index databases is the EXCESSIVE_INDEX_CI_OR_CA_SPLITS exception.

If you observe no performance degradation in IMS applications that use the indexed databases, you can ignore this exception.

If you observe performance degradation, you can resolve this exception by removing the CI splits and CA splits by rebuilding the index database. However, performance degradation can be caused by the indexed databases rather than the index databases. Therefore, consider analyzing the fragmentation and optimization exceptions that were issued for the indexed databases.

Resolving optimization exceptions for index databases

Resolve the optimization exceptions that you received for your index databases.

Before you begin

Ensure that you have completed the analysis steps for index database exceptions and that you have identified the possible causes of the database exceptions. If you have not yet analyzed the optimization exceptions, see [“Analyzing optimization exceptions for index databases”](#) on page 88.

Procedure

Reduce the number of duplicated keys. To reduce the duplicated keys, you must change the DBD definition, and reconstruct the indexed databases and index database.

For information about making the keys unique, see the topic "Making keys unique using system related fields" in *IMS Database Administration*.

Part 3. Analyzing and tuning IMS data entry databases

You can monitor, analyze, and tune IMS data entry databases (DEDBs) by using the tools and components that are provided by IMS Fast Path Solution Pack, IMS Tools Base, and IMS Administration Foundation or Management Console.

The following topics explain how to analyze and tune DEDBs.

Topics:

- [Chapter 9, “Overview of DEDB analysis and tuning,” on page 95](#)
- [Chapter 10, “Understanding the status of DEDB areas,” on page 97](#)
- [Chapter 11, “Determining actions to resolve exceptions,” on page 115](#)
- [Chapter 12, “Tuning DEDB areas,” on page 119](#)

Chapter 9. Overview of DEDB analysis and tuning

DEDB tuning typically involves setting the database and area attributes to minimize the physical I/O requirements.

Subsections:

- [“Understanding the parts of a DEDB area” on page 95](#)
- [“Major tuning factors” on page 95](#)
- [“Typical performance characteristics” on page 96](#)
- [“Criteria for triggering tuning activities” on page 96](#)

Understanding the parts of a DEDB area

A DEDB area consists of three parts:

- Root addressable area part (RAA)
- Independent overflow part (IOVF)
- Sequential dependent part (SDEP)

The RAA part is divided into units-of-work (UOW), which are the basic elements of space allocation and free space reorganization.

A UOW consists of a user-specified number of CIs located physically contiguous. Each UOW in the RAA part is further divided into a base section (BASE or RAA BASE) and a dependent overflow section (DOVF).

The BASE section contains CIs of a UOW that are addressed by the randomizing module by using the root anchor point (RAP) that is allocated for each of those CIs. These CIs are often referred to as *RAP CIs*.

The DOVF section is used as a logical extension of a CI within that UOW, and are shared by all RAP CIs in the UOW. Root and direct dependent segments are stored in the BASE section. However, both can be stored in the DOVF section if free space in the RAP CI for the root segment is not enough.

The IOVF part contains CIs that can be used by any UOW in the area. When a UOW gets a CI from the IOVF part, the CI can be used only by that UOW. A CI in the IOVF part (IOVF CI) can be considered an extension of the DOVF section in the RAA part. The IOVF CI remains allocated to a specific UOW unless, after a reorganization, it is no longer required, at which time it is freed.

Physical I/O requirements typically depend on how these parts and sections are used.

For more information about the parts of DEDB areas, see the topic "Parts of a DEDB area" in *IMS Database Administration*.

Major tuning factors

DEDB tuning is a subjective process. Tuning decisions that you make depend on application activities that were performed and that will be performed against the database. Generally, however, you must consider the following factors:

Free space utilization

Predetermining the reorganization criteria to maximize the free space utilization

Definition and sizing of UOW and overflow

Selecting the optimal CI size and the UOW and ROOT parameter values to minimize the overflow to IOVF

Randomizing efficiency

Tuning the randomizer to optimize utilization of RAP CIs and to minimize overflow of database records

Typical performance characteristics

A tuned DEDB area has certain performance characteristics. A well-tuned area typically has the following characteristics:

No or modest use of IOVF

- The percentage of UOWs that use IOVF is low or zero.
- The average and maximum numbers of IOVF CIs that are used by a UOW are low.
- The percentage of database records that use IOVF is low.

Sufficient free space in RAA for each UOW

- The percentage of free space in each section of the area—that is, the RAA BASE section, the DOVF section, and the IOVF section—is sufficient after a free space reorganization.
- The percentage of available IOVF CIs is sufficient so that IOVF CIs are not used up during application activities.

Minimum I/O to access both root segments and database records

- The average and maximum physical I/O that is required to retrieve a root segment are few.
- The average and maximum physical I/O that is required to retrieve a database record are few.

DEDB tuning involves understanding how each DEDB area satisfies the criteria that you set on these performance characteristics and determining the best approach to resolve exceptions to achieve the ideal state of the area.

Criteria for triggering tuning activities

The following criteria are good indicators of how efficiently space is being used and of the overall performance of a DEDB area.

1. Overflow usage
2. Free space availability
3. Free space relationship
4. Root I/O
5. Record I/O
6. Synonym chain

You can use Policy Services to set rules for these six types of criteria. Policy Services uses these rules to detect exceptions based on the criteria and thresholds that you defined for measuring space and performance characteristics. Exceptions that are detected indicate that you likely need to start analyzing and tuning the database with the exception condition. You can also use rules for detecting free space fragmentation exceptions, which identify the need for free space reorganization.

The rules are defined in reorganization policies through Policy Services. Autonomics Director automates the process of exception detection based on the defined policies. For a scenario of policy-based database monitoring, see [Chapter 2, “Scenario: Policy-based database monitoring and tuning,” on page 11.](#)

Chapter 10. Understanding the status of DEDB areas

The first step in tuning databases is to understand the status of DEDB areas from the results of a policy evaluation.

Before you begin

This task helps you understand the database exceptions that were detected and notified by Policy Services.

This topic assumes that you have received Policy Services exceptions or an Autonomics Director recommendation list as a result of policy-based database monitoring.

To set up policy-based database monitoring and to receive monitoring results, see [Chapter 2, “Scenario: Policy-based database monitoring and tuning,”](#) on page 11.

About this task

This topic and subsequent topics provide a tutorial that guides you through the steps for reviewing the state of DEDB areas.

If, during a policy evaluation, a DEDB area is determined to be in an exceptional state, you must review the area. You must understand the exceptions that were detected, and identify the utility reports that provide you the information to help you analyze the exceptional state of the area.

Various reports that are generated by the IMS Fast Path Solution Pack utilities provide information about space utilization problems, randomizing problems, and the I/O efficiency of root segments and database records. This tutorial explains how each of the utility reports is related to certain exceptions and how you can use these reports to analyze and tune a DEDB area.

The following table summarizes the exception categories and related reports that can help you analyze the exceptions in DEDB areas.

Table 6. Exception categories and the corresponding analysis reports

Exception category	Corresponding analysis reports
Free space fragmentation	UOW Exception report
Overflow usage	Freespace Analysis report
Free space availability	Freespace Analysis report
Free space relationship	Freespace Analysis report (IOVF and DOVF usage analysis sections)
Root I/O	<ul style="list-style-type: none">• DB Record Profile Analysis report• DB Record Placement Analysis report• Segment I/O Analysis report• Segment Placement Analysis report
Record I/O	<ul style="list-style-type: none">• DB Record Profile Analysis report• DB Record Placement Analysis report• Segment I/O Analysis report• Segment Placement Analysis report
Synonym chain	Synonym Chain Analysis report

Consider reviewing not only the current state of the area, but also historical trends for the area. To do so, you can store the utility reports from each run of the utility jobs in the Output repository of IMS Tools Knowledge Base and retain these reports for historical analysis. By storing reports in the Output repository, you can easily locate the list of historical reports for a specific DEDB area.

For more information about reports, see the following resources:

- For details about the reports that can be generated by each utility and that can be stored in the Output repository, see the topic "Generating the analysis reports" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.
- For details about searching the reports by using the IMS Tools Knowledge Base ISPF user interface, see the *IMS Tools Base IMS Tools Knowledge Base User's Guide*.
- You can also search the reports by using the graphical interface provided by Management Console. For details about using Management Console, see the *Management Console User's Guide*.

Procedure

The following steps explain the exceptions in each exception category, and explain how to obtain information that will help you understand why the exceptions occurred.

1. Determine if the free space in the area is fragmented.

The following table lists the exceptions that are related to free space fragmentation.

Table 7. Exceptions that are related to free space fragmentation

Exception category	Exception and exception detection rule	Meaning of the exception
Free space fragmentation	DEDB_EXCESSIVE_UOWS_MATCH_COND Rule: IBM.DEDB_RFS.10	The number of UOWs that match the RBASEFS or the RDOVFFS condition exceeded a threshold.
	DEDB_EXCESS_PCT_UOWS_MATCH_COND Rule: IBM.DEDB_RFS.20	The percentage of UOWs that match the RBASEFS or the RDOVFFS condition exceeded a threshold.

The RBASEFS and the RDOVFFS conditions are specified in the database sensor job, and the thresholds are specified in the policy that was used for sensor data evaluation. If you receive one or more of the exceptions that are listed in this table, review the UOW Exception report that was generated when the sensor data was collected.

Requirement: Rules IBM.DEDB_RFS.10 and IBM.DEDB_RFS.20 are not included in the IBM provided default policy SYS.DBDBTYPE.DEDB. You must add either or both of the rules or create your own policy to use these rules. See the topic "Using Policy Services" in the *IMS Tools Base Policy Services User's Guide* to customize an existing policy or create a new policy.

See "Analyzing DEDB areas at the UOW level" on page 112 to understand how to interpret each line of the UOW Exception report. Also use the report to identify the UOWs that were recognized as fragmented.

2. Determine if the exceptions that you received are related to overflow usage and free space availability.

The following table lists the exceptions that are related to overflow usage and free space availability.

Table 8. Exceptions that are related to overflow usage and free space availability

Exception category	Exception and exception detection rule	Meaning of the exception
Overflow usage	DEDB_EXCESS_PCT_UOWS_USING_DOVF Rule: IBM.DEDB_OVERFLOW.10	The percentage of UOWs that are using DOVF exceeded a threshold.
	DEDB_EXCESS_PCT_UOWS_USING_IOVF Rule: IBM.DEDB_OVERFLOW.20	The percentage of UOWs that are using IOVF exceeded a threshold.
	DEDB_EXCESS_NUM_UOWS_USING_IOVF Rule: IBM.DEDB_OVERFLOW.30	The number of UOWs that are using IOVF CIs exceeded a threshold.
	DEDB_EXCESS_AVG_IOVF_CI_PER_UOW Rule: IBM.DEDB_OVERFLOW.40	The average use of IOVF CIs per UOW exceeded a threshold.
	DEDB_UOW_USING_EXCESSIVE_IOVF_CI Rule: IBM.DEDB_OVERFLOW.50	The maximum use of IOVF CIs by a UOW exceeded a threshold.
	DEDB_EXCESS_MIN_IOVF_CI_PER_UOW Rule: IBM.DEDB_OVERFLOW.60	The minimum use of IOVF CIs by a UOW exceeded a threshold.
	DEDB_EXCESSIVE_IOVF_CI_USED Rule: IBM.DEDB_OVERFLOW.70	The percentage of the number of IOVF CIs used exceeded a threshold.
	DEDB_EXCESS_RAP_CI_USING_OVFLOW Rule: IBM.DEDB_OVERFLOW.80	The percentage of RAP CIs that are using overflow exceeded a threshold.
	DEDB_EXCESSIVE_DBREC_USING_IOVF Rule: IBM.DEDB_OVERFLOW.90	The percentage of DB records that are using IOVF exceeded a threshold.
Free space availability	DEDB_FREE_SPACE_AVAIL_IN_RAA Rule: IBM.DEDB_FS.10	The percentage of free space in RAA BASE fell below a threshold.
	DEDB_FREE_SPACE_AVAIL_IN_DOVF Rule: IBM.DEDB_FS.20	The percentage of free space in DOVF fell below a threshold.
	DEDB_FREE_SPACE_AVAIL_IN_IOVF Rule: IBM.DEDB_FS.30	The percentage of free space in IOVF fell below a threshold.
	DEDB_FREE_SPACE_AVAIL_IN_SDEP Rule: IBM.DEDB_FS.80	The percentage of free space in SDEP fell below a threshold.
Free space relationship	DEDB_FREE_SPACE_IN_RAA_VS_DOVF Rule: IBM.DEDB_FS.40	Free spaces in RAA BASE and DOVF are used inefficiently.
	DEDB_FREE_SPACE_IN_RAA_VS_IOVF Rule: IBM.DEDB_FS.50	Free spaces in RAA BASE and IOVF are used inefficiently.
	DEDB_FREE_SPACE_AVAIL_IN_OVFLOW Rule: IBM.DEDB_FS.60	Free spaces in DOVF and IOVF are used inefficiently.
	DEDB_FREE_SPACE_IN_RAA_VS_OVFLOW Rule: IBM.DEDB_FS.70	Free spaces in RAA, DOVF, and IOVF are used inefficiently.

If you receive one or more of the exceptions that are listed in this table, review the free space utilization in the Freespace Analysis report that was generated when the sensor data was collected.

See “Analyzing free space usage” on page 100 to understand how to interpret the Freespace Analysis report and to understand the overflow usage and free space availability of the area.

3. Determine if the exceptions that you received are related to root and record I/Os.

The following table lists the exceptions that are related to root and record I/Os.

Table 9. Exceptions that are related to root and record I/Os

Exception category	Exception and exception detection rule	Explanation
Root I/O	DEDB_EXCESSIVE_AVG_NUM_ROOT_IO Rule: IBM.DEDB_ROOT_IO.10	The average number of I/Os that are required to read a root segment exceeded a threshold.
	DEDB_ROOT_SEGMENT_WITH_EXCESS_IO Rule: IBM.DEDB_ROOT_IO.20	The maximum number of I/Os that are required to read a root segment exceeded a threshold.

Table 9. Exceptions that are related to root and record I/Os (continued)

Exception category	Exception and exception detection rule	Explanation
Record I/O	DEDB_EXCESSIVE_AVG_NUM_RECORD_IO Rule: IBM.DEDB_DBREC_IO.10	The average number of I/Os that are required to read a database record exceeded a threshold.
	DEDB_DBRECORD_WITH_EXCESSIVE_IO Rule: IBM.DEDB_DBREC_IO.20	The maximum number of I/Os that are required to read a database record exceeded a threshold.

If you receive one or more of the exceptions that are listed in this table, review the database record profile and other relevant reports that were generated when the sensor data was collected.

See “Analyzing database record profile” on page 105 to understand the cause of these I/O exceptions.

- Determine if the exceptions that you received are related to synonym chains.

The following table lists the exceptions that are related to synonym chains.

Table 10. Exceptions that are related to synonym chains

Exception category	Exception and exception detection rule	Meaning of the exception
Synonym chain	DEDB_EXCESSIVE_AVG_LEN_SYNONYMS Rule: IBM.DEDB_SYN_LEN.10	The average length of RAP synonym chains exceeded a threshold.
	DEDB_LONG_SYNONYM_CHAIN Rule: IBM.DEDB_SYN_LEN.20	The maximum length of RAP synonym chains exceeded a threshold.

If you receive one or more of the exceptions that are listed in this table, review the distribution of randomizing synonyms, the relationship with segment overflow, and the effects on I/O efficiency in the Synonym Chain Analysis report that was generated when the sensor data was collected.

See “Analyzing the randomizing routine” on page 109 to understand how to interpret the Synonym Chain Analysis report and to understand the randomizing efficiency.

- Optional: Analyze the DEDB area at unit-of-work (UOW) level.

For more information about free space utilization and database record length at UOW level, generate and review the UOW report.

See “Analyzing DEDB areas at the UOW level” on page 112 to understand how to interpret the UOW report and to learn more about the area at UOW level.

What to do next

After you have reviewed the status of the DEDB area from the utility reports and have identified the causes of the exceptions, the next step is to determine the actions that will resolve the exceptions. Because exceptions tend to be interrelated, you might have received exceptions from multiple categories. When you are determining the actions to resolve these exceptions, consider these exceptions, not as separate exceptions, but rather as a group of interrelated exceptions. See Chapter 11, “Determining actions to resolve exceptions,” on page 115.

Analyzing free space usage

Analyze the Freespace Analysis report to understand the status of free space usage of an area.

Before you begin

Ensure that you can access the Freespace Analysis report for the subject DEDB area. To generate a Freespace Analysis report, see the topic “Generating the analysis reports” in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

About this task

The amount of free space within each area of a database and the distribution of that free space must be monitored on a regular basis for the following reasons:

- To avoid application failures that might be caused by out-of-space conditions
- To determine whether reorganization is required
- To determine the optimal timing for database expansion

The Freespace Analysis report provides information that helps you determine the optimal values for the UOW and ROOT parameters of a DBD. The report also provides information for monitoring DOVF and IOVF usage to determine the effectiveness of free space reorganization.

By completing the steps in this topic, you can determine the following free space characteristics of the area:

- The total amount of free space in RAA BASE, DOVF, IOVF, and SDEP
- The number and the size of the free space elements in RAA BASE, DOVF, and IOVF
- The relationship between available free space in each section of the area and pointers to other sections, except for the SDEP part and SDEP-related pointers

Procedure

Complete the following steps to understand the status of free space usage of the area:

1. Determine the percentage of free space in each section of the area.

Locate the overview section of the Freespace Analysis report. This section provides information about the free space that is available in each section of the area. Free space availability is the first indicator that you should check.

The following figure shows an example of the overview section of the report.

```

FREESPACE ANALYSIS - *** OVERVIEW ***          %FREESPACE - RAA BASE:  70% %USABLE FREESPACE:  69%
                                                DOVF   :  47%
                                                IOVF   :  45%
                                                SDEP   :  94%
    
```

Figure 23. Overview section of the Freespace Analysis report

2. Determine the free space availability in RAA BASE and the overflow from RAP CIs.

Locate the table for RAA BASE in the report. This table shows the relationship between free space availability in RAP CIs and overflow to DOVF and IOVF.

The following figure shows an example of the RAA BASE table in the report.

```

FREESPACE ANALYSIS - RAA BASE
%FS  NO. CI'S  %CI  ---NO. OF FSE'S---  -----SIZE OF FSE'S-----  # USE  ---PTR. TO DOVF---  ---PTR. TO IOVF---
      AVG  S/D  MAX  MIN  AVG  S/D  MAX  MIN  OVFLOW  AVG  S/D  MAX  MIN  AVG  S/D  MAX  MIN
100   110   56
90-99   0
80-89   0
70-79   25   13   2.0   .0   2   2   376  317   700   53   24   1.2   .8   2   0   .8   .7   2   0
60-69   0
50-59   2    1   2.0   .0   2   2   251  249   500    2    2   1.0   .0   1   1   1.0   .0   1   1
40-49   0
30-39   0
20-29   27   14   1.4   .5   2   1   178  113   251    1   26   1.3   .8   2   0   .5   .7   2   0
10-19   0
0-9     31   16   .0    .2   1   0    2    0    2    0   27   .9   .9   2   0   .8   .9   2   0

OVFL USAGE SUMMARY
CI'S USING OVERFLOW:  NO.:      114  %CI'S:  95%
    
```

Figure 24. RAA BASE table in the Freespace Analysis report

By analyzing the data in this table, you can identify the state of the following free space characteristics:

%FS

The range of total free space percentages in a RAP CI in the RAA BASE

NO. CI'S

Distribution of RAP CIs by %FS range

NO. OF FSE'S

The number of the free space elements (FSEs) in RAP CIs for each %FS range

SIZE OF FSE'S

The size of the FSEs in RAP CIs for each %FS range

#USE OVFLOW

The amount of RAP CIs that overflow to DOVF or IOVF for each %FS range

PTR. TO DOVF

The amount of overflow to DOVF for each %FS range

PTR. TO IOVF

The amount of overflow to IOVF for each %FS range

You can determine if fragmented free space elements exist by analyzing the total bytes of free space in RAP CIs.

- If the total number of bytes of free space in RAP CIs is sufficient, but the number of free space elements that are smaller than the average segment size has increased, you can assume that fragmented free space elements exist.
 - If the total size of free space in RAP CIs is sufficient, but the RAP CIs have many references to the DOVF or IOVF section, you can assume that free space fragmentation or bulk deletion of segment occurrences has occurred in the RAP CIs after overflow.
3. Determine to what extent UOWs are using the DOVF and IOVF sections.

Locate the DOVF and IOVF sections in the report. The following figure shows an example of the DOVF and IOVF sections of the report.

FREESPACE ANALYSIS - DOVF

%DOVF FREE	#UOWS	% UOW	-DOVF CI'S USED				-NO. OF FSE'S--				----SIZE OF FSE'S----			--PTR TO BASE--			--PTR TO DOVF--			--PTR TO IOVF--				
			AVG	S/D	MAX	MIN	AVG	S/D	MAX	MIN	AVG	S/D	MAX	MIN	AVG	S/D	MAX	MIN	AVG	S/D	MAX	MIN		
100	13	20																						
90-99	0																							
80-89	1	2	1.0	.0	1	1	1.0	.0	1	1	753	0	753	753	0		0							
70-79	13	20	1.0	.0	1	1	1.3	.5	2	1	383	174	502	76	0		0		2.2	2.9	11	0		
60-69	0																							
50-59	0																							
40-49	0																							
30-39	0																							
20-29	29	45	2.0	.0	2	2	1.0	.3	2	0	238	182	502	0	0		.2	.5	2	0	.6	1.0	4	0
10-19	0																							
0-9	9	14	2.0	.0	2	2	0				0						1.0	1.2	4	0	1.3	1.5	4	0

DOVF USAGE SUMMARY

UOW'S USING DOVF: NO.: 52 %UOW'S: 80%
 NO. DOVF CI'S USED: AVG: 1.73 SDEV: .44 MAX: 2 MIN: 1

FREESPACE ANALYSIS - IOVF

%DOVF FREE	#UOWS	% UOW	#USE IOVF	-IOVF CI'S USED				-NO. OF FSE'S--				----SIZE OF FSE'S----			--PTR TO BASE--			--PTR TO DOVF--			--PTR TO IOVF--		
				AVG	S/D	MAX	MIN	AVG	S/D	MAX	MIN	AVG	S/D	MAX	MIN	AVG	S/D	MAX	MIN	AVG	S/D	MAX	MIN
100	13	20																					
90-99	0																						
80-89	1	2	0																				
70-79	13	20	10	1.3	1.0	3	1	.3	.5	1	0	323	291	753	0	0		0		.2	.4	1	0
60-69	0																						
50-59	0																						
40-49	0																						
30-39	0																						
20-29	29	45	22	1.3	1.1	4	1	.5	.5	1	0	305	283	945	0	0		0		.1	.4	2	0
10-19	0																						
0-9	9	14	6	1.2	1.2	4	1	.1	.4	1	0	24	21	45	0	0		0		1.1	1.4	3	0

IOVF USAGE SUMMARY

UOW'S USING IOVF: NO.: 48 %UOW'S: 74%
 NO. IOVF CI'S USED: AVG: 1.89 SDEV: .94 MAX: 4 MIN: 1
 AVAILABLE IOVF CI'S: NO.: 57 %CI'S: 38%

Figure 25. DOVF and IOVF sections of the Freespace Analysis report

In these sections, check for the following information:

- For UOWs that use IOVF, determine if any free space is available in the DOVF. Refer to the threshold values that you set for data elements DB_PCT_BYTES_FS_DOVF and DB_PCT_BYTES_FS_IOVF (in rule IBM.DEDB_FS.60) to determine if free space in DOVF and IOVF is being used efficiently.

If free space is available in the DOVF of the UOW, and if the IOVF is being used by the UOW, the space in the DOVF might not be used efficiently or the free space might need to be reorganized.

IOVF should not be used if a sufficient amount of free space is available in the DOVF section of most UOWs.
- In the DOVF usage summary section, check the value in the UOW'S USING DOVF field. Refer to the threshold values that you set for data element DB_PCT_NUM_UOW_USE_DOVF (in rule IBM.DEDB_OVERFLOW.10) to determine if the percentage of UOWs that use the DOVF section is low enough.

The percentage of UOWs that use DOVF sections should not be high.
- In the IOVF usage summary section, check the value in the UOW'S USING IOVF field. Refer to the threshold values that you set for data element DB_PCT_NUM_UOW_USE_IOVF (in rule IBM.DEDB_OVERFLOW.20) or DB_NUM_UOW_USE_IOVF (in rule IBM.DEDB_OVERFLOW.30) to determine if the percentage or the number of UOWs that use the IOVF section is low enough.

The values should be zero for those areas that are heavily accessed and that need high performance.
- In the IOVF usage summary section, check the values in the NO. IOVF CI'S USED field. Refer to the threshold values that you set for data elements DB_AVG_NUM_IOVFCI_BY_UOW (in rule IBM.DEDB_OVERFLOW.40) and DB_MAX_NUM_IOVFCI_BY_UOW (in rule IBM.DEDB_OVERFLOW.50) to determine if the average and maximum use of IOVF CIs per UOW are low enough.

The values should be zero for those areas that are heavily accessed and that need high performance.

The DOVF and IOVF sections also provide information about whether skewed distribution of overflow exists among UOWs.

4. Optional: Analyze free space utilization and overflow at UOW level.

To obtain more information about free space utilization at UOW level, especially when you identify skewed distribution of overflow among UOWs, generate and analyze the UOW report.

See [“Analyzing DEDB areas at the UOW level”](#) on page 112 to understand how to interpret the UOW report and to understand more about the area at the UOW level.

5. Summarize the observation of Freespace Analysis report.

First, determine if the free space of the area needs to be reorganized. Free space reorganization is recommended if one or both of the following conditions are observed:

- Free space elements are fragmented, and the percentage of FSEs with sizes that are smaller than the sizes of segments to be added has been increased.
- The total size of free space in the CIs is sufficient, but there are numerous references to other sections of the area.

Next, determine if the areas need to be expanded based on your analysis of the free space availability in Step [“1”](#) on page 101. You might need to expand the areas if the packing density of RAA BASE is high (based on your criteria) and the size of free space in both DOVF and IOVF has fallen below the predetermined threshold that is set for each section.

You might need to expand the SDEP part if the amount of free space has fallen below a threshold value for the SDEP free space.

The packing density of RAA BASE is the value that shows the percentage of bytes that are used by segment data compared to the total bytes of all the allocated RAP CIs in the RAA BASE (here, total bytes do not include the bytes that are used by VSAM and IMS control data). You can calculate the packing density by using the following formula:

$$100 - \text{the percentage of free space in RAA BASE}$$

where *the percentage of free space in RAA BASE* is the value that is shown in the Overview section of the Freespace Analysis report (See Step [“1”](#) on page 101).

What to do next

If the only exceptions that you received are related to the overflow usage and free space availability, and if you determined that free space needs to be reorganized, follow the instructions in [“Planning free space reorganization and usage monitoring”](#) on page 116 to reorganize free space and to review the reorganization schedule.

If you received other exceptions that are not related to free space availability, and you determined that free space does not need to be reorganized, continue with the steps in [Chapter 10, “Understanding the status of DEDB areas,”](#) on page 97.

If you received other exceptions that are not related to free space availability, and if you determined that free space needs to be reorganized and the area needs to be expanded, consider reorganizing free space before working on other exceptions. Because the exceptions are interrelated, free space reorganization might reduce the number or lower the severity of exceptions. After reorganizing the areas, run policy evaluation for the reorganized areas to determine how the free space reorganization changed the situation by following the steps in [Chapter 10, “Understanding the status of DEDB areas,”](#) on page 97.

Analyzing database record profile

Analyze the reports that provide the database record information so that you can understand the statuses of database records in the DEDB area.

Before you begin

Ensure that you can access the following reports for the subject DEDB area:

- DB Record Profile Analysis report
- DB Record Placement Analysis report
- Segment I/O Analysis report
- Segment Placement Analysis report
- Largest Database Records report (optional)

Not all utilities of IMS Fast Path Solution Pack can produce this report. Use the appropriate utilities and runtime options to generate these reports.

For details about the utility options and reports that can be generated, see the topic "Generating the analysis reports" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

About this task

One of the most important tasks in analyzing DEDB areas is analyzing the database record profile. The process of analyzing a database record profile includes reviewing the effect on I/O efficiency that might be caused by the changes in database record characteristics. Results from this analysis are the key factors for determining the best approach to resolve the exceptions.

In this task, you will review the average length of the database records and then review the patterns and statistics of database record overflow. You will also review the database record overflow at the segment occurrence level. These factors help you learn how I/O efficiency is affected by segment overflow.

Optionally, if you identify a large database record while reviewing the length of database records, you will also learn how to obtain more information about such records. You will also learn how to obtain information about database records at the UOW level.

By completing this task, you will determine whether the database record profile has been changed and whether the change might affect I/O efficiency.

Procedure

Complete the following steps to understand the statuses of database records in the area:

1. Check the average length of database records.

The DB Record Profile Analysis report provides statistics about the number and length of the database records and statistics about the frequencies and length of segment occurrences.

The following figure is an example of the DB Record Profile Analysis report.

DB RECORD PROFILE ANALYSIS

NO. DB RECORDS: 168
 AVG REC LENGTH: 1,234
 STD. DEV. : 879
 MAX REC LENGTH: 4,012
 MIN REC LENGTH: 218

SEGNAME	SEG CD	SEG LVL	TOT #OCCS	-----SEGMENT FREQUENCY-----				DEFINED LENGTH		-----ACTUAL LENGTH-----			
				AVG.	SDEV	MAX	MIN	MAX	MIN	AVG.	SDEV	MAX	MIN
ROOTSEG1	1	1	168	N/A	N/A	N/A	N/A	900	40	243	145	900	134
DD1	3	2	280	1.67	1.34	6	0	900	30	292	206	900	102
DD11	4	3	261	1.55	2.79	18	0	900	20	226	105	900	154
DD12	5	3	0					100	*				
DD2	6	2	60	0.36	0.61	2	0	50	*	52	0	52	52

Figure 26. DB Record Profile Analysis report

The DB Record Profile Analysis report provides information that helps you determine whether to expand CI size and to calculate the optimal values for the ROOT and UOW parameters for the DBD.

Generally, if the average database record length multiplied by the average number of root segments on a RAP is less than the current CI size, the CI size is appropriate. If the calculated value significantly exceeds the current CI size, consider tuning the CI size and UOW definition, changing the randomizer, or both.

Tip: The average number of root segments on a RAP can be found in the Roots per RAP Distribution Graph. See [“Analyzing the randomizing routine”](#) on page 109.

2. Analyze the patterns and statistics of database record overflow.

The DB Record Placement Analysis report includes statistics about the number of database records that are contained in each of the following seven placement categories:

- Records that are placed only in BASE section
- Records that are placed in BASE and DOVF sections
- Records that are placed in BASE, DOVF, and IOVF sections
- Records that are placed in BASE and IOVF sections
- Records that are placed only in DOVF section
- Records that are placed in DOVF and IOVF section
- Records that are placed only in IOVF section

The following figure is an example of the DB Record Placement Analysis report.

DB RECORD PLACEMENT ANALYSIS

	NO. RECS	P/C	---DOVF CI'S USED---				---IOVF CI'S USED---				
			AVG.	SDEV	MAX	MIN	AVG.	SDEV	MAX	MIN	
RECORDS IN: BASE ONLY	54	32.1									
BASE + DOVF	29	17.3	1.14	.34	2	1					
BASE + DOVF + IOVF	33	19.6	1.21	.41	2	1	1.18	.46	3	1	
BASE + IOVF	17	10.1					1.29	.67	3	1	
DOVF ONLY	9	5.4	1.44	.50	2	1					
DOVF + IOVF	23	13.7	1.43	.50	2	1	1.78	1.02	4	1	
IOVF ONLY	3	1.8					2.67	1.25	4	1	

Figure 27. DB Record Placement Analysis report

The DB Record Placement Analysis report provides information that helps you determine whether to expand CI size and whether to change the UOW definition in the DBD.

In an ideal area, all database records reside only in the BASE section. If many database records reside outside of the BASE section, you must review other reports to further investigate the cause of overflow and to avoid many database records overflowing to IOVF.

3. Check the database record overflow in segment occurrence level.

Browse the Segment Placement Analysis report. The Segment Placement Analysis report provides an overview of where the segments reside in the area, especially occurrences of segment types that contribute to overflow usage. This report provides proportion of the segments that reside in RAA BASE, DOVF, and IOVF for each segment type.

The following figure is an example of the Segment Placement Analysis report.

SEGMENT PLACEMENT ANALYSIS											
SEGNAME	SCD	LVL	TOT #OCCS	----IN RAA BASE---		-----IN DOVF-----		-----IN IOVF-----		NO. OCCS	P/C
				NO. OCCS	P/C	NO. OCCS	P/C	NO. OCCS	P/C		
ROOTSEG1	1	1	168	133	79.2	32	19.0	3	1.8		
DD1	3	2	280	47	16.8	105	37.5	128	45.7		
DD11	4	3	261	60	23.0	33	12.6	168	64.4		
DD12	5	3	0								
DD2	6	2	60	52	86.7	0	0.0	8	13.3		

Figure 28. Segment Placement Analysis report

4. Determine how I/O efficiency is affected by segment overflow.

The Segment I/O Analysis report provides information about the database record I/Os in the granularity of the number of segment I/Os.

This report provides the following statistics:

- Contribution of occurrences of each segment type to a database record
- The number of physical I/Os that are required to retrieve the root segment
- The number of physical I/Os that are required to retrieve an occurrence of each segment type
- The number of physical I/Os that are required to retrieve an entire database record

The Segment I/O Analysis report provides information that is useful for evaluating the database I/O performance requirements for specific user transactions. It is also used to validate the effects of UOW and ROOT value parameter changes.

The following figure is an example of the Segment I/O Analysis report. The report shows the average, standard deviation from the average, maximum, and minimum number of I/Os to read a database record or a root segment.

SEGMENT I/O ANALYSIS - AVERAGE DB RECORD													
SEGNAME	SEG CD	SEG LVL	AVG. FREQ	-----SEGMENT PLACEMENT STATISTICS-----						---PHYSICAL I/O STATS---			
				--IN RAABASE--		---IN DOVF---		---IN IOVF---		AVG.	SDEV	MAX	MIN
				AVG./SDEV	MAX/MIN	AVG./SDEV	MAX/MIN	AVG./SDEV	MAX/MIN				
ROOTSEG1	1	1	1.00	0.79 0.41	N/A	0.19 0.39	N/A	0.02 0.13	N/A	1.21	0.41	2	1
DD1	3	2	1.67	0.28 0.45	1 0	0.63 0.75	3 0	0.76 1.30	6 0	1.08	1.09	5	0
DD11	4	3	1.55	0.36 0.77	2 0	0.20 0.74	4 0	1.00 2.73	18 0	0.05	0.27	2	0
DD12	5	3	0.00										
DD2	6	2	0.36	0.31 0.59	2 0	0.00		0.05 0.24	2 0	0.02	0.15	1	0
** RECORD I/O **			AVG:	2.36	SDEV:	1.31	MAX:	7	MIN:	1			
*** ROOT I/O ***			AVG:	1.21	SDEV:	0.41	MAX:	2	MIN:	1			

Figure 29. Segment I/O Analysis report

5. Optional: Obtain more information about large database records.

Tuning an area of a database that has exceptionally large database records can be difficult.

You can obtain a list of database records and information about these records by browsing the Largest Database Records report. This report lists the database records in descending order by their size, which can help you identify the database records that are exceptionally large.

Analyzing the randomizing routine

Analyze the Synonym Chain Analysis report to understand how efficiently the root segments are distributed by the randomizing routine (randomizer).

Before you begin

Ensure that you can access the Synonym Chain Analysis report for the subject DEDB area.

Not all utilities of IMS Fast Path Solution Pack can produce this report. Use the appropriate utilities and runtime options to generate these reports. For details about utility options and reports that can be generated, see the topic "Generating the analysis reports" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Before you start, consider reviewing the database record profile to obtain the average length of database records, the average number of root I/Os, and the average number of database record I/Os even if you have not received exceptions that are related to the average database record length, root I/O, or record I/O (see "[Analyzing database record profile](#)" on page 105). These figures are useful indicators to complete the steps in this topic.

About this task

One of the most frequently overlooked aspects of the database design and implementation process is the selection of the randomizer that best matches the key values of the database records. Using a randomizer that does not match the key values can significantly affect the performance characteristics of a database.

In this task, you will first check the synonym chain distribution, then check the synonym chain effect on root I/Os.

By completing this task, you will be able to evaluate whether the current randomizing logic matches the current distribution of database record key values. If the logic does not match, you will be able to evaluate how this mismatch affects the efficiency of root I/Os.

Procedure

Complete the following steps to determine the randomizing efficiency:

1. Check the synonym chain distribution.

Browse the Roots per RAP Distribution Graph in the Synonym Chain Analysis report. The following figure shows an example of the Roots per RAP Distribution Graph in the Synonym Chain Analysis report.

ROOTS PER RAP DISTRIBUTION GRAPH

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	% RAPS
#	0- *	(234)													
	1- ***	(963)													
R	2- *****	(2009)													
O	3- *****	(2834)													
O	4- *****	(2954)													
T	5- *****	(2493)													
S	6- *****	(1812)													
	7- ***	(995)													
/	8- **	(569)													
	9- *	(258)													
R	10- *	(124)													
A	11- *	(32)													
P	12- *	(16)													
	13- *	(7)													
	14-														

SYNONYM CHAIN STATISTICS

SYN CHAIN LONGEST:	13	RAP RBA:	006C4000
SYN CHAIN MAX I/O:	2	RAP RBA:	006C4000
SYN CHAIN LENGTH :	AVG:	4.49	
NO. ROOTS PER RAP:	AVG:	4.27	

Figure 31. Roots per RAP Distribution Graph of Synonym Chain Analysis report

Analyze this graph and the synonym chain statistics below the graph to determine the following information:

- Whether the general distribution of RAP synonym chain lengths is acceptable.
 - In each row, the number on the left represents the number of root segments that are assigned to a RAP. The number of asterisks (*) represents the percentage of the RAPs to which the same number of root segments are assigned. The numbers on the right (in parentheses) represent the number of RAPs. For example, the third row (where the number on the left is 2) represents the percentage and the number of the RAPs to which two root segments are assigned.
 - The average length of RAP synonym chains is shown on the SYN CHAIN LENGTH line in the Synonym Chain Statistics section.

Generally, if the peak in the graph is near the bottom, the graph suggests that many long synonym chains exist.

- Whether an exceptionally long synonym chain exists.

Existence of such synonym chain might indicate that the randomizer does not match the distribution of root sequence key values.
- Whether the percentage of unused RAPs, which is indicated by the first row in the graph, is high.

If this number is high and the value for the average length of RAP synonym chains is also high, the current randomizer might not match the distribution of root sequence key values.

2. Check the synonym chain effect on root I/Os.

Browse the synonym chain analysis section in the Synonym Chain Analysis report.

This section of the report shows statistics about the placement of root segments in RAA BASE, DOVF, and IOVF sections of the area. The report also shows the effect on the number of physical I/Os that would be required to read a root segment in a synonym chain for each synonym chain length.

The first three columns are tabular representations of the Roots per RAP Distribution Graph.

The next six columns indicate, for each RAP synonym chain length, the following statistics about the root segment occurrences in each section of the area:

- The average number (AVG.)

- The maximum number (MAX)
- The minimum number (MIN)
- The standard deviation (SDEV)

The first two columns are for the RAA BASE section, the next two columns are for the DOVF section, and the last two columns are for the IOVF section.

AVG./SDEV

The AVG./SDEV columns (the first column for each section) indicate the average number of root segment occurrences over all the synonym chains of that chain length are displayed with the standard deviation from the average.

MAX/MIN

The MAX/MIN columns (the second column for each section) indicate the maximum and the minimum number of root segment occurrences among all the synonym chains of that chain length are displayed.

For each section, a low standard deviation indicates that, within a group of synonym chains that have the same length, the number of root segments that occur in a synonym chain of that group and that reside in that section tends to be very close to the average number, over all synonym chains in that group, of root segment occurrences in that section. A high standard deviation indicates that the root segment occurrences are spread out over a large range of values.

The last four columns indicate the statistics for physical I/Os as follows:

AVG.

The AVG. column indicates the average number of I/Os that are required to read a root segment in the synonym chain of that chain length. The average is calculated based on all the synonym chains that have the same length.

SDEV

The SDEV column indicates the standard deviation from the average number.

MAX and MIN

The MAX and the MIN columns indicate the maximum and the minimum number of root I/Os among all the synonym chains that have the same length.

In this section, pay attention to the existence of synonym chains that make the average number of I/Os high or that have an exceptionally high (maximum) number of I/Os.

The following figure shows the synonym chain analysis section in the Synonym Chain Analysis report.

SYNONYM CHAIN ANALYSIS

#ROOTS/ RAP	NO. RAPS	P/C	-----ROOT PLACEMENT STATISTICS-----						---PHYSICAL I/O STATS---			
			--IN RAABASE--		---IN DOVF---		---IN IOVF---		AVG.	SDEV	MAX	MIN
			AVG./ SDEV	MAX/ MIN	AVG./ SDEV	MAX/ MIN	AVG./ SDEV	MAX/ MIN				
0	234	1.5	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1	963	6.3	1.00	1	0.00		0.00		1.00	0.00	1	1
2	2,009	13.1	2.00	2	0.00		0.00		1.00	0.00	1	1
3	2,834	18.5	3.00	3	0.00		0.00		1.00	0.00	1	1
4	2,954	19.3	4.00	4	0.00		0.00		1.00	0.00	1	1
5	2,493	16.3	5.00	5	0.00		0.00		1.00	0.00	1	1
6	1,812	11.8	6.00	6	0.00		0.00		1.00	0.00	1	1
7	995	6.5	7.00	7	0.00		0.00		1.00	0.00	1	1
8	569	3.7	8.00	8	0.00		0.00		1.00	0.00	1	1
9	258	1.7	9.00	9	0.00		0.00		1.00	0.00	1	1
10	124	0.8	10.00	10	0.00		0.00		1.00	0.00	1	1
11	32	0.2	11.00	11	0.00		0.00		1.00	0.00	1	1
12	16	0.1	12.00	12	0.00		0.00		1.00	0.00	1	1
13	7	0.0	12.00	12	1.00	1	0.00		2.00	0.00	2	2
14	0		0.00	12	0.00	1						

Figure 32. Synonym chain analysis section of the Synonym Chain Analysis report

What to do next

After you identify the cause of the exceptions, the next step is to determine the actions to tune the subject area. The actions might include tuning of the randomizer to make the database fit the current characteristics of database records and application usage. See Chapter 11, “Determining actions to resolve exceptions,” on page 115 to determine the appropriate tuning tasks.

Analyzing DEDB areas at the UOW level

By analyzing the RAP CI usage, the free space distribution, and database record length distribution at the UOW level, you can obtain information about the status of the DEDB area in more detail.

Before you begin

Ensure that you can access either of the following reports for the subject DEDB area:

- UOW report
- UOW Exception report

About this task

The following two lines show the column headers in the UOW report and the UOW Exception report:

UOW NO.	---NUMBER CI'S USED---			#RAPS USE OVFL CI'S	---%RAP FS---			% DOVF PART FS	#DB RECORDS	----- RECORD LENGTH STATS -----		
	#RAPS (1)	#DOVF (2)	#IOVF (3)		AVG (5)	MAX (6)	MIN (7)			AVERAGE (10)	MAXIMUM (11)	MINIMUM (12)

Figure 33. Column headers in the UOW report

The first column, UOW NO., contains the UOW number, and other 12 columns contain information about the UOW.

The UOW report provides the following information for each UOW that is defined for the area:

- The number of RAP CIs in the UOW that contain data (column 1)

- The number of DOVF CIs that contain data in the UOW (column 2)
- The number of IOVF CIs that are allocated to the UOW (column 3)
- The number of RAP CIs of the UOW that contain a pointer to segment data in either the DOVF or the IOVF section (column 4)
- Statistics about the average, maximum, and minimum percentages of free space in RAP CIs that are used in the UOW (columns 5, 6, and 7)
- The percentage of free space in the DOVF section of the UOW (column 8)
- The number of database records in the UOW (column 9)
- Statistics about the average, maximum, and minimum lengths of the database records in the UOW (columns 10, 11, and 12)

Note: The information in the last two list items is reported only when the utility that generates these reports runs in specific run mode.

You can use the UOW report to monitor and manage space utilization at the UOW level. The report helps you determine if the database records are randomized so that RAP CIs and the space inside RAP CIs are used effectively in each UOW.

The contents of the UOW Exception report for a UOW is the same as that of the UOW report. However, the UOW Exception report is printed only when certain utility options are specified. The report contains information about those UOWs that satisfy specific conditions.

You can use the UOW Exception report to simplify the following tasks:

- To identify which UOWs require reorganization
- To list only the UOWs that match the predefined space utilization criteria

For the utilities that can generate these reports and necessary utility options, see the topic "Generating the analysis reports" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Procedure

Complete the following steps for each UOW that you want to obtain detailed information about:

1. Evaluate the efficiency of the randomizing routine that is being used.

Inefficient randomizing can be identified by the data in columns 1 through 3, which are under the heading NUMBER CI'S USED.

The #RAPS column (column 1) indicates the number of root anchor points in the UOW, to which at least one database record is assigned. The #DOVF column (column 2) and the #IOVF column (column 3) indicate the number of control intervals (CIs) that are used in DOVF and IOVF respectively.

If the number of root anchor points in column 1 is significantly less than the total number of root anchor points per UOW, and if fair amount of DOVF or IOVF CIs are used, inefficient randomizing is possibly present in the UOW.

2. Determine the average length of synonym chains in the UOW.

The #DB RECORDS column (column 9) indicates the number of database records in the UOW. Divide this number by the number in the #RAPS column (column 1) to determine the average number of database records per used RAP CIs. If this value is larger than what you expected, consider tuning the randomizer.

Note: The #DB RECORDS column is printed only when the utility that generates this report runs in a specific run mode.

3. Determine the free space utilization in RAP CIs.

The efficiency of free space utilization in RAP CIs of the UOW can be determined by analyzing the values in columns 4, 5, 6, and 7.

The #RAPS USE OVFL CI'S column (column 4) indicates the number of RAP CIs that contain a pointer to a segment data in either DOVF or IOVF. The %RAP FS AVG column (column 5), the %RAP FS MAX column (column 6), and the %RAP FS MIN column (column 7) indicate the average, maximum, and minimum percentages of free space in used RAP CIs. Empty RAP CIs are not included in these values.

If many RAP CIs use overflow despite the average percentage of free space in RAP CIs is fairly high, it is highly possible that either free space in the RAP CIs is fragmented or fair amount of segment occurrences in the UOW were deleted. In either case, free space reorganization might remedy the overflow from RAP CIs in the UOW.

4. Determine the free space utilization in DOVF.

You can determine the efficiency of free space utilization in the DOVF section of the UOW by analyzing the values in columns 3 and 8.

The %DOVF PART FS column (column 8) indicates the percentage of free space in the DOVF section of the UOW. This value includes all the free space elements that are in the used DOVF CIs and all the empty DOVF CIs.

If an IOVF CI (column 3) is used by the UOW, despite the fact that the percentages of free space in the DOVF section are fairly high, it is likely that free space in the DOVF section is fragmented or a fair amount of segment occurrences in the DOVF section were deleted. In either case, free space reorganization might resolve the overflow to IOVF.

5. Determine the length of database records.

You can determine the average number of database records per used RAP CIs by analyzing the values in columns 1, 9, and 10.

The #DB RECORDS column (column 9) indicates the number of database records. Divide this number by the NUMBER CI'S USED #RAPS column (column 1) to determine the average number of database records per used RAP CIs.

Multiply the calculated value with the average length of database records (the RECORD LENGTH STATS - AVERAGE column, column 10). If the resulting value exceeds the CI size of the area, the randomizing to the RAPs in the UOW might not be optimal or the CI size might not match the average database record length.

If the randomizing is optimal or satisfactory, consider expanding the CI size to improve I/O performance.

Also, check the maximum record length (column 11) to determine if a large database record exists in the UOW. When large database records exist, your tuning step might require optional steps, which are discussed in the tuning topics in [Chapter 12, "Tuning DEDB areas,"](#) on page 119.

Note: The #DB RECORDS column (column 9) and the RECORD LENGTH STATS columns (columns 10, 11, and 12) are printed only when a request is made in a utility job.

Chapter 11. Determining actions to resolve exceptions

After you identify the possible causes of the exceptions, you can determine the best approach for tuning the areas.

Before you begin

Ensure that you have completed the steps in [Chapter 10, “Understanding the status of DEDB areas,”](#) on [page 97](#) and that you have identified the possible causes of the database exceptions.

About this task

In this task, you will determine the best approach for resolving exceptional states of a DEDB area. Key decision indicators are the database record profile of the area, the distribution of randomizing synonyms, and their effects on the root I/Os and database record I/Os.

By completing this task, you will be able to determine whether reorganizing the free space in the area will resolve the exceptional state or whether elaborate tuning tasks, such as tuning database definitions or randomizer, are required.

Procedure

The approach for database tuning varies depending on whether the record profile and the number of database records have changed or not.

If the record profile and the number of database records have been relatively unchanged

If exceptions that are related to free space fragmentation, free space usage, or overflow usage were reported earlier than you expected, consider reviewing the current criteria for reorganization or the current reorganization schedule, and refine them if necessary. If the exceptions are related to free space fragmentation, free space usage, or overflow usage, and if the number of database records in the area has been relatively unchanged, consider reorganizing free space, which does not require changes to the DBD definitions.

Free space reorganization is especially recommended if a free space fragmentation exception was reported or if the Freespace Analysis report indicates either or both of the following conditions:

- Free space elements are fragmented, and the number of free space fragments that are smaller than the segment size has increased
- The percentage of CIs that have enough total free space but that have many references to other sections of the database has increased

See [“Planning free space reorganization and usage monitoring”](#) on [page 116](#) for information about performing free space reorganization.

If the record profile or the number of database records have significantly changed

Determine if tuning of the DBD definitions or the randomizer is required.

As data is added and the packing density of RAA BASE increases, root segments or database records are likely to overflow to DOVF and IOVF. Overflow to DOVF and IOVF can increase the average number of physical I/Os that are required to retrieve a root or a record.

If the average database record length or the number of database records in the area has significantly increased, and the amount of free space in any section of the area has fallen below a predetermined threshold, consider expanding the area. When you expand the area, consider expanding a RAP CI or increasing the number of RAP CIs. When you increase the number of RAP CIs, also consider tuning the randomizer because the randomizing might change.

Database records might overflow from RAA BASE to DOVF or from UOW to IOVF when root segments that have unexpected sequence keys are inserted or when exceptionally large database records exist.

The number of physical I/Os that are required to retrieve a root segment can increase if the root segment is in a long synonym chain. The number of physical I/Os that are required to read a database record can increase if the database record has become large.

Determine if the record profile, including key distribution, or the number of database records in the area has significantly changed from what you expected when you defined the area and when you selected the current randomizer. If the record profile or number of database records in the area have significantly changed, consider tuning the randomizer.

Tip: Tuning of a randomizer is difficult if exceptionally large database records exist in a specific UOW or UOWs. When these records exist, and if your database management policy permits, consider defining new areas that have enough CI size, and moving the large database records to the new areas. The new randomizer might need to be a two-stage randomizer because the randomizer must determine the area from a given root sequence key value before calculating a RAP number.

For details, see [Chapter 12, “Tuning DEDB areas,”](#) on page 119.

Planning free space reorganization and usage monitoring

When you identify free space fragmentation in the area, consider reorganizing free space. Also, consider reviewing the reorganization schedule for that area if exceptions that are related to free space usage or overflow usage were reported earlier than you expected.

About this task

In this task, you will learn how to reorganize a DEDB area to resolve free space fragmentation. After reorganizing the area, you will determine the effectiveness of the reorganization.

You will also learn how the reports of IMS Fast Path Solution Pack utilities can help you evaluate whether the current reorganization schedule is appropriate for the application activities on the area.

Procedure

Complete the following steps to reorganize free space and review the reorganization schedule:

1. Reorganize free space.

When you identify free space fragmentation in the area, reorganize free space by using the Online Expert Reorganization (OER) utility, the IMS High-Speed Direct Reorganization (HSR) utility, or another free space reorganization program to remove free space fragmentation.

To use the OER utility, see "Reorganizing a DEDB online with FPO" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

2. Collect the utility reports for the reorganized area.

If you determine that exceptions still remain after reorganization, collect the utility reports again and analyze the area by using the new set of reports. Then, take the appropriate actions to remove the exceptions.

For information about detailed analysis, see [Chapter 10, “Understanding the status of DEDB areas,”](#) on page 97.

3. Review the reorganization schedule and ensure that the reorganization schedule is well-tuned for the area. The schedule must consider the timing when the area becomes severely fragmented.

When you review the free space reorganization schedule, ensure that the scheduled free space reorganization can prevent degradation of application performance that might be caused by excessive CI accesses to retrieve a target segment and to prevent application failures that might be caused by insufficient space.

To understand when and how often fragmentation can occur, use either of the following methods:

- Generate a UOW Exception report by running the Analyze function of the Fast Path Advanced Tool. This report helps you identify the UOWs that might have fragmented free space.

- Use the OER utility. You can easily identify the existence of highly fragmented UOWs and how well the online reorganization can resolve the free space fragmentation in the UOW or reduce overflow to IOVF or DOVF from the RAP CIs. Consider running the OER utility periodically for the areas that can become fragmented by application activities.

If you determine that the free space in the area can become fragmented earlier than predicted, consider changing the reorganization schedule so that data overflow to IOVF can be prevented.

Tip: Changing the reorganization cycle might be difficult in certain environments. Alternatively, you can change the CI size, UOW definition, or the randomizer so that more free space can be reserved in RAA BASE. These tasks are referred to as tuning activities and are described in [Chapter 12, “Tuning DEDB areas,”](#) on page 119.

For more information, see the following resources:

- For information about reading the UOW Exception reports, see [“Analyzing DEDB areas at the UOW level”](#) on page 112.
- For information about generating the UOW Exception report, see the topic "Generating the analysis reports" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Chapter 12. Tuning DEDB areas

Tuning DEDB areas mainly involves changing the DBD parameters and randomizers. When exceptionally large database records exist in the areas, additional preparation steps might be necessary.

Basic guidelines

The following basic guidelines generally apply when you tune areas in which exceptions are reported:

- You should configure the DBD and the randomizing routine to optimize utilization of RAP CIs and to minimize the overflow to IOVF. Select an optimal CI size and optimal parameter values for the UOW and ROOT parameters at DBDGEN, and tune the randomizing routine so that the database record distribution can be optimized for the distribution of root key values and the distribution of database record lengths.
- You should select database attributes to minimize the physical I/O requirements to meet the application performance criteria.

Optional: Customizing the randomizer for exceptionally large database records

Tuning areas that contain exceptionally large database records can be difficult.

Before you begin

In this task, you will move large database records to new areas to simplify the tuning tasks. Before you start, review the applications that use the database and ensure that those applications are not affected by moving such records to the new areas.

About this task

If your administration policies allow, one possible approach to tuning areas that contain exceptionally large database records is to define new areas that have enough CI size, and to move these large records to the new areas before tuning the areas.

Procedure

Complete the following steps to define new areas and to move large database records to the new areas:

1. Determine whether you can move large database records to different areas. Then obtain the information about the large database records to move.

Generate a Largest Database Records report and browse the report. This report provides a list of database records and the information about these records. Analyze the information in the report to determine whether you can move those records or a subset of those records to a new area that matches the record characteristics.

2. Customize the randomizing module for the selected database records and determine the DBD statements for the new areas.

You must ensure that the selected large database records be randomized into the new areas, which means that you must customize the randomizing routine for the database. In general, the randomizing routine that is used needs to be a *two-stage randomizer*. In this context, a two-stage randomizer determines an area from the root sequence key value and then determines a RAP number within the selected area.

To customize or tune the randomizing routine, you need at least the root sequence key values for the selected database records and information about the database record length for each key. You can obtain this information from the Largest Database Records report.

When you customize the randomizing routine, be aware that the root segments of those large database records can be randomized on different RAPs in new areas.

For the information that you can obtain from the Largest Database Records report, see Step [“5”](#) on page 107 of the topic [“Analyzing database record profile”](#) on page 105.

What to do next

Follow the steps in [“Tuning randomizer and DBD parameters for DEDB”](#) on page 120 to tune the areas.

Tuning randomizer and DBD parameters for DEDB

Confirm that necessary information has been collected, tune the area, and verify the result of tuning.

Before you begin

Confirm that you have completed the following tasks:

- You have reviewed the free space utilization by completing the steps in [“Analyzing free space usage”](#) on page 100.
- You have reviewed the existing database record characteristics by completing the steps in [“Analyzing database record profile”](#) on page 105.
- You have reviewed the current randomizing statistics by completing the steps in [“Analyzing the randomizing routine”](#) on page 109.

Several key indicators are used to measure the overall health of an area. You have already checked some of the key indicators in the analysis reports that you used when you analyzed the area. The key indicators are also stored in the Sensor Data repository as sensor data of the area. Before you start this task, ensure that you can access the Sensor Data Statistics report that contains the latest sensor data. If you cannot access the report, run the FP DB Sensor Printing utility to obtain the report.

If you have decided to tune the randomizer, information about the characteristics of root sequence keys of database records, key values of all current database records, and additional key values for database records that are predicted to be added in near future can also help you tune the randomizing routine.

About this task

This task describes the steps for tuning the DBD parameters and the steps for tuning the randomizing routine. The objective is to adjust both the DBD parameters and the randomizing routine to suit the current characteristics of the database records in the area.

Specifically, the goal of this tuning task is to achieve the following results:

Shorter synonym chains with fewer unused RAPs

Generally, having zero synonyms is hard to achieve. Synonym chains are acceptable if they are short and they do not cause CI contention in the online environment.

A higher percentage of segments in RAA BASE

In a database where high performance is required, all root segments and direct dependent segments should reside in RAA BASE because any segments that are not in RAA BASE incur an extra physical I/O.

Fewer physical I/Os to access a root segment and a complete database record

Root I/O should be reduced to slightly greater than 1, and record I/O should be approximately 1.5.

For a high performance database, the maximum value for both root I/O and record I/O should be 1.00.

Procedure

Tuning of DEDB areas involves the tuning of randomizing modules, changing the DBD definition, and verifying the effectiveness of the tuned areas.

Tip: Consider using the Tune function of Fast Path Advanced Tool or the DEDB Tuning Aid utility of Fast Path Basic Tools. These utilities help you determine the effective database definition by evaluating the potential benefits, or impact, of changes to an existing area without the overhead of unloading and reloading the databases.

1. Optional: Collect the segment information records for use with the DEDB Tuning Aid utility.

This step is required only if you use the DEDB Tuning Aid utility. If you use the Tune function, you can skip this step.

To use the DEDB Tuning Aid utility, you must generate the segment information record data set for load simulation. Run the Online Pointer Checker (OPC) utility of Fast Path Online Tools by specifying the TYPRUN=MODEL option.

See the following topics in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*:

- For information about using the DEDB Tuning Aid utility, see the topic "Tuning a DEDB offline with FPB".
- For information about using the OPC utility, see the topic "Analyzing a DEDB online with FPO".
- For the TYPRUN=MODEL option of OPC, see the topic "TYPRUN keyword".
- For JCL examples to run OPC with TYPRUN=MODEL and to use DEDB Tuning Aid, see the topic "Example 5: Collecting segment information records for database tuning".

2. Refer to the DB Record Profile Analysis report to determine the average, maximum, and minimum record lengths (including the prefix lengths).

You can also obtain these values by viewing the Sensor Data Statistics report. Look at the values for data elements DB_AVG_DBREC_LENGTH, DB_MAX_DBREC_LENGTH, and DB_MIN_DBREC_LENGTH.

If average and maximum synonym chain lengths are long and CI contention can occur, consider expanding the RAA BASE section and verify that the number and the length of the synonym chains have been reduced.

To generate a DB Record Profile Analysis report, see [“Analyzing database record profile” on page 105](#).

3. Determine the appropriate CI size and appropriate values for the UOW and the ROOT parameters by evaluating following statistics values and user preferences:

- The total number of records (the DB_NUM_ROOT value in the Sensor Data Statistics report)
- Database record lengths (the three data element values that are mentioned in the previous step)
- Overflow usage (DOVF and IOVF)

Look at the values of the following data elements in the Sensor Data Statistics report:

- DB_PCT_BYTES_FS_DOVF
- DB_PCT_BYTES_FS_IOVF
- DB_PCT_NUM_RAPCI_OVFL
- DB_PCT_NUM_UOW_USE_DOVF
- DB_AVG_NUM_DOVFCI_BY_UOW
- DB_MAX_NUM_DOVFCI_BY_UOW
- DB_NUM_UOW_USE_IOVF
- DB_PCT_NUM_UOW_USE_IOVF
- DB_AVG_NUM_IOVFCI_BY_UOW
- DB_MAX_NUM_IOVFCI_BY_UOW
- DB_MIN_NUM_IOVFCI_BY_UOW
- DB_PCT_NUM_IOVFCI_USED
- Number of areas
- Distribution of root sequence keys and randomizing logic

- An optimal packing density. The density of 60% is reasonable as a general rule. However, you must consider future application activities and the database tuning cycle when you determine the optimal packing density. Examine at the DB_PCT_BYTES_FS_RAA value in the Sensor Data Statistics report to calculate the current packing density.
- The preferred percentage of records that should not extend into IOVF. Look at the DB_PCT_NUM_DBREC_IOVF value in the Sensor Data Statistics report to see the current percentage.
- The current values for CI size, UOW parameters, and ROOT parameters. You can obtain these values by examining the following data element values in the Sensor Data Statistics report:
 - DB_AREADEF_CISIZE
 - DB_AREADEF_UOW1
 - DB_AREADEF_UOW2
 - DB_AREADEF_ROOT1
 - DB_AREADEF_ROOT2

To determine appropriate parameter settings, consider the following factors:

- A DEDB area that has a high percentage of large database records can experience performance problems that are caused by a large number of database record I/Os. Tuning the randomizer or DBD parameters for such an area might be difficult. To alleviate these difficulties, consider these possible actions:
 - Use a larger CI size for the area.
 - Evaluate the usage of database segment compression to reduce the average size of database records.

Using a compression exit routine to store the segments on DASD in a compressed format can be useful, especially if the compression can significantly reduce the average database record size in such a way that the average number of database record I/Os can be reduced.

When you use compression routines, be careful about segment overflow that can be caused by replace calls that extend the compressed size of a segment.

- The randomizing module must be tuned to accommodate the new UOW geometry and the new total number of RAPs. You must also consider the current and the future distribution of root sequence keys and database record lengths.
- UOW size can be large or small, but the number of CIs that are reserved for DOVF should not exceed 1 (that is, DOVF should never be used). Similarly, the number of UOWs that are reserved for IOVF should not exceed 1.

4. Optional: Run the Tune function or the DEDB Tuning Aid utility.

Use the Tune function or the DEDB Tuning Aid utility to simulate a reload for the area applying new DBD definitions or randomizer changes. By using either of the utilities, you can determine the effects of the changes without the overhead of unloading and reloading the databases.

- With the Tune function, you can generate the Tuning Aid record data sets, which can be reused in the subsequent Tune jobs. By reusing the Tuning Aid records, you can run the Tune function faster and simulate various parameters with iterative runs with less amount of time. For more information about the Tune function, see the topic "Tuning a DEDB offline with FPA" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.
- With the DEDB Tuning Aid utility, you can use the segment information record data set that you prepared in Step "1" on page 121 as the input to the utility.

Repeat this tuning activity until you determine the optimal changes.

5. Unload and reload the area by using the new UOW definition and the new randomizing module.

You can use the Unload and the Reload functions of FPA. If you have defined additional areas in Step "3" on page 121, you can use the FPA Unload function to unload the current area into multiple unloaded segment records (USR) data sets, which means that you can create one data set for each

new area, and apply the new area definitions and new randomizing logic. Then, you can use the FPA Reload function to reload all new areas in one job step by specifying these USR data sets as input.

Tip: FPA Reload can generate analysis reports and other relevant reports that contain information about the reloaded areas. FPA Reload can also store sensor data for the reloaded areas in the Sensor Data repository and generate the Sensor Data Statistics report for the reloaded area. By comparing the values in this report with the values in the old Sensor Data Statistics report, you can easily identify how the health of the area has changed.

For instructions for running the FPA Unload and the Reload functions, see the topic "Scenario: Restructuring DEDB areas" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Notices

This information was developed for products and services offered in the U.S.A.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licenseses of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119

Armonk, NY 10504-1785
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and the section titled "Cookies, Web Beacons, and Other Technologies" in IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details>. Also, see the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Index

A

accessibility
overview [10](#)

C

cookie policy [125](#)

D

DBD parameters
DEDB areas [120](#)
DEDB areas
database record profile [105](#)
free space usage [100](#)
randomizing [109](#)
UOWs [112](#)
documentation
accessing [9](#)
sending feedback [9](#)
documentation changes [3](#)

E

environment [5](#)
exception categories
DEDB areas [97](#)
full-function databases [18](#)
exceptions
fragmentation
HDAM [37](#)
HIDAM [63](#)
HISAM [77](#)
indexes [87](#)
PHDAM [37](#)
PHIDAM [63](#)
SHISAM [77](#)
free space availability
DEDB areas [97](#)
free space relationship
DEDB areas [97](#)
optimization
HDAM [41](#)
HIDAM [67](#)
HISAM [80](#)
indexes [88](#)
PHDAM [41](#)
PHIDAM [67](#)
SHISAM [80](#)
overflow usage
DEDB areas [97](#)
Record I/O
DEDB areas [97](#)
Root I/O
DEDB areas [97](#)

exceptions (*continued*)

space use
HDAM [29](#)
HIDAM [29](#)
HISAM [29](#)
indexes [29](#)
PHDAM [29](#)
PHIDAM [29](#)
SHISAM [29](#)
Synonym chain
DEDB areas [97](#)

F

fragmentation exceptions
HDAM [37](#)
HIDAM [63](#)
HISAM [77](#)
indexes [87](#)
PHDAM [37](#)
PHIDAM [63](#)
SHISAM [77](#)
free space
full-function databases [21](#)
free space reorganization
DEDB areas [116](#)

H

HALDB partitioned secondary indexes [87](#)
HDAM [37](#)
HIDAM [63](#)
HIDAM primary indexes [87](#)
HIDAM secondary indexes [87](#)
HISAM [77](#)
home block [48](#)

I

IMS free space
full-function databases [21](#)
index databases [87](#)

L

large database records
DEDB areas [119](#)
legal notices
cookie policy [125](#)
notices [125](#)
programming interface information [125](#)
trademarks [125](#)

N

notices [125](#)

O

optimization exceptions

- [HDAM 41](#)
- [HIDAM 67](#)
- [HISAM 80](#)
- [indexes 88](#)
- [PHDAM 41](#)
- [PHIDAM 67](#)
- [SHISAM 80](#)

overview [3, 4](#)

P

[PHDAM 37](#)

[PHIDAM 63](#)

prerequisite knowledge [9](#)

prerequisite publications [9](#)

programming interface information [125](#)

R

randomizer

- [DEDB areas 120](#)

RAP chain [52, 53](#)

reader comment form [9](#)

resolving exceptions

- [DEDB areas 115](#)

fragmentation

- [HDAM 46](#)

- [HIDAM 71](#)

- [HISAM 83](#)

- [indexes 90](#)

- [PHDAM 46](#)

- [PHIDAM 71](#)

- [PHIDAM primary indexes 75](#)

- [SHISAM 83](#)

optimization

- [HDAM 47](#)

- [HIDAM 71](#)

- [HISAM 83](#)

- [indexes 91](#)

- [PHDAM 47](#)

- [PHIDAM 71](#)

- [SHISAM 83](#)

space use

- [HDAM 45](#)

- [HIDAM 69](#)

- [HISAM 82](#)

- [indexes 89](#)

- [PHDAM 45](#)

- [PHIDAM 69](#)

- [PHIDAM primary indexes 74](#)

- [SHISAM 82](#)

S

scenario

- [full-function databases 19](#)

screen readers and magnifiers [10](#)

[SHISAM 77](#)

space use

- [full-function databases 21](#)

space use exceptions [29](#)

summary of changes [3](#)

T

technotes [9](#)

terminology [7](#)

trademarks [125](#)

tuning

- [DEDB areas 119](#)

- [full-function databases 17](#)

W

what's new [3](#)



SC19-3395-03

