IBM IMS Database Utility Solution for z/OS
2.1

Overview and Customization

Note:

Before using this information and the product it supports, read the information in "Notices" on page 81.

Third Edition (July 2022)

This edition applies to Version 2.1 of IBM IMS Database Utility Solution for z/OS (program number 5698-DUL) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC27-8778-01.

Contents

About this information

IBM® IMS Database Utility Solution for z/OS® (also referred to as IMS Database Utility Solution) is a product that combines a number of IBM products into a consolidated solution that provides a set of efficient management utilities for IMS full-function databases and High Availability Large Databases (HALDBs).

IMS Database Utility Solution combines the following products into a single offering:

- IBM IMS Database Reorganization Expert for z/OS
- IBM IMS Database Utility Solution for z/OS IMS High Availability Large Database (HALDB) Toolkit
- IBM IMS High Performance Unload for z/OS
- IBM IMS High Performance Load for z/OS
- IBM IMS Index Builder for z/OS
- IBM IMS High Performance Image Copy for z/OS
- IBM IMS High Performance Pointer Checker for z/OS
- IBM IMS Library Integrity Utilities for z/OS

IMS Database Utility Solution also includes Database Repair Facility and Database Sensor.

These topics provide an overview of the products and capabilities delivered with IMS Database Utility Solution. It also includes post-SMP/E installation instructions and customization instructions which must be performed before these products can be used.

These topics are designed to help database administrators, system programmers, application programmers, and system operators perform the following tasks:

- Understand the capabilities of the functions associated with IMS Database Utility Solution tools
- Plan for the installation of IMS Database Utility Solution
- Migrate from previous releases of the tools that comprise IMS Database Utility Solution
- Perform the post-SMP/E installation of IMS Database Utility Solution
- Customize IMS Database Utility Solution tools

Before using this product, you should understand basic IMS concepts.

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

Chapter 1. IMS Database Utility Solution for z/OS overview

IBM IMS Database Utility Solution for z/OS (also referred to as IMS Database Utility Solution) combines data management tools to provide the capabilities you need to manage your IMS full-function databases and High Availability Large Databases (HALDBs).

IMS Database Utility Solution is specifically designed to provide database administrators (DBAs) with smart solutions to database operation and maintenance tasks.

Topics:

What's new in IMS Database Utility Solution?

In addition to the tools and functions included IMS Database Utility Solution1.1, IMS Database Utility Solution 2.1 consolidates IBM IMS High Performance Pointer Checker for z/OS, IBM IMS Library Integrity Utilities for z/OS, and Database Repair Facility.

New IMS Tools products in IMS Database Utility Solution 2.1

IBM IMS High Performance Pointer Checker for z/OS
IBM IMS High Performance Pointer Checker for z/OS (also referred to as IMS HP Pointer Checker) provides a number of utilities designed to help ensure IMS databases are operational, tuned, repaired, and ready for use.

IBM IMS Library Integrity Utilities for z/OS
IBM IMS Library Integrity Utilities for z/OS (also referred to as IMS Library Integrity Utilities) provides a set of utilities designed to validate, compare, map, recover, report, and regenerate a number of IMS libraries needed for operations.

Database Repair Facility
Database Repair Facility is a powerful tool to repair VSAM and OSAM organized IMS databases that contain pointer or data errors. It can be run both interactively for VSAM and OSAM data sets or in batch mode for VSAM only data sets that are either IMS and non IMS data sets.

New functions and enhancements

This section summarizes some of the major new functions and features that IMS Database Utility Solution 2.1 provides. For a complete list of new function APARs, see IMS Database Utility Solution 2.1 new functions in IBM Documentation.

IMS 15 support
All of the tools in IMS Database Utility Solution support IMS 15.

IMS management of ACBs support
Many of the tools in IMS Database Utility Solution support the use of ACBs managed by IMS. If the IMS management of ACBs is enabled, the tools that support this function can read database

1

definitions from the IMS catalog instead of from DBD libraries. For details, see IBM technote 0731745: IBM IMS Tools and Managed ACB Support.

Data set encryption support
Many of the tools in IMS Database Utility Solution support data set encryption for IMS data sets. For details, see IBM technote 0733513: IBM IMS Tools and Data Set Encryption Support.

Documentation changes

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.

SC27-8778-02 (July 2022)

Description	Related APARs
Documentation updates to support IBM IMS Tools Base for z/OS 1.7 and IMS Administration Foundation. IMS Administration Foundation activates the IMS administration web-browser interface of IBM Unified Management Server for z/OS to enable the management of IMS systems and resources.	N/A

SC27-8778-01 (July 2021)

Description	Related APARs
Updated the following topics to support IMS Library Integrity Utilities enhancement: • "What is IMS Database Utility Solution?" on page 2 • "Benefits" on page 5 • "LICON data sets and global option modules" on page 47 • "Activating Integrity Checker" on page 58 • "Creating an RDE to register DMB information" on page 60	PH24779
"Configuring IMS Tools Base" on page 25 is updated.	N/A

What is IMS Database Utility Solution?

IMS Database Utility Solution combines IMS Tools products into a single solution to provide an effective and efficient approach for managing your IMS full-function databases and HALDBs quickly and accurately.

The following tools are included in IMS Database Utility Solution but are shipped in separate target and distribution libraries:

- "IBM IMS Database Reorganization Expert for z/OS" on page 3
- "IBM IMS High Performance Unload for z/OS" on page 3
- "IBM IMS High Performance Load for z/OS" on page 3
- "IBM IMS Index Builder for z/OS" on page 3

- "IBM IMS High Performance Image Copy for z/OS" on page 4
- "IBM IMS High Performance Pointer Checker for z/OS" on page 4
- "IBM IMS Library Integrity Utilities for z/OS" on page 4
- "IMS High Availability Large Database Toolkit" on page 4

IMS Database Utility Solution also provides the following functions:

- "Database Sensor" on page 5
- "Database Repair Facility" on page 5

IBM IMS Database Reorganization Expert for z/OS

IBM IMS Database Reorganization Expert for z/OS (also referred to as IMS Database Reorganization Expert) facilitates your reorganization scenario by releasing you from the complex database diagnosis tasks that are needed to acknowledge the database health.

When a database is diagnosed to be in need of reorganization, the product automatically runs the reorganization job, and you will receive a well-summarized report on the database status. By using IMS Database Reorganization Expert, you can reduce the amount of work and expertise that is required for database reorganization tasks and realize higher productivity with less maintenance time and training costs.

To learn more about IMS Database Reorganization Expert, see the *IMS Database Reorganization Expert User's Guide*.

IBM IMS High Performance Unload for z/OS

IBM IMS High Performance Unload for z/OS (also referred to as IMS HP Unload) provides high speed unloading of IMS databases and improves performance of IMS data retrieval application programs by using the Unload application programming interface (API).

As processing volumes increase, more work needs to be done in a shorter period of time due to shrinking batch windows. The tool saves your time and money by reducing the CPU and elapsed time required for IMS database unloading and for running IMS data retrieval application programs. Powerful functions such as the ability to continue processing after a pointer error, a user exit facility, and a variety of unloaded record formats are provided—all with the goal of improving availability and throughput.

To learn more about IMS HP Unload, see the *IMS High Performance Unload User's Guide*.

IBM IMS High Performance Load for z/OS

IBM IMS High Performance Load for z/OS (also referred to as IMS HP Load) provides high-speed loading of IMS databases and improves the performance of IMS database initial loading application programs by using the Load Application Programming Interface (API).

As processing volumes increase, more work needs to be done in a shorter period of time due to shrinking batch windows. This tool saves you time and money by reducing the CPU and elapsed time that is required for IMS database loading and reorganization.

To learn more about IMS HP Load, see the *IMS High Performance Load User's Guide*.

IBM IMS Index Builder for z/OS

IBM IMS Index Builder for z/OS (also referred to as IMS Index Builder) provides improved technology for the recovery, maintenance, and addition of IMS indexes and eliminates the need to image copy index data sets for recovery and backup purposes.

IMS Index Builder allows you to rebuild IMS indexes rather than recover or reorganize them using the traditional unload and reload processes used for the primary data store. This eliminates the need to image copy indexes, means faster recovery and reorganization times, and allows for adding new indexes quickly without the need to reorganize the primary databases. IMS Index Builder provides an easy-to-use,

one-step procedure for building all IMS index types including primary, secondary, and indirect list data sets.

To learn more about IMS Index Builder, see the *IMS Index Builder User's Guide*.

IBM IMS High Performance Image Copy for z/OS

IBM IMS High Performance Image Copy for z/OS (also referred to as IMS HP Image Copy) enables you to quickly back up and recover database data sets by providing support for advanced copying features.

Using image copies to create backup data is essential to the timely recovery of databases after a loss of data or a programming error. However, taking data offline can reduce user productivity and negatively impact your business. IMS HP Image Copy provides rapid image copy features that reduce the amount of time that database data sets are unavailable. These features enable automated operation that takes the database offline before taking a batch image copy and restarts it after the process. IMS HP Image Copy helps you speed recovery time by supporting quick copy and restarting methods, and allows your users to become more productive and avoid expensive losses from missed business opportunities.

To learn more about IMS HP Image Copy, see the *IMS High Performance Image Copy User's Guide*.

IBM IMS High Performance Pointer Checker for z/OS

IBM IMS High Performance Pointer Checker for z/OS (also referred to as IMS HP Pointer Checker) provides a number of utilities designed to help ensure IMS databases are operational, tuned, repaired, and ready for use.

The product reports space utilization, and detects and reports physical and logical direct pointer problems in IMS full-function databases. These reports pinpoint both the errors and their locations within the database. Data is also collected for analysis and report by several utilities provided by this product.

To learn more about IMS HP Pointer Checker, see the *IMS High Performance Pointer Checker User's Guide*.

IBM IMS Library Integrity Utilities for z/OS

IBM IMS Library Integrity Utilities for z/OS (also referred to as IMS Library Integrity Utilities) provides a set of utilities designed to validate, compare, map, recover, report, and regenerate a number of IMS libraries needed for operations.

Supported libraries include the ACB (application control block), DBD (database description), PSB (program specification block), MFS (Message Format Service) libraries, libraries containing DFSMDA members, IMS catalog, IMS directory, and RECON (recovery control) data sets. The utilities are used in conjunction with database maintenance tasks and migration operations, and to prevent accidental misuse in production.

To learn more about IMS Library Integrity Utilities, see the *IMS Library Integrity Utilities User's Guide*.

IMS Library Integrity Utilities also provides the DBD/PSB Map Viewer, which is an IMS Library Integrity Utilities extension for Management Console and IMS Administration Foundation. This extension enables the DBD/PSB Map Viewer for Management Console and the DBD/PSB Map feature for IMS Administration Foundation. The DBD/PSB Map Viewer and the DBD/PSB Map feature display graphical visualization of a database structure map, program specifications, DBD and PSB macro source, and, for Management Console, DBD and PSB XML documents. You can also use the cross reference feature to list logical DBDs and PSBs that refer to a specific DBD.

IMS High Availability Large Database Toolkit

IMS High Availability Large Database Toolkit (also referred to as IMS HALDB Toolkit) provides High Availability Large Database (HALDB) functions for HALDB maintenance and operation tasks.

Features of IMS HALDB Toolkit include application enabling functions and system utilities to make better use of the HALDB environment. IMS HALDB Toolkit can help reduce the time and minimize the skills that

are required to perform application support tasks and provide IMS HALDB maintenance, modeling, and analysis.

To learn more about IMS HALDB Toolkit, see the *IMS Database Utility Solution IMS HALDB Toolkit User's Guide.*

Database Sensor

Database Sensor collects statistics about IMS databases and optionally stores them as *sensor data* in the Sensor Data repository of IMS Tools Knowledge Base. Sensor data can be used for monitoring and maintaining the health, performance, and recoverability of the database.

To learn more about Database Sensor, see the *IMS Solution Packs Data Sensor User's Guide.*

Database Repair Facility

Database Repair Facility is a powerful tool to repair VSAM and OSAM organized IMS databases that contain pointer or data errors. It can be run both interactively for VSAM and OSAM data sets or in batch mode for VSAM only data sets that are either IMS and non IMS data sets.

The interactive capabilities of Database Repair Facility let you make the repair in a short time, which means that the affected database need be taken offline only for a brief period. In this way, Database Repair Facility enhances not only the integrity but also the availability of IMS databases.

To learn more about Database Repair Facility, see the *IMS Database Repair Facility for IMS Solution Packs User's Guide.*

Benefits

The tools of IMS Database Utility Solution help you simplify and accelerate the database maintenance tasks for IMS full-function databases and HALDBs.

IMS Database Utility Solution provides a set of tools that help IMS DBAs in every phase of database management. The tools are high performance, easy-to-use, and powerful, and include capabilities for database reorganization, data backup, tuning, monitoring, and optimization of IMS databases.

IMS Database Utility Solution provides the following benefits:

- The high performance reorganization tool can save time and money by reducing the CPU and elapsed time required for IMS database reorganization tasks, and for the IMS database retrieval and initial loading application programs by providing the APIs.
- IMS Database Reorganization Expert further reduces time and resource requirements by enabling unload, load, index build, and backup tasks to run concurrently, and by eliminating I/Os for intermediate data sets. It also enhances the productivity of DBAs by making the reorganization processes simpler and easier, and by minimizing manual interventions.
- IMS Database Reorganization Expert saves online system resources by evaluating the database with user-specified criteria, and running the reorganization job only when reorganization is determined to be effective for the database.
- IMS Index Builder eliminates the need to image copy and log updates to IMS index data sets by providing a quick and efficient method for rebuilding indexes. It also provides the ability to add new indexes to existing IMS databases as needed.
- IMS HP Image Copy reduces the unavailability time of databases by providing rapid image copy features, parallel processing, and minimized manual interventions. It also reduces operational and media costs by compressing and stacking the image copies.
- IMS HP Pointer Checker helps ensure that your database pointers are error free. Used in conjunction with Database Repair Facility, it helps you detect and correct database errors quickly and repair them with minimum downtime to your IMS environment.
- IMS Library Integrity Utilities enhances the productivity of DBAs by providing utilities that effectively manage IMS ACB, DBD, PSB, MFS libraries, libraries containing DFSMDA members, IMS catalog, IMS

directory, and RECON data sets in database maintenance tasks and migration operations. It also avoids application downtime by preventing database corruption that might occur from accidental misuse of IMS libraries in production.

- Provides a graphical view of IMS databases and program specifications through Management Console and IMS Administration Foundation. Database structures, program specifications, and source code can be reviewed to simplify database administration tasks. The graphical interface can be used as an educational tool for new DBAs to learn about databases.
- With the automated sensor data collection and exception detection that are provided by Autonomics Director and IMS Database Utility Solution, you can save time and resources in monitoring and tuning IMS full-function databases.
- With the graphical web interface that is provided by IMS Administration Foundation and Management Console, you can analyze your IMS environment to rapidly identify database exceptions and tuning opportunities.
- IMS HALDB Toolkit efficiently diagnoses HALDB structures and suggests effective database structures.

Installation requirements

The tools that are provided in IMS Database Utility Solution have dependencies on IMS and require the tools provided in IBM IMS Tools Base for z/OS (also referred to as IMS Tools Base).

Installation requirements

IMS Database Utility Solution requires z/OS 2.3 (5650-ZOS) or later.

In addition, IMS Database Utility Solution requires IBM IMS Tools Base for z/OS 1.6 (5655-V93) or later. IBM IMS Tools Base for z/OS must be installed into the same SMP/E CSI zone as IMS Database Utility Solution.

Operational requirements

An operational requisite is defined as a product that is required and must be present or a product that is not required but should be present on the system in order for this product to operate all or some of its functions.

IMS Database Utility Solution operates with the following IMS versions:

- IMS 15.1 (5635-A06) or later
- IMS Database Value Unit Edition 15.1 (5655-DS5) or later

To use IMS Database Reorganization Expert, IMS High Performance Load and IMS High Performance Unload must be installed.

To use IMS High Performance Pointer Checker, Database Repair Facility must be installed.

Common tool requirements

The following common tools, which are provided in IMS Tools Base, are used by the tools of IMS Database Utility Solution:

- Autonomics Director
- Distributed Access Infrastructure
- IMS Tools Common Services
- IMS Tools Knowledge Base
- Policy Services

These common tools are not distributed with IMS Database Utility Solution but are instead distributed with IMS Tools Base. IMS Tools Base must be installed and configured before IMS Database Utility Solution can be used.

IMS Tools Base 1.7 and later also provide IMS Administration Foundation, which is a browser-based graphical user interface that extends the capabilities of IBM Unified Management Server for z/OS to manage IMS systems and resources. To enable the graphical user interface of IMS Administration Foundation, the following products must be installed:

- IBM Unified Management Server for z/OS 1.1 with APAR PH47708 or later, and its prerequisite software
- IBM IMS Tools Base for z/OS 1.7 or later

If you plan to use the DBD/PSB Map Viewer in Management Console, which is an IMS Library Integrity Utilities extension that provides a graphical view of IMS databases and program specifications, the following products must be installed:

- IBM Installation Manager for z/OS 1.8 or later or IBM Installation Manager for Windows 1.8 or later
- IBM Management Console for IMS and Db2® for z/OS

Conditional operational requirements

A conditional operational requisite identifies the following products that are not required for the basic function but are needed at run time for the tool to utilize specific functions.

Table 1. Conditional operational requirements

Tool	Function	Required product
Database Sensor	Stand-alone Database Sensor function	IMS Database Reorganization Expert (with APAR PI94832)
Database Repair Facility	Process through TSO terminals	The Interactive System Productivity Facility (ISPF), which is included in z/OS
IMS HALDB Toolkit	Any functions of HALDB Toolkit	The DFSORT program, which is included in z/OS, or a functionally equivalent sort program
	Enable all the functions of IMS HALDB Toolkit	• IMS High Performance Load • IMS High Performance Image Copy • IMS High Performance Pointer Checker • IMS Library Integrity Utilities
	Process through TSO terminals	The Interactive System Productivity Facility (ISPF), which is included in z/OS
IMS Database Reorganization Expert	Enable IMS Database Reorganization Expert	• IMS Index Builder • IMS High Performance Image Copy • IMS High Performance Pointer Checker • IMS Library Integrity Utilities
IMS HP Image Copy	HDPC Hash Pointer Check function	IMS High Performance Pointer Checker
	DEDB Hash Pointer Check function	IMS Fast Path Solution Pack
	Integrated Database Sensor function	IMS Database Reorganization Expert

Table 1. Conditional operational requirements (continued)

Tool	Function	Required product
IMS HP Load	PSSR function	The DFSORT program, which is included in z/OS, or a functionally equivalent sort program
	Language Environment® interface for user exit routines written in COBOL or PL/I	Language Environment (LE), which is included in z/OS
IMS HP Pointer Checker	• HD Pointer Checker utility • HD Tuning Aid utility • Space Monitor utility	The DFSORT program, which is included in z/OS, or a functionally equivalent sort program
	DB Historical Data Analyzer utility (to display charts on TSO terminals)	The Interactive System Productivity Facility (ISPF) and the Graphical Data Display Manager (GDDM), which are included in z/OS
	DBD Decode and DBD Mapping options of HD Pointer Checker utility	IMS Library Integrity Utilities
	Integrated Database Sensor function	IMS Database Reorganization Expert
IMS HP Unload	Language Environment interface for user exit routines written in COBOL or PL/I	Language Environment (LE), which is included in z/OS
IMS Index Builder	Index build processing	The DFSORT program, which is included in z/OS, or a functionally equivalent sort program
IMS Library Integrity Utilities	• ACBLIB Analyzer utility • Integrity Checker utility	The DFSORT program, which is included in z/OS, or a functionally equivalent sort program
	DBD/PSB Map feature	Unified Management Server
	DBD/PSB Map Viewer	Management Console for IMS and Db2

Service updates and support information

Service updates and support information for this product, including software fix packs, PTFs, frequently asked questions (FAQs), technical notes, troubleshooting information, and downloads, are available from the web.

To find service updates and support information, see the following website:

IBM Support: IMS Database Utility Solution for z/OS

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/

Product publications

For detailed information about tools and components of IMS Database Utility Solution and related products, refer to the publication libraries of the individual products.

IMS Database Reorganization Expert for z/OS 4.1
  User's Guide, SC19-1137

IMS Database Repair Facility for IMS Solution Packs
  User's Guide, SC19-2916

IMS Database Utility Solution for z/OS 2.1 IMS HALDB Toolkit
  User's Guide, SC27-8779

IMS High Performance Image Copy for z/OS 4.2
  User's Guide, SC19-2756

IMS High Performance Load for z/OS 2.1
  User's Guide, SC18-9222

IMS High Performance Pointer Checker for z/OS 3.1
  User's Guide, SC19-2401

IMS High Performance Unload for z/OS 1.2
  User's Guide, SC27-0936

IMS Index Builder for z/OS 3.1
  User's Guide, SC18-9101

IMS Library Integrity Utilities for z/OS 2.2
  User's Guide, SC19-3979

IMS Solution Packs Data Sensor
  User's Guide, SC19-3283

IMS Solution Packs IMS Database Space Tuning Guide
  User's Guide, SC19-3395

All the information about these tools and common tools (IMS Tools Base, IMS Administration Foundation, and Management Console) can be found in IBM Documentation.

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 https://www.ibm.com/support/mynotifications
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.

Prerequisite knowledge

Before using this information, you must understand basic IMS concepts, the IMS environment, and your installation's IMS system. Therefore, IMS publications are prerequisite for all IMS Database Utility Solution products and components.

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 IMS Database Utility Solution enable users to perform the following activities:

- 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. Migration considerations

If you have one or more of the products that are included in IMS Database Utility Solution installed on your system or if you have an earlier release of IMS Database Utility Solution, complete the migration tasks.

This release of IMS Database Utility Solution includes the following tools:

- IBM IMS Database Reorganization Expert for z/OS 4.1
- IBM IMS High Performance Unload for z/OS 1.2
- IBM IMS High Performance Load for z/OS 2.1
- IBM IMS High Performance Image Copy for z/OS 4.2
- IBM IMS High Performance Pointer Checker for z/OS 3.1
- IBM IMS Index Builder for z/OS 3.1
- IBM IMS Library Integrity Utilities for z/OS 2.2

In addition, IMS Database Utility Solution includes the following components:

- IMS High Availability Large Database Toolkit
- Database Sensor
- Database Repair Facility

Product and component changes in IMS Database Utility Solution 2.1

Product and component changes in IMS Database Utility Solution 2.1 from IMS Database Utility Solution 1.1 are as follows:

- Added: IBM IMS High Performance Pointer Checker for z/OS 3.1
- Added: IBM IMS Library Integrity Utilities for z/OS 2.2
- Added: Database Repair Facility

Topics:

Migrating from IMS Database Utility Solution 1.1

If you have been using IMS Database Utility Solution 1.1, complete the following steps to migrate to IMS Database Utility Solution 2.1.

Procedure

1. Determine the SMP/E CSI zone to install IMS Database Utility Solution 2.1.

   IMS Database Utility Solution 2.1 can be installed in the same SMP/E CSI zone that is used for IMS Database Utility Solution 1.1 or in a different SMP/E CSI zone.

2. If you plan to install IMS Database Utility Solution 2.1 in the same SMP/E CSI zone, complete the following steps:

   a) Create a backup copy of all SMP/E data sets, target libraries, and distribution libraries.

      Installing IMS Database Utility Solution 2.1 into the same SMP/E CSI zone replaces existing IMS Database Utility Solution resources with 2.1 resources. Make sure you create a backup copy of all the data sets and libraries.

b) If you plan to reuse all the data sets that you have been using with IMS Database Utility Solution 1.1 in IMS Database Utility Solution 2.1 environment, ensure that all the required libraries, which are described in the *Program Directory for IBM IMS Database Utility Solution for z/OS 2.1*, GI13-4194, exist.

The required size of data sets in IMS Database Utility Solution 2.1 can be larger than the size required in version 1.1. See the *Program Directory for IBM IMS Database Utility Solution for z/OS 2.1*, GI13-4194, and determine the required size for the data sets. If the size of the data sets is insufficient, increase the size.

3. If any of the following started tasks or servers are running, stop it:

- IMS Tools Base: IMS Tools Knowledge Base
- IMS Tools Base: Autonomics Director (IAV*)
- IMS Tools Base: Distributed Access Infrastructure TCP server
- IMS Tools Base: Distributed Access Infrastructure TAS (Tools Access Server)
- IMS Tools Base: Distributed Access Infrastructure subordinate TAS (SOT)
- Management Console server
- UMS server (Unified Management Server instance)

4. Install the latest version of IBM IMS Tools Base for z/OS in the target SMP/E CSI zone. For SMP/E installation procedure, see the *Program Directory for IBM IMS Tools Base for z/OS*, GI10-8819.

IMS Database Utility Solution 2.1 requires IMS Tools Base for z/OS 1.6 or later. IMS Tools Base must be installed before IMS Database Utility Solution 2.1 can be installed.

5. Install IMS Database Utility Solution 2.1 by following the instructions in the *Program Directory for IBM IMS Database Utility Solution for z/OS 2.1*, GI13-4194.

If you are installing IMS Database Utility Solution 2.1 into the same SMP/E CSI zone as where the earlier release of IMS Database Utility Solution is installed and if you already have one or more of the following products installed from IMS Database Utility Solution 1.1, apply all maintenance for these products and ensure that they are at the latest maintenance level.

- IMS Database Reorganization Expert for z/OS 4.1
- IMS High Performance Unload for z/OS 1.2
- IMS High Performance Load for z/OS 2.1
- IMS High Performance Image Copy for z/OS 4.2
- IMS Index Builder for z/OS 3.1

Important: When you install IMS Database Utility Solution 2.1, you do not need to install the tools that you already installed in IMS Database Utility Solution 1.1. However, you must install IMS Database Utility Solution Database Sensor to replace with new modules.

6. Apply maintenance to all the tools and components of IMS Tools Base and IMS Tools that you use and make sure that they are at the latest maintenance level.

7. Migrate the IMS Tools Base components by using IMS Tools Setup.

Migration of IMS Tools Base components can be achieved by using IMS Tools Setup. IMS Tools Setup is a function that helps you quickly and efficiently perform the required post-SMP/E-installation customization process for IMS Tools solution pack products.

For instructions, see the topic "Migrating from IMS Tools Base 1.6 to IMS Tools Base 1.7" or "Migrating from Tools Base 1.5 to Tools Base 1.6" in the *IMS Tools Base Configuration Guide*. IMS Tools Setup also covers migration and customization of IMS Database Utility Solution products.

If you need to customize IMS Database Utility Solution products manually, follow the steps in Chapter 5, "Customizing the tools of IMS Database Utility Solution," on page 25 after you migrate IMS Tools Base components. Some of the customization steps that were done when you customized an earlier release of IMS Database Utility Solution can be bypassed. However, because IMS Database Utility

Solution 2.1 provides new functions and features, review all the customization steps and complete the steps that are required for your environment.

Migrating from individual products

If you have been using one or more of the products included in IMS Database Utility Solution, complete the following steps to migrate to the IMS Database Utility Solution environment.

Procedure

1. Determine the SMP/E CSI zone to install IMS Database Utility Solution.

   IMS Database Utility Solution can be installed in the same SMP/E CSI zone that is used for the existing product or in a different SMP/E CSI zone.

2. If you plan to install IMS Database Utility Solution in the same SMP/E CSI zone, create a backup copy of all SMP/E data sets, target libraries, and distribution libraries.

   Installing IMS Database Utility Solution 2.1 into the same SMP/E CSI zone replaces resources of the existing products with IMS Database Utility Solution 2.1 resources. Make sure you create a backup copy of all the data sets and libraries.

3. Install the latest version of IBM IMS Tools Base for z/OS in the same target SMP/E CSI zone. For SMP/E installation procedure, see the *Program Directory for IBM IMS Tools Base for z/OS*, GI10-8819.

   IMS Database Utility Solution requires IMS Tools Base for z/OS 1.6 or later. IMS Tools Base must be installed before IMS Database Utility Solution can be installed.

4. Install IMS Database Utility Solution 2.1 by following the instructions in the *Program Directory for IBM IMS Database Utility Solution for z/OS 2.1*, GI13-4194.

5. Apply maintenance to all the tools and components of IMS Tools Base and IMS Tools that you use and make sure that they are at the latest maintenance level.

6. Migrate the IMS Tools Base components by using IMS Tools Setup.

   The individual products that you have been using may previously required older versions of IMS Tools Knowledge Base, IMS Tools Online System Interface, and IMS Tools Generic Exits than what is required by this release of IMS Database Utility Solution. These common tools, along with several other tools are now delivered with IMS Tools Base.

   Migration of IMS Tools Base components can be achieved by using IMS Tools Setup. IMS Tools Setup is a function that helps you quickly and efficiently perform the required post-SMP/E-installation customization process for IMS Tools solution pack products.

   For instructions, see the topic "Migrating from IMS Tools Base 1.6 to IMS Tools Base 1.7" or "Migrating from Tools Base 1.5 to Tools Base 1.6" in the *IMS Tools Base Configuration Guide*. IMS Tools Setup also covers migration and customization of IMS Database Utility Solution products.

   All the components of IMS Tools Base are compatible with earlier releases of IMS Tools Knowledge Base, IMS Tools Online System Interface, and IMS Tools Generic Exits.

   If you need to customize IMS Database Utility Solution products manually, follow the steps in Chapter 5, "Customizing the tools of IMS Database Utility Solution," on page 25 after you migrate IMS Tools Base components. Some of the customization steps that were done when you customized individual IMS Tools can be bypassed. However, because IMS Database Utility Solution 2.1 provides new functions and features, review all the customization steps and complete the steps that are required for your environment.

Chapter 3. Installing IMS Database Utility Solution

Before you install IMS Database Utility Solution, you must first complete the SMP/E installation of IMS Tools Base.

Complete the SMP/E installation of the products in the following order:

Installing the required common tools from IMS Tools Base

Many of the tools included in IMS Database Utility Solution require the use of the common tools that are included in IMS Tools Base.

The following common tools integrate with the tools of IMS Database Utility Solution:

- Autonomics Director
- Distributed Access Infrastructure
- IMS Tools Common Services
- IMS Tools Knowledge Base
- Policy Services

To perform the SMP/E installation, see the *Program Directory for IBM IMS Tools Base for z/OS*, GI10-8819. This process allocates and downloads the target and distribution libraries and performs the SMP/E installation of the tools included in IMS Tools Base.

Optionally, if you are installing IMS Tools Base 1.7 or later, you can install IMS Administration Foundation, which activates the IMS administration web-browser interface of IBM Unified Management Server for z/OS to enable the management of IMS systems and resources. To install IMS Administration Foundation, install IBM Unified Management Server for z/OS by following the instructions in the *Program Directory for IBM Unified Management Server for z/OS*, GI13-5914, and then follow the instructions in the *Program Directory for IBM IMS Tools Base for z/OS*, GI10-8819.

After you install IMS Tools Base, continue with the installation of Management Console (optional) and IMS Database Utility Solution before customizing the tools included in IMS Tools Base.

Installing Management Console

If you plan to use the graphical web interface of Management Console, install IBM Management Console for IMS and Db2 for z/OS.

To perform the installation, see the *Program Directory for IBM Management Console for IMS and Db2 for z/OS*, GI10-9007.

While you install Management Console, make sure that you install the Management Console feature that includes capabilities for managing IMS environments. This feature is installed by the following parameter or option during installation:

- For z/OS: `imsbase.feature` parameter on the **install** command
- For Windows: **Management Console for IMS Management Base Feature** option

Installing the components of IMS Database Utility Solution

Install the IMS Database Utility Solution libraries provided to you by IBM.

The tools of IMS Database Utility Solution are shipped in separate target and distribution libraries.

To perform the SMP/E installation, see the *Program Directory for IBM IMS Database Utility Solution for z/OS 2.1*, GI13-4194.

The following table lists the data sets of the target libraries distributed in IMS Database Utility Solution.

Table 2. Data sets of the target libraries for IMS Database Utility Solution

Type	Name	Description	Local DSN name
CLIST	SFABCLIB	Executable CLIST members for Database Repair Facility	
	SHPSCLIB	Executable CLIST members for IMS HP Pointer Checker	
	SIHCSKEL	Executable CLIST members for IMS HALDB Toolkit	
LMOD	SFABLMD0	Executable load library for Database Repair Facility	
	SHPSLMD0	Executable load library for IMS Database Utility Solution	
	SIHCLOAD	Executable load library for IMS HALDB Toolkit	
	SIIULMOD	Executable load library for IMS Index Builder	
MAC	SHPSMAC0	Distributed product macros	
	SIHCMACS	Distributed product macro for IMS HALDB Toolkit	
MESSAGE	SFABMLIB	ISPF messages for Database Repair Facility	
	SHPSMLIB	ISPF messages for IMS HP Pointer Checker	
	SIHCMESG	ISPF messages for IMS HALDB Toolkit	
PANEL	SFABPLIB	ISPF panels for Database Repair Facility	
	SHPSPLIB	ISPF panels for IMS HP Pointer Checker	
	SIHCPANL	ISPF panels for IMS HALDB Toolkit	

Table 2. Data sets of the target libraries for IMS Database Utility Solution (continued)

Type	Name	Description	Local DSN name
SAMPLE	SDFABASE	SMP/E JCL members for installation	
	SHPSBASE	SMP/E JCL members for installing individual products	
	SHPSJCL0	IVP and customization JCL members	
	SHPSSAMP	Sample JCL members, exit codes, and cataloged procedures	
	SIIUBASE	SMP/E JCL members for installing IMS Index Builder	
	SIIUCNTL	Sample JCL members for IMS Index Builder	
	SIIUPROC	Sample cataloged procedures for IMS Index Builder	
	SIHCBASE	SMP/E JCL members for installation	
	SIHCSAMP	Sample JCL members for IMS HALDB Toolkit	
File system path	SHPSIHFS	File system path for IMS Library Integrity Utilities Maps to: *pathprefix*`/usr/lpp/ InstallationManagerRepository/ H27P220/IBM/`	

Gathering the data set names of IMS Database Utility Solution and IMS Tools Base

After you installed IMS Tools Base and IMS Database Utility Solution, record your data set names.

Fill in the following tables with the data set names for modifying the procedures and JCL members.

Subsections:

Data set names of IMS Database Utility Solution

IMS Database Utility Solution provides the libraries that are summarized in the following tables.

The SHPS*xxxx* libraries include the data sets for the following IMS Database Utility Solution tools and resources:

- IMS Database Reorganization Expert
- IMS HP Image Copy
- IMS HP Load
- IMS HP Pointer Checker

- IMS HP Unload
- IMS Library Integrity Utilities
- Database Sensor
- IMS Database Utility Solution macro library
- IMS Database Utility Solution sample JCL library

Table 3. Data set names of IMS Database Utility Solution resources (SHPSxxxx library)

Name	Description	Your data set name
SHPSLMD0	Executable load module library	
SDFABASE	SMP/E JCL members for installation	
SHPSBASE	SMP/E JCL members for installation	
SHPSJCL0	Sample JCL members	
SHPSSAMP	Sample cataloged procedures	
SHPSMAC0	Distributed product macros	
SHPSCLIB	Executable CLIST members	
SHPSMLIB	ISPF messages	
SHPSPLIB	ISPF panels	

The SIIU*xxxx* libraries include the data sets for IMS Index Builder.

Table 4. Data set names of IMS Index Builder (SIIUxxxx library)

Name	Description	Your data set name
SIIUBASE	SMP/E JCL members for installation	
SIIULMOD	Executable load module library for IMS Index Builder	
SIIUCNTL	Sample JCL members for IMS Index Builder	
SIIUPROC	Sample cataloged procedures for IMS Index Builder	

The SIHC*xxxx* libraries include the data sets for IMS HALDB Toolkit.

Table 5. Data set names of IMS HALDB Toolkit (SIHCxxxx library)

Name	Description	Your data set name
SIHCLOAD	Executable load module library for IMS HALDB Toolkit	
SIHCBASE	SMP/E JCL members for installation	
SIHCSKEL	Executable CLIST member for IMS HALDB Toolkit	
SIHCMESG	ISPF messages for IMS HALDB Toolkit	
SIHCPANL	ISPF panels for IMS HALDB Toolkit	

Name	Description	Your data set name
SIHCMACS	Distributed product macro for IMS HALDB Toolkit	
SIHCSAMP	Sample JCL members for IMS HALDB Toolkit	

The SFAB*xxxx* libraries include the data sets for Database Repair Facility.

Table 6. Data set names of Database Repair Facility (SFABxxxx library)

Name	Description	Your data set name
SFABLMD0	Executable load module library for Database Repair Facility	
SFABCLIB	Executable CLIST members for Database Repair Facility	
SFABMLIB	ISPF messages for Database Repair Facility	
SFABPLIB	ISPF panels for Database Repair Facility	

Data set names of IMS Tools Base

The following table summarizes the libraries of IMS Tools Base.

Table 7. Data set names of IMS Tools Base

Name	Description	Your data set name
SAIILINK	Executable load module library for Distributed Access Infrastructure	
SFOILOAD	Executable load module library for IMS Tools Online System Interface	
SGLXLOAD	Executable load module library for IMS Tools Generic Exits	
SHKTLOAD	Executable load module library for Autonomics Director, IMS Tools Knowledge Base, and Policy Services	
SHKTCEXE	Executable CLIST members for IMS Tools Knowledge Base	

Data set names of IMS libraries

The following table summarizes the IMS resources.

Table 8. Data set names of IMS resources

Name	Your data set name
IMS SDFSRESL	
IMS SDFSMAC	
IMS SMP/E CSI data set	

Table 8. Data set names of IMS resources (continued)

Name	Your data set name
IMS PROCLIB	
DBDLIB	
PSBLIB	
ACBLIB	
MDALIB	
USERLIB (see Note)	
RECON1	
RECON2	
RECON3	

Note: USERLIB is the load module library where your randomizer, segment compression exits, user exit routines, and other resources exist.

Data set names of other libraries

The following table summarizes other resources.

Table 9. Data set names of other resources

Name	Your data set name
GDDM (SADMMOD)	
ISPF (SISPLPA)	
ISPF (SISPLOAD)	
LE (SCEELKED)	
MVS™ PROCLIB	

Data set names of your program libraries

Use the following table to record the program libraries in which your user exit modules, site default modules, and base configuration modules are stored.

Table 10. Data set names of your program libraries

Description	Your data set name

Chapter 4. Initial product customization using IMS Tools Setup

IMS Tools Setup is a function that helps you quickly and efficiently perform the required post-SMP/E-installation customization process for IMS Tools solution pack products. IMS Tools Setup is provided by IBM IMS Tools Base.

What does IMS Tools Setup do?

After the selected IMS Tools solution pack products have been installed into SMP/E data sets, IMS Tools Setup provides a process to simplify the initial configuration that is required to begin using the products. IMS Tools Setup generates JCL members that you then submit as jobs, or perform as tasks, to complete the customization process.

IBM IMS Tools Base components are also configured and customized during the IMS Tools Setup process. IBM IMS Tools Base provides important supporting components and infrastructure that are required for the operation of many IMS Tools functions, such as storage repositories, autonomics, and interaction with IMS.

The goal of IMS Tools Setup is to greatly ease the time and effort it takes to have IMS Tools products up and running in your environment.

IMS Tools Setup is intended only for initial product installations, first-time users, and product evaluations. IMS Tools Setup is not intended for maintenance purposes.

IMS Tools products that can use IMS Tools Setup

The following IMS Tools products and solution packs can use IMS Tools Setup for initial configuration:

- IBM IMS Tools Base
- IBM IMS Database Solution Pack for z/OS
- IBM IMS Fast Path Solution Pack for z/OS
- IBM IMS Recovery Solution Pack for z/OS
- IBM IMS Database Utility Solution for z/OS
- IBM IMS Administration Tool for z/OS
- IBM IMS Cloning Tool for z/OS
- IBM IMS Program Restart Facility for z/OS

Starting IMS Tools Setup

The IMS Tools Setup function (HKTQSETU) can be found in IBM IMS Tools Base. You can start the function by running the following REXX EXEC:

```
EXEC 'smpehlq.SHKTCEXE(HKTQSETU)' 'HLQ(smpehlq)'
```

Note: *smpehlq* is the high-level qualifier for the IMS Tools Base SMP/E data sets.

The IMS Tools Setup ISPF panels provide an organized and logical approach to the customization tasks. The panels explain the operation and sequence of each member that is generated in the CUSTJCL data set. The correct JCL job and task operation order is very important.

Each panel contains embedded panel-context and individual field-context Help information. All information about using IMS Tools Setup is contained in the embedded Help. There is no separate user guide.

While using the IMS Tools Setup function to install IMS Tools products or to migrate your products from an earlier release to a later release, you need to specify the names of the libraries, or data sets, that are required for each IMS Tools product. Use the worksheets in the topic "Data set names for IMS Tools Setup" in the *IMS Tools Base Configuration Guide* to make a summary of the data set names that will be used in your environment.

Completing IMS Tools Setup

After you run the HKTQSETU REXX EXEC, you can refer to the $$READ member in the generated *hlq*.CUSTJCL data set to view summary information about the JCL members that were generated. Additionally, all individual *hlq*.CUSTJCL members contain detailed descriptions of the functions for each job.

Each of the generated JCL members begin with the number sign (#) and are named in the logical sequence of operation. Any members ending with the at sign (@) require manual steps. You must begin with the first *#xxxx* member and submit the JCL job or perform the task. After that job or task completes, you continue on to the next member and submit that JCL job or perform that task, and so forth.

The first few members are all SYSPROG related (APF, LPA, SSN, MVSPPT), followed by security related members, TCP/IP administration, DBA related members, and others.

You must process all members in the correct order to complete the full customization task properly.

Each JCL member has its own descriptive comment section that explains what the member does and which group it might belong to.

Chapter 5. Customizing the tools of IMS Database Utility Solution

Before using any of the tools that are included in IMS Database Utility Solution, many of the tools must be customized. Use the following topics to manually customize tools in IMS Database Utility Solution. These topics also cover customization of common tools that are included in IMS Tools Base for z/OS.

Customization checklist

Use the following checklist to customize the tools and components for IMS Database Utility Solution.

Table 11. Customization checklist

Status	Task
	1. "Configuring IMS Tools Base" on page 25
	2. "APF authorizing the program libraries" on page 27
	3. "Performing security-related tasks" on page 28
	4. "Obtaining XCF group names for IMS Tools Base servers" on page 35
	5. "Updating sample procedures and JCL" on page 35
	6. "Setting up Management Console" on page 39
	7. "Setting up Unified Management Server for IMS Administration Foundation" on page 44
	8. "Setting up IMS HALDB Toolkit" on page 44
	9. "Setting up the Integrity Checker utility of IMS Library Integrity Utilities" on page 46
	10. "Setting up the ISPF interfaces" on page 74

Configuring IMS Tools Base

Identify the IMS Tools Base components that will be required in your environment and configure the components.

About this task

Many of the tools included in IMS Database Utility Solution integrate with the components of IMS Tools Base. The following IMS Tools Base components might be used in your IMS Database Utility Solution environment.

Autonomics Director
Autonomics Director provides automation of IMS database monitoring based on an understanding of the current state of your IMS databases. Autonomics Director must be configured if you want Database Sensor to send a sensor data collection notification to Autonomics Director. Autonomic Director will use this notification as a trigger to start analyzing and generating recommendations for the database.

Distributed Access Infrastructure
Distributed Access Infrastructure acts as a gateway for communication between distributed platforms and z/OS. It operates in a z/OS environment and uses a TCP Server, the Tools Access Server (TAS), and the Subordinate TAS (SOT) to provide distributed clients access to IMS Tools through TCP/IP socket communication. Distributed Access Infrastructure is used for providing TCP gateway for various IMS Tools services that are provided by IMS Administration Foundation and Management Console. Those services include:

- Viewing exceptions monitored by using Autonomic Director monitor list
- Viewing sensor data history in trend charts
- Viewing IMS Tools utility reports
- Viewing sources and tree structures defined in DBDs and PSBs

Distributed Access Infrastructure must be configured to use IMS Administration Foundation or Management Console.

IMS Tools Common Services

IMS Tools Generic Exits provides the ability to define multiple IMS exits and enable them to be called within the IMS control region. One of the exits supported is the Partner Product exit routine (DFSPPUE0). This exit is required to use IMS Tools Online System Interface.

IMS Tools Online System Interface is a command interface that allows IMS Tools to interface with all supported versions of IMS.

IMS Tools Knowledge Base

IMS Tools Knowledge Base is the foundational infrastructure that provides a centralized information management environment for IMS Tools products and Policy Services. IMS Tools Knowledge Base allows you to store, manage, and access resources (such as reports, sensor data, policies, rules, and notification lists, and decoded DBDs) that are generated or used by any tool product that has been enabled and registered to participate in an IMS Tools Knowledge Base information management environment. IMS Tools Knowledge Base must be configured if you want to store reports and sensor data in central repositories, use autonomics capabilities provided by Autonomics Director, or use the graphical web interface of IMS Administration Foundation or Management Console.

Policy Services

Policy Services is a core IMS Tools technology that can monitor specific database state by evaluating the sensor data collected by an IMS Tools product, and by providing a response to any conditions that exceed the threshold values specified for this state. Policy Services must be configured if you plan to use the Conditional Reorganization Support Service in Smart Reorg utility jobs of IMS Database Reorganization Expert or policy evaluation.

Procedure

1. Identify which IMS Tools Base components will be used in your environment.

   The following table shows the IMS Tools Base components and whether they are supported by each tool of IMS Database Utility Solution. IMS Tools Base components that are not listed in the table are not used by the tools of IMS Database Utility Solution.

IMS Tools	Autonomics Director	Distributed Access Infrastructure	IMS Tools Common Services	IMS Tools Knowledge Base	Policy Services
IMS Database Reorganization Expert	Optional	Optional	Optional	Optional	Optional
IMS HP Unload	-	Optional	Optional	Optional	-
IMS HP Load	-	Optional	Optional	Optional	-
IMS Index Builder	-	Optional	Optional	Optional	-
IMS HP Image Copy	Optional	Optional	Optional	Optional	Optional
IMS HP Pointer Checker	Optional	Optional	Optional	Optional	Optional
IMS Library Integrity Utilities	-	-	Optional	-	-

IMS Tools	Autonomics Director	Distributed Access Infrastructure	IMS Tools Common Services	IMS Tools Knowledge Base	Policy Services
IMS Library Integrity Utilities extension (DBD/PSB Map Viewer)	-	Required	-	Required	-
IMS HALDB Toolkit	-	-	Optional	-	-
Database Sensor	Optional	Optional	Optional	Optional	Optional
Database Repair Facility	-	-	-	-	-

2. Follow the instructions in the *IMS Tools Base Configuration Guide* to configure the IMS Tools Base components that you identified in step .

   The following considerations and additional information help you configure the IMS Tools Base components for use with IMS Database Utility Solution.

   IMS Tools Knowledge Base
   Register IMS Tools products to IMS Tools Knowledge Base to enable the following functions:

   - Store product reports in the IMS Tools Knowledge Base repository
   - Collect and store sensor data for sensor data evaluation
   - IMS autonomics capabilities
   - JCL generation of IMS Administration Tool

   For more information about configuring IMS Tools Knowledge Base and registering IMS Tools products, see the *IMS Tools Base Configuration Guide*.

   Autonomics Director
   While you configure Autonomics Director, you will be guided to create skeleton JCL for Database Sensor. You can use the sample skeleton JCL, DFASJCL2, in the SHPSSAMP library to do so.

APF authorizing the program libraries

You must make certain changes to your MVS system to allow IMS Database Utility Solution to operate with the required level of authority.

About this task

Before using any of the tools in IMS Database Utility Solution, you must ensure that all the authorizations are set up correctly for your environment. The following libraries require authorized program facility (APF) authorization:

- The library that contains the IMS Database Utility Solution executable modules
- All tools that are part of IMS Database Utility Solution that are not in the SHPSLMD0 library
- The libraries of related common tools of IMS Tools Base
- The libraries that contain the user exit modules, site default table modules, and base configuration modules, if they exist

Procedure

1. Add the required libraries to your MVS APF list.

   Add the libraries in the following table to your MVS APF list and any other library that is specified in the STEPLIB of each batch utility job or IMS control region.

Table 12. Load libraries to be APF-authorized

Library	Descriptions
SAIILINK	Distributed Access Infrastructure
SFOILOAD	IMS Tools Online System Interface
SGLXLOAD	IMS Tools Generic Exits
SHKTLOAD	Autonomics Director, Distributed Access Infrastructure, IMS Tools Knowledge Base, and Policy Services
SHPSLMD0	IMS Database Utility Solution
SIIULMOD	IMS Index Builder
SIHCLOAD	IMS HALDB Toolkit

2. Complete either of the following steps to activate the updated APF list:

   - To temporarily APF-authorize the libraries, use the SETPROG APF system command. For more information, see *z/OS MVS System Commands*.
   - To permanently APF-authorize the libraries, add the libraries to the APF statement in the PROG*xx* member of SYS1.PARMLIB and re-IPL the system. For more information, see the *z/OS MVS Initialization and Tuning Reference*.

Performing security-related tasks

IMS Database Utility Solution requires specific RACF and security settings. You must ensure that all products have the necessary security authorizations in order for them to operate correctly.

Data spaces

Data spaces are used by IMS HP Load and IMS HP Image Copy under certain conditions.

Your environment might restrict the use of data spaces. If your site restricts which tools can create and use data spaces, you must authorize the following two products to allow the tool to function properly.

The following information pertains to the tools that use data spaces and the parameters that you can use to control their use.

IMS HP Load

IMS HP Load uses data spaces to reorganize the database records in the overflow part of the DBDS and to resolve long synonym chains after the database is reorganized.

The minimum amount of space that is used is 128 KB for each database. If IMS HP Load needs more data spaces, it obtains an additional 128 KB each time.

IMS HP Image Copy

IMS HP Image Copy uses data spaces only when it is called within IMS Database Reorganization Expert jobs that process secondary index databases with Type-A Image Copy processing.

IMS HP Image Copy uses data spaces to share control data among address spaces. Specifically, it uses data spaces to hold data that is being processed or to pass data between multiple address spaces. IMS HP Image Copy keywords control the use of the data spaces.

The amount of space used is 128 KB plus an additional 8 KB for each DBDS that is being processed.

If your site restricts which tools can create and use data spaces, you must authorize the load module library of IMS HP Image Copy (SHPSLMD0) to allow the tool to function properly.

Address spaces

IMS Index Builder and IMS HP Image Copy use the multiple address space architecture.

A common example is the user or job scheduler initiating the master address space (MAS) by submitting a set of saved JCL. The MAS then initiates one or more subordinate address spaces (SAS) to perform the requested processing. In some cases, the SAS also initiates other subordinate address spaces.

IMS Index Builder address space considerations

IMS Index Builder executes in a multiple address space environment. The main job can be submitted by the user or a job scheduler, or initiated as a started task.

The job name is determined by the user. The authority is obtained from the user ID that is associated with the job. IMS Index Builder can then initiate either of the following subordinate address spaces types, depending on the processing that is being performed:

IIUSORTS
> The IMS Index Builder sort subordinate address space that is initiated as a started task. The name of the started task, by default, is IIUSORTS. However, the name can be changed in the IIURPRMS customization module by using the IIURDFLT SORTP parameter.
>
> By default, IMS Index Builder uses the IIUBSRT procedure, found in the system procedure library to create the JCL for this started task. This procedure name can be overridden in the IIURPRMS customization module using the IIURSORT parameter. However, changing this procedure affects all the created address spaces.

IIUSCANS
> The IMS Index Builder scan address space which is initiated as a started task. The name of the started task, by default, is IIUSCANS. However, the name can be changed in the IIURPRMS customization module using the IIURDFLT SCAN parameter.
>
> By default, IMS Index Builder uses the IIUBSRT procedure, found in the system procedure library to create the JCL for this started task. This procedure name can be overridden in the IIURPRMS customization module using the IIURSORT parameter. However, changing this procedure affects all the created address spaces.

IIUAPIFC
> The IMS Index Builder API address space. This address space is initiated as a started task when the Parallel Reorganization Service of IMS Database Reorganization Expert initiates IMS Index Builder. The name of the started task, by default, is IIUAPIFC. However, the name can be changed in the IIURPRMS customization module by using the IURDFLT APIP parameter.
>
> IMS Index Builder uses the same procedure for creating both the IIUAPIFC and IIUSORTS address spaces.

All started tasks that are created by IMS Index Builder run with the same authority as the main job that was initiated. When IMS Index Builder initiates a started task, code that is executing in that address space creates a security control block that matches the one associated with the main job. This control block is swapped in the new address space so that it has the same level of authority as the main job. However, there is a short time in which the started task is running with default authority. For details on this situation and potential problems, see "Authorizing IMS Index Builder subordinate address space" on page 34.

The user ID that is associated with the IMS Index Builder job needs the following authority to the listed data sets:

- CONTROL access to the RECON data sets
- READ access to database data sets
- ALTER access to index data sets

IMS HP Image Copy address space considerations

IMS HP Image Copy executes in a multiple address space environment. The main job can be submitted by the user or a job scheduler, or initiated as a started task.

About this task

Advanced Image Copy Services of IMS HP Image Copy schedules the DFSMSdss program (server) and uses its application programming interface (API). The DFSMSdss server runs in a different address space from the address space of IMS HP Image Copy. When a batch job starts, it starts the DFSMSdss server address space. The server remembers which job started the server. While the server is running, it accepts other connections until the originating job ends. At the end of the originating job, the job notifies the server that it has finished and this notification alerts the server that it can stop. Until the DFSMSdss address space is stopped, the address space is reused by the DFSMSdss API.

You can also start the DFSMSdss server within IMS HP image Copy jobs. When the server is started, the server can be used by other applications that run in the same system.

Tip: The DFSMSdss server address space simplifies the running of multiple jobs. Therefore, consider starting the DFSMSdss server address space in the system whenever possible.

Procedure

To start the DFSMSdss server address space within an IMS HP Image Copy job, follow these steps:

1. Create the following started task procedure and add it to SYS1.PROCLIB.

```
//**********************************************************************
//*   THIS PROCEDURE WILL CREATE AN APPROPRIATE DFSMSDSS CROSS     *
//*   MEMORY SERVER TO BE USED WITH APPLICATIONS THAT INVOKE CROSS *
//*   MEMORY REQUESTING THE DEFAULT DFSMSDSS SERVER NAME.          *
//*                                                                *
//*   TO USE, ENTER THE FOLLOWING AT A CONSOLE:                    *
//* START DFSMSDSS,PROG=ADRXMAIB                                   *
//*                                                                *
//*   WHEN THE DFSMSDSS CROSS MEMORY SERVER IS NO LONGER REQUIRED  *
//*   ISSUE THE FOLLOWING MODIFY COMMAND:                          *
//* F DFSMSDSS,STOP                                                *
//*                                                                *
//**********************************************************************
//DFSMSDSS PROC PROG=IEFBR14
//IEFPROC  EXEC PGM=&PROG,REGION=0M,TIME=1440,DYNAMNBR=1635
```

2. Start the DFSMSdss server address space by using either of the following methods:

    • Issue the following command:

    ```
    S DFSMSDSS,PROG=ADRXMAIB
    ```

    • Add the procedure as a started task in your IPL procedure.

3. When the DFSMSdss server address space is started, run the IMS HP Image Copy jobs.

4. When your IMS HP Image Copy jobs end, and if you want to stop the server address space, issue the following command:

    ```
    F DFSMSDSS,STOP
    ```

Defining RACF (or equivalent security) authorization for IMS commands

You must define RACF (or equivalent security software) authorizations for Database Sensor, IMS Database Reorganization Expert, IMS HP Image Copy, and IMS HP Pointer Checker to operate.

About this task

The Database Sensor, IMS Database Reorganization Expert, IMS HP Image Copy, and IMS HP Pointer Checker functions allow you to execute IMS commands through a batch job. IMS commands must

be secured with RACF or other equivalent security software to control access to who can issue IMS commands. There are three different options for securing IMS commands through this feature. The information about IMS command authorization is defined in the IMS Tools Base IMS Tools Knowledge Base RECONID record.

Procedure

In the **IMSCMD Security** field, which is located in the RECONID record, specify the type of security that you want to implement by entering one of these values:

NONE
    No security authorization is performed. This value is intended for test environments only and should not be used in a secured environment. The security level of NONE allows anyone to issue any supported IMS command.

APPL
    The RACF application (APPL) resource class is used for command security checking. This method uses the application resource class (APPL) to protect the IMS commands. The **RACF Class** field in the RECONID record indicates the specific application resource name, within the RACF APPL class, that is used for securing the IMS commands.

    When you use the APPL resource class method, you must first define the application resource name (*applname* in the next example) to RACF. Define the application resource name to RACF by using the RDEFINE command, as shown in this example:

```
RDEFINE APPL applname UACC(NONE) AUDIT(NONE)
```

    After you have defined the application resource name, your RACF administrator can permit users either READ or UPDATE levels of command authorization by using the PERMIT command, as shown in this example:

```
PERMIT applname CLASS(APPL) ID(userid) ACCESS(READ|UPDATE)
```

    The userid specifies a particular RACF user or RACF group. ACCESS specifies either read-level access or update-level access. Read-level access allows a user to issue IMS commands that display IMS system information, and update-level access allows users to issue IMS commands that alter IMS system resources. The next table indicates which IMS commands are allowed and the corresponding level of access that is required to run them.

IMS
    The IMS resource class is used for command security checking. This method uses the same IMS command security class that is used by the online IMS system. The **RACF Class** field in the RECONID record indicates the RACF security class that is used for securing IMS commands. This value must match the one that you specified to IMS using the **RCLASS=** keyword on the SECURITY macro. This method allows you to use the same security class that is used by your online IMS control region.

Commands and required access levels for RACF APPL resource class authority

During the customization process, you will need to issue various commands. Make sure that your user ID has the required authority to issue these commands.

IMS commands and their corresponding required access levels for RACF APPL resource class authority are shown here.

IMS commands

The following table summarizes the IMS commands that are used in product jobs and the access levels that are required to issue the commands.

Command	Access level required	Product
/DBDUMP	Update	• IMS Database Reorganization Expert • IMS Index Builder • IMS HP Image Copy
/DBRECOVERY	Update	IMS HP Image Copy
/DISPLAY	Read	• IMS Database Reorganization Expert • IMS Index Builder • IMS HP Image Copy
/START	Update	• IMS Database Reorganization Expert • IMS Index Builder • IMS HP Image Copy
QUERY	Read	IMS HP Image Copy
UPDATE	Update	IMS HP Image Copy

DBRC API requests and DBRC commands

The following table summarizes the DBRC API requests and the DBRC commands that are used in product jobs.

DBRC API request or command	Product
DBRC API - AUTH	IMS Index Builder
DBRC API - QUERY	• IMS Database Reorganization Expert • IMS HALDB Toolkit • IMS HP Image Copy • IMS HP Load • IMS HP Pointer Checker • IMS Index Builder • IMS Library Integrity Utilities
DBRC API - RELBUF	• IMS Database Reorganization Expert • IMS HALDB Toolkit • IMS HP Image Copy • IMS HP Load • IMS HP Pointer Checker • IMS Index Builder • IMS Library Integrity Utilities

DBRC API request or command	Product
DBRC API - STARTDBRC/STOPDBRC	• IMS Database Reorganization Expert • IMS HALDB Toolkit • IMS HP Image Copy • IMS HP Load • IMS HP Pointer Checker • IMS Index Builder • IMS Library Integrity Utilities
DBRC command - CHANGE	• IMS Database Reorganization Expert • IMS HALDB Toolkit • IMS HP Image Copy • IMS Index Builder
DBRC command - INIT/DELETE	IMS HALDB Toolkit
DBRC command - LIST	• IMS Database Reorganization Expert • IMS HP Image Copy • IMS HP Pointer Checker • IMS Index Builder • IMS Library Integrity Utilities
DBRC command - NOTIFY	• IMS Database Reorganization Expert • IMS HALDB Toolkit • IMS HP Image Copy • IMS Index Builder

For the access levels that are required to issue these requests and commands, see the following topics:

• Topics about DBRC security in *IMS System Administration*
• Topics about DBRC API security features in *IMS System Programming APIs*

To set RACF control information for these IMS commands, DBRC API requests, and DBRC commands, see *IMS Commands*.

Defining RACF authority to dynamically allocate and catalog data sets

Set up the RACF authority to dynamically allocate and catalog data sets beginning with the high-level qualifier that you specified in the JCL and PROC members.

Here are some examples:

• Database data sets
• Index data sets
• Image copy data sets
• Work data sets

Setting up security for IMS Index Builder

Setting up security for IMS Index Builder includes three tasks: Authorizing IMS Index Builder subordinate address space, granting READ access to the IMS Index Builder load library for the user ID, and granting ALTER work file access to the IMS Index Builder users.

Authorizing IMS Index Builder subordinate address space

You must authorize the IMS Index Builder subordinate address space to run under a security system. Before using the subordinate address space, make sure that you have established proper security authorization and access.

About this task
The subordinate address space's user ID is obtained in two phases, as described in the following steps:

Procedure

1. Before the IMS Index Builder code is dispatched in the subordinate address space and has the opportunity to set its security identity, the operating system attempts to use the user ID from the profile definitions in the security system's STARTED class as the user ID for these started tasks.

   If you use the IMS Index Builder default job names, you can use the following RDEFINE to specify the user ID. The STARTED class is RACLIST'ed and must be refreshed after you make updates.

   ```
   RDEFINE STARTED IIU*.* STDATA(USER(userid))
   SETROPTS RACLIST(STARTED) REFRESH
   ```

   You can also use the started procedures module ICHRIN03 to associate started procedures and jobs with user IDs, instead of using the **RDEFINE STARTED** command.

   If you use the IIURPRMS module to override the default job names, you must define the STARTED class profiles that correspond to the installation names.
2. After the IMS Index Builder code is dispatched in the subordinate address space, IMS Index Builder processing determines the user ID under which the master address space runs. The subordinate address space changes its own initial user ID to match that of the master address space. Once this change occurs, the subordinate address space runs with the same security authorities as the master.

Granting READ access to the IMS Index Builder load library for the user ID

You must grant READ access to the IMS Index Builder load library SIIULMOD for the user ID.

Defining the IMS Index Builder load library to the security system with UACC(READ) is recommended so that all user IDs have access to the STEPLIB. If you do not want to define UACC(READ), you must set READ access to the load library for the user ID or for the group associated with the STARTED class profile, and also for the user ID under which the master address space runs.

Granting ALTER work file access to the IMS Index Builder users

Depending on which processing mode IMS Index Builder uses, IMS Index Builder can use stripe data sets to pass records between its address spaces. The stripe data sets exist only for the duration of that particular IMS Index Builder run.

When stripe data sets are used, the user ID associated with the master address space requires ALTER access to these data sets. You associate ALTER access to the stripe data sets by defining an IIU.STRIPE.* DATASET profile with UACC(ALTER). If your installation security policies do not allow UACC(ALTER), you must add each user that executes IMS Index Builder to this DATASET profile's access list with ALTER access.

By default, stripe data sets are allocated with the high-level qualifiers IIU.STRIPE. You can override this default by using the IIURDFLT CLASS setting in the IIURPRMS parameter module. However, if

you override the default high-level qualifiers, you must define a DATASET profile corresponding to the installation's qualifiers.

Obtaining XCF group names for IMS Tools Base servers

You must obtain the XCF group names for the IMS Tools Base server groups to set up JCL streams, cataloged procedures, and site default options for IMS Database Utility Solution tools.

XCF group name for the IMS Tools Knowledge Base servers

The XCF group name for the IMS Tools Knowledge Base servers must be specified in the JCL streams for IMS Database Utility Solution tools to use one of the following services of IMS Tools Base directly from tools' jobs:

- IMS Tools Knowledge Base report repository service
- Policy Services
- Autonomics Director

For example, specify the ITKBSRVR=*server_name* parameter in the HPSIN input of IMS Database Reorganization Expert. You can set default values for the runtime options by using the installation default option to make the specification applied to all the run of IMS Database Reorganization Expert jobs, without specifying them in each JCL stream.

XCF group name for the Autonomics Director servers

The XCF group name for the Autonomics Director servers must be specified in the JCL steams for IMS Database Utility Solution tools to use the sensor data notification service from various database sensor functions of the tools.

For example, specify the ADXCFGRP parameter in the PROC control statement of IMS HP Pointer Checker. You can set default values for the runtime options by using the installation default option to make the specification applied to all the run of IMS HP Pointer Checker jobs, without specifying them in each JCL stream.

XCF group name for the IMS Tools Online System Interface server tasks

The XCF group name for the IMS Tools Online System Interface server tasks must be specified in the JCL streams for IMS Database Utility Solution tools to issue IMS commands from the tools' jobs.

For example, specify the TOSIXCFGRP=*grpname* parameter in the HPSIN input of IMS Database Reorganization Expert, where *grpname* is the 5-character XCF group name.

Updating sample procedures and JCL

Update the required sample members in the SHPSSAMP library and sample PROC in the SIIUPROC library.

Updating the required sample members in the SHPSSAMP library

Create or modify the required sample members in the SHPSSAMP library for IMS Database Utility Solution.

Each sample member has instruction on how to customize each member.

Copy and modify the necessary PROC and JCLPDS members to your IMS PROCLIB, system PROCLIB, or JCLPDS data sets.

The following tables summarize the sample members that are used to implement each product distributed in the IMS Database Utility Solution sample library SHPSSAMP.

Subsections:

Database Sensor sample members

The following table summarizes the sample members for IMS Database Utility Solution Database Sensor.

Table 13. IMS Database Utility Solution Database Sensor sample members in the SHPSSAMP library

Sample member	Description	Type
DFASJCL1	Stand-alone Database Sensor sample JCL **Note:** You can use this JCL to verify Database Sensor installation.	JCL
DFASJCL2	Stand-alone Database Sensor sample skeleton JCL **Note:** You can use this JCL to run Database Sensor automatically by using the scheduling functions of Autonomics Director. See the topic "Customizing skeleton JCL for full-function (and HALBD) DB Sensor" in the *IMS Tools Base Configuration Guide*.	JCL
DFASDFL1	Database Sensor procedure to generate a site default table (Database Sensor Site Default Generation utility)	PROC
DFASDFL2	Database Sensor procedure to report the contents of the site default table (Database Sensor Site Default Generation utility)	PROC

IMS HP Unload sample members

The following table summarizes the sample members for IMS HP Unload.

Table 14. IMS HP Unload sample members in the SHPSSAMP library

Sample member	Description	Type
FABHDBB	IMS HP Unload DBB region procedure	PROC
FABHDB2	IMS HP Unload DLI region procedure	PROC
FABHDLI	IMS HP Unload DLI region procedure	PROC
FABHULU	IMS HP Unload ULU region procedure	PROC
FABIRGEN	IMS HP Unload Asm/Link procedure to generate Sequential Subset Randomizer	PROC

IMS HP Load sample members

The following table summarizes the sample members for IMS HP Load.

Table 15. IMS HP Load sample members in the SHPSSAMP library

Sample member	Description	Type
FABSPSSI	IMS HP Load procedure to invoke Physical Sequence Sort Reload	PROC
FABSPSSR	IMS HP Load procedure to invoke Physical Sequence Sort Reload	PROC

IMS HP Image Copy sample members

The following table summarizes the sample members for IMS HP Image Copy.

Table 16. IMS HP Image Copy sample members in the SHPSSAMP library

Sample member	Description	Type
FABJICD1	Defaults for GENJCL.IC	Input
FABJICD2	Defaults for GENJCL.IC	Input
FABJICS0	Skeleton JCL for GENJCL.IC	JCLPDS
FABJRVGR	Skeleton JCL for GENJCL.RECOV	JCLPDS

IMS HP Pointer Checker sample members

The following table summarizes the sample members for IMS HP Pointer Checker.

Table 17. IMS HP Pointer Checker sample members in the SHPSSAMP library

Sample member	Description	Type
FABKSPMP	IMS HP Pointer Checker procedure to invoke Space Monitor.	PROC
FABPP	IMS HP Pointer Checker procedure to invoke HD Pointer Checker to check pointers. This procedure uses a PSB. The work data sets are defined in the procedure.	PROC
FABPPA	IMS HP Pointer Checker procedure to invoke HD Pointer Checker to check pointers. This procedure uses a PSB. The job runs in the DBB region by using ACBLIB. HD Pointer Checker dynamically allocates the work data sets.	PROC
FABPPD	IMS HP Pointer Checker procedure to invoke HD Pointer Checker to check pointers. This procedure uses a DBD. The work data sets are not defined in the procedure; HD Pointer Checker dynamically allocates them.	PROC
FABPPTA	IMS HP Pointer Checker procedure to invoke HD Pointer Checker and HD Tuning Aid to check pointers and root-segment locations. This procedure uses a PSB.	PROC

Table 17. IMS HP Pointer Checker sample members in the SHPSSAMP library (continued)

Sample member	Description	Type
FABPC	IMS HP Pointer Checker procedure to invoke HD Pointer Checker to check pointers. This procedure uses a PSB. The work data sets are defined in the procedure. Use this procedure if you want Database Sensor to store sensor data or Space Monitor to monitor IMS online full-function databases while checking pointers.	PROC
FABPCD	IMS HP Pointer Checker procedure to invoke HD Pointer Checker to check pointers. This procedure uses a DBD. The work data sets are not defined in the procedure; HD Pointer Checker dynamically allocates them. Use this procedure if you want Database Sensor to store sensor data or Space Monitor to monitor IMS online full-function databases while checking pointers.	PROC
FABPCTA	IMS HP Pointer Checker procedure to invoke HD Pointer Checker and HD Tuning Aid to check pointers and root-segment locations. This procedure uses a PSB. Use this procedure if you want Database Sensor to store sensor data or Space Monitor to monitor IMS online full-function databases while checking pointers.	PROC
FABTIMS	IMS HP Pointer Checker procedure to invoke HD Tuning Aid under IMS.	PROC
FABTMVS	IMS HP Pointer Checker procedure to invoke HD Tuning Aid for MVS batch.	PROC

IMS Library Integrity Utilities sample members

The following table summarizes the sample member for IMS Library Integrity Utilities.

Table 18. IMS Library Integrity Utilities sample members in the SHPSSAMP library

Sample member	Description	Type
FABLPGEN	IMS Library Integrity Utilities procedure to generate LIU global option module	PROC

Updating the required sample PROC in the SIIUPROC library

Copy the required sample PROC in the SIIUPROC library to your IMS PROCLIB or system PROCLIB data sets and modify the member.

Table 19. Sample PROC in the SIIUPROC library

Sample member	Product	Description	Type
IIUBSRT	IMS Index Builder	Sort address space procedure	PROC

Setting up Management Console

Configure IBM Management Console for IMS and Db2 for z/OS (Management Console) if you want to use Management Console to view sensor data charts, database exceptions, reports, graphical view of database and program structures, and DBD/PSB source through a web browser.

About this task

You can install the web server for Management Console on Windows or on z/OS. The Management Console web server is a central server that consolidates information from various z/OS sources, so only a single server must be installed.

Procedure

For instructions for configuring the Management Console server, see the topic "Configuring Management Console" in the *Management Console User's Guide*.

What to do next

To display graphical view of database structures, program specifications, and DBD and PSB source, you must also install the DBD/PSB Map Viewer, the IMS Library Integrity Utilities extension for Management Console.

Install the DBD/PSB Map Viewer on a Windows system or a z/OS system where Management Console is installed:

Installing the DBD/PSB Map Viewer on z/OS

Use Installation Manager to install the IMS Library Integrity Utilities extension to the Management Console server that is running on z/OS.

Before you begin

- Ensure that Installation Manager for z/OS 1.8 or later is installed. Installation Manager is delivered with IBM Management Console for IMS and Db2 for z/OS. For more information, see the *Program Directory for IBM Management Console for IMS and Db2 for z/OS*, GI10-9007.
- Ensure that the following IMS Tools Base components are installed and configured:
  - IMS Tools Knowledge Base
  - Distributed Access Infrastructure
- Ensure that the following requirements are satisfied:

– Management Console is installed with the feature that includes capabilities for managing IMS environments. This feature is installed by the `imsbase.feature` parameter on the install command during installation.

– Management Console is at the latest maintenance level.

– Management Console server is configured on z/OS.

• Ensure that IMS Library Integrity Utilities for z/OS is installed and that APAR PI50732 is applied. The DBD/PSB Map Viewer is provided by IMS Library Integrity Utilities as an extension to Management Console.

About this task

During the SMP/E installation of IMS Library Integrity Utilities for z/OS, the IMS Library Integrity Utilities extension installation package is delivered in the file liu220mc.zip. This file is placed in the Installation Manager package directory.

You must use Installation Manager to install the IMS Library Integrity Utilities extension with this package.

Procedure

To install the DBD/PSB Map Viewer on z/OS:

1. If the Management Console server is started, stop the started task DYWZSRV.
2. Mount the zFS file systems that contain the following resources.

   • Installation Manager package

      Default mount point: `/usr/lpp/InstallationManagerRepository/H27P220`

   • Installation directory of Management Console

      Default mount point: `/usr/lpp/IBM/mgmtconsole/v110`

   Tip: You can make sure that the zFS file systems are mounted by issuing the **DISPLAY OMVS,FILE** command from the MVS console.

3. Customize and run the Installation Manager job, HPSAIINS, which is located in the SHPSBASE data set. The HPSAIINS job starts Installation Manager to install the IMS Library Integrity Utilities extension.

   The following conditions must be satisfied to run the HPSAIINS job:

   • The required privileges of user ID depend on the Installation Manager mode that is being used:

     – If you are using administrator mode, the user ID must be a superuser.

     – If you are using nonadministrator mode, the user ID must be the one that owns the instance of Installation Manager that you are using. The user ID must also have read permission to the Installation Manager package and write permission to the installation directory of Management Console.

     – If you are using group mode, you must run the HPSAIINS job under a user ID that belongs to the UNIX System Services group that is associated with Installation Manager. The user ID must also have read permission to the Installation Manager package and write permission to the installation directory of Management Console.

   • The user ID must have access to the SAF profiles that are specified in the Installation Manager sample job GIN2ADMN.

   • The user ID must be capable of running a job that requires more than 600 MB region size.

4. Verify that the job ends with return code 0.
5. Start the Management Console server by running the started task DYWZSRV.

What to do next

After the installation is complete, configure the DBD/PSB Map Viewer environment. See .

Tips:

- You can also use the HPSAIINS job to update the IMS Library Integrity Utilities extension by using a new Installation Manager package provided by a PTF.
- To uninstall the IMS Library Integrity Utilities extension, use the HPSAIUNI member in the SHPSBASE data set. Before you run the job, you must change the installation directory in the JCL to `/usr/lpp/IBM/mgmtconsole/v110` or the directory where Management Console is installed. When you uninstall Management Console, you must delete the IMS Library Integrity Utilities extension before you uninstall Management Console.

Installing the DBD/PSB Map Viewer on Windows

Use Installation Manager to install the IMS Library Integrity Utilities extension to the Management Console server that is running on Windows.

Before you begin

- Ensure that Installation Manager for Windows 1.8 or later is installed. If it is not installed, you can download it from the IBM Support Portal at http://www.ibm.com/support.

  Download either of the following files:

  - agent.installer.win32.win32.x86
  - agent.installer.win32.win32.x86_64

- Ensure that the following IMS Tools Base components are installed and configured:

  - IMS Tools Knowledge Base
  - Distributed Access Infrastructure

- Ensure that the following requirements are satisfied:

  - Management Console is installed with the feature that includes capabilities for managing IMS environments. This feature is installed by the **Management Console for IMS Management Base Feature** option during installation.
  - Management Console is at the latest maintenance level.
  - Management Console server is configured on Windows.

- Ensure that IMS Library Integrity Utilities for z/OS is installed and that APAR PI50732 is applied. The DBD/PSB Map Viewer is provided by IMS Library Integrity Utilities as an extension to Management Console.

About this task

During the SMP/E installation of IMS Library Integrity Utilities for z/OS, the IMS Library Integrity Utilities extension installation package is delivered in the file liu220mc.zip. This file is placed in the Installation Manager package directory.

You must use Installation Manager to install the IMS Library Integrity Utilities extension with this package.

Procedure

To install the DBD/PSB Map Viewer on Windows:

1. Stop the Management Console server by clicking **Start** > **IBM Management Console for IMS and Db2 for z/OS** > **Stop Management Console Web Server**.
2. Mount the zFS file system that contains the Installation Manager package on z/OS.

The default mount point is `/usr/lpp/InstallationManagerRepository/H27P220`.

Tip: You can make sure that the zFS file system is mounted by issuing the **`DISPLAY OMVS,FILE`** command from the MVS console.

3. Use FTP to transfer the IMS Library Integrity Utilities extension installation package (liu220mc.zip) from z/OS to your preferred location on Windows. This package is placed in the Installation Manager package directory during the SMP/E installation of IMS Library Integrity Utilities for z/OS.

   Important: Transfer the file in binary mode. Otherwise, the package cannot be used by Installation Manager.

4. Start Installation Manager from the Windows Start menu by clicking **Start** > **IBM Installation Manager** > **IBM Installation Manager**.

5. Add the file liu220mc.zip as a repository in Installation Manager by clicking **File** > **Preferences** and then clicking **Add Repository**. Specify the .zip file and click **OK**. Close the Preferences dialog by clicking **OK**.

   If you receive a `failed to connect repository` error, ensure that you transferred the file in binary mode during FTP transfer.

6. Click **Install**.

7. Select IMS Library Integrity Utilities for z/OS and click **Next**.

8. Complete the installation by following the directions in Installation Manager.

9. Start the Management Console server by clicking **Start** > **IBM Management Console for IMS and Db2 for z/OS** > **Start Management Console Web Server**.

What to do next

After the installation is complete, configure the DBD/PSB Map Viewer environment. See .

Tip: Later, you might use the panel in step to update or uninstall the extension.

- To update the IMS Library Integrity Utilities extension by using a new Installation Manager package provided by a PTF, click **Update**.
- To uninstall the IMS Library Integrity Utilities extension, click **Delete**. When you uninstall Management Console, you must delete the IMS Library Integrity Utilities extension before you uninstall Management Console.

Configuring the DBD/PSB Map Viewer environment

To set up the DBD/PSB Map Viewer, you must first work on the z/OS system and then through the Management Console web interface.

About this task

The following figure shows the required configuration setup for the DBD/PSB Map Viewer.
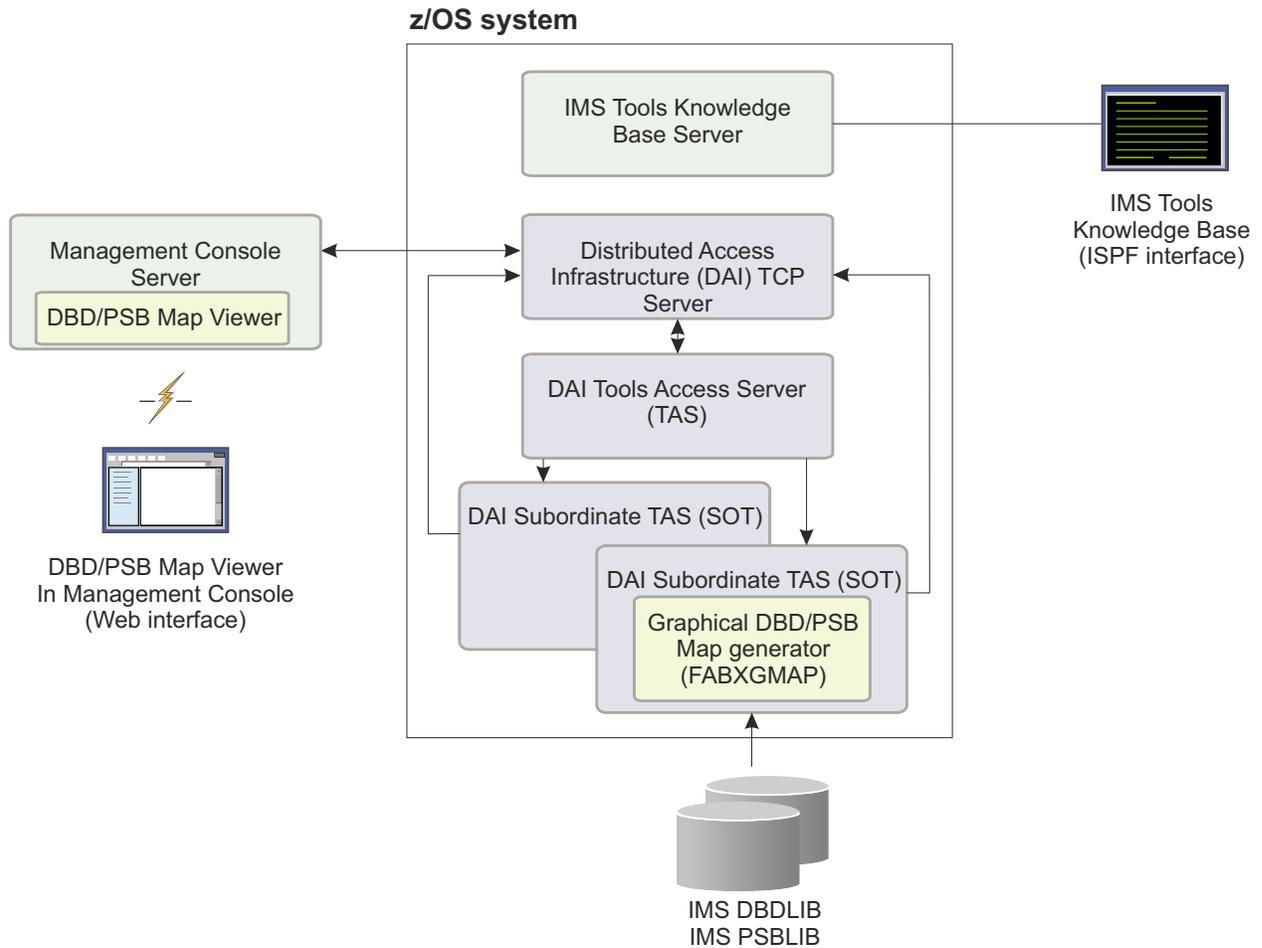
Figure 1. DBD/PSB Map Viewer configuration

The DBD/PSB Map Viewer uses Distributed Access Infrastructure to communicate with the graphical DBD/PSB Map generator (FABXGMAP), which is provided by IMS Library Integrity Utilities. When the generator receives a request message, it decodes the binary data in the load modules in the IMS.DBDLIB and IMS.PSBLIB libraries and sends the result back to Management Console. All the control statements that were used for defining the database, data sets, and segments can be viewed with the DBD/PSB Map Viewer.

Complete the following procedure to set up the DBD/PSB Map Viewer configuration.

Procedure

1. Start the IMS Tools Knowledge Base server, the Distributed Access Interface TCP server and TAS, and, if not started yet, the Management Console server.
2. Add your RECON environment to IMS Tools Knowledge Base.

   For instructions, see the topic "RECON ID (locale) administration" in the *IMS Tools Base Configuration Guide*.

   For the **Add IMS Datasets** field in the Add Recon Information panel, specify the IMS DBDLIB and PSBLIB data sets.
3. Submit the IMS Tools Discovery Utility batch job.

   For instructions, see the topic "IMS Tools Discovery Utility" in the *IMS Tools Base IMS Tools Knowledge Base User's Guide and Reference*.

What to do next

When you finish the configuration steps on the z/OS system, complete the configuration steps through the Management Console web interface.

Log in to the web interface and click **Help**. Locate the topic "Getting started with DBD/PSB Map Viewer" and complete the configuration steps.

Setting up Unified Management Server for IMS Administration Foundation

If you have installed IBM Unified Management Server for z/OS, you can configure and activate IMS Administration Foundation features, which provide a sub set of the features that have been provided by IBM Management Console for IMS and Db2 for z/OS.

About this task

The IMS Administration Foundation features that are activated by IMS Tools Base and IMS Database Utility Solution include viewing charts based on sensor data, database exceptions, reports, graphical view of database or program structures, and DBD/PSB source. The graphical user interface is provided by IBM Unified Experience for z/OS that runs as an application for Zowe virtual desktop.

The IBM Unified Management Server for z/OS provides a web service infrastructure and a unified user interface that consolidates information from IMS subsystems and Db2 subsystems on a sysplex.

At least one Unified Management Server server instance must be set up on each sysplex and on top of a Zowe infrastructure running on the sysplex.

Procedure

1. For instructions for activating and configuring IMS Administration Foundation features for IBM Unified Management Server for z/OS, see the topic "Post-SMP/E installation for IMS Administration Foundation" in the *Unified Management Server User Guide*.
2. To use the IMS Administration Foundation features on the graphical user interface provided by IBM Unified Experience for z/OS, see the topic "Using IMS Administration Foundation" in the *Unified Management Server User Guide*.

Setting up IMS HALDB Toolkit

IMS HALDB Toolkit provides functions to maintain HALDBs and to convert IMS full-function databases to HALDBs. If you want to use IMS HALDB Toolkit, you must create the base configuration module for IMS HALDB Toolkit.

About this task

IMS HALDB Toolkit uses temporary RECON data sets and a temporary DBD library for some functions. The base configuration module defines the allocation parameters for creating these resources when IMS HALDB Toolkit is started.

Procedure

1. Edit the sample member, IHCCSET, which is provided in the SIHCSAMP library.

   The following figure shows the IHCCSET job. This job creates the base configuration module for IMS HALDB Toolkit.

```
//IHCCSET    JOB
//C       EXEC  PGM=ASMA90,
//              PARM='LIST,RENT,DECK,NOOBJ'
//SYSPRINT DD   SYSOUT=*
//SYSLIB   DD   DSN=SYS1.MACLIB,DISP=SHR
//         DD   DSN=DBSP.SIHCSAMP,DISP=SHR
//SYSUT1   DD   SPACE=(CYL,(2,2)),UNIT=SYSDA
//SYSPUNCH DD   DSN=&&OBJ,SPACE=(CYL,(1,1)),
//              UNIT=SYSDA,DISP=(MOD,PASS),
//              DCB=(BLKSIZE=2400,LRECL=80,RECFM=FB)
//SYSIN    DD   *
WSET0    IHCXSET0  RECON=your.TEMP.RECON,
               RCVOL=rcvol,
               RCDATAC=*,
               RCSTORC=*,
               DBDDSN=your.TEMP.DBDLIB,
               DBDVOL=dbdvol,
               DBDCLAS=*,
               DBSCLAS=*
         END
//L       EXEC  PGM=HEWL,PARM='XREF,LIST',COND=(4,LE,C)
//SYSPRINT DD   SYSOUT=*
//SYSLIB   DD   DISP=SHR,DSN=DBSP.SIHCLOAD
//SYSLMOD  DD   DISP=SHR,DSN=DBSP.SIHCLOAD
//SYSUT1   DD   SPACE=(CYL,(5,2)),
//              UNIT=SYSDA
//SYSLIN   DD   DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD   *
  NAME  IHCWSET0(R)
//
```

a) Define the allocation information for the temporary RECON data set.

The following allocation parameters are used to create the temporary RECON data set. Depending on your SMS definition, you might use only a subset of these parameters. However, if you are not using SMS, you must specify a volume serial number.

Parameter	Description
RECON	Required. Specifies the temporary RECON data set name high-level qualifier.
	You can specify up to 26 characters of the data set name. This parameter will be appended internally with the DBD name plus a time stamp.
RCVOL	Required for non-SMS environments. Specifies the volume serial number for the temporary RECON data set.
RCDATAC	Optional. Specifies the DATACLAS for the temporary RECON data set.
RCSTORC	Optional. Specifies the STORCLAS for the temporary RECON data set.

b) Define the allocation information for the temporary DBD library.

The following allocation parameters are used to create the DBD library:

Parameter	Description
DBDDSN	Required. Specifies the temporary DBDLIB data set name high-level qualifier.
	You can specify up to 26 characters of the data set name.
DBDVOL	Required for non-SMS environments. Specifies the volume serial number for the temporary DBDLIB.
DBDCLAS	Optional. Specifies the DATACLAS for the temporary DBDLIB data set.
DBSCLAS	Optional. Specifies the STORCLAS for the temporary DBDLIB data set.

2. Run the IHCCSET job.

   HLASM and the LINKAGE editor create the base configuration module.
3. Add the base configuration module to the STEPLIB DD concatenation of IMS HALDB Toolkit runtime JCL. Ensure that the executable load libraries of IMS HALDB Toolkit are APF-authorized.

Setting up the Integrity Checker utility of IMS Library Integrity Utilities

The Integrity Checker utility of IMS Library Integrity Utilities prevents accidental misuse of ACBs or DBDs in your production environment. If you want to use the Integrity Checker utility, you must configure the utility.

Before you begin

Before you set up the Integrity Checker utility, see the topic "Integrity Checker overview" in the *IMS Library Integrity Utilities User's Guide* and learn how the utility prevents database corruptions and how it works.

Planning for Integrity Checker configuration

Before you activate the DMB verification function of Integrity Checker, you must understand the resources used by Integrity Checker and design the most suitable Integrity Checker configuration for your IMS environment.

After you design your Integrity Checker configuration, activate Integrity Checker by completing the instructions in "Activating Integrity Checker" on page 58.

Use the following topics to design the Integrity Checker configuration:

- "LIU load module library customization" on page 46
- "LICON data sets and global option modules" on page 47
- "Integrity Checker configuration requirements" on page 48
- "Runtime options and environments" on page 53
- "Historical data maintained in LICON data sets" on page 54
- "Considerations for activating Integrity Checker" on page 56

LIU load module library customization

The IMS Library Integrity Utilities load module library (also referred to as LIU load module library) must be customized to use Integrity Checker.

The DMB verification function of Integrity Checker is triggered by IMS jobs (IMS online, application, IMS standard utility, and IMS Tools utility jobs) that access databases. Therefore, unlike other IMS Tools programs that run with JCL in which the EXEC PGM= parameter specifies the utility program name to execute, Integrity Checker must be customized to automatically start DMB verification when a database is accessed.

The DMB verification process runs as part of the DBRC authorization process. Therefore, instead of coding JCL statements, you must customize the LIU load module data set (SHPSLMD0 data set) to work with DBRC module DSPCRTR0.

Important: In IMS Library Integrity Utilities 2.1 and earlier, alias name DSPCRTR0 was assigned for the FABLRTR0 module by default, but in IMS Library Integrity Utilities 2.2 and later, the alias name is not assigned. You must customize the LIU load module library.

Methods for customizing the LIU load module library

Integrity Checker supports two methods for customizing the LIU load module data set. Compare the two methods and determine the best method for your environment.

Method 1: Defining alias name DSPCRTR0 for FABLRTR0 (recommended method)
Module FABLRTR0 is the program load module that is stored in the LIU load module library. Module DSPCRTR0 is the DBRC router module, which is one of the DBRC authorization modules.

If alias name DSPCRTR0 is defined for the FABLRTR0 module, and the LIU load module library precedes the IMS resident library (SDFSRESL data set) in the STEPLIB DD in JCL and procedures of IMS jobs, when a database is accessed by such jobs, instead of the DBRC module DSPCRTR0, the FABLRTR0 module starts and activates DMB verification. To apply this method, the LIU load module library must be APF-authorized.

- Advantage: You can use the standard SMP/E methods to apply program temporary fixes (PTFs) for both IMS Library Integrity Utilities and IMS.
- Disadvantage: You must modify the STEPLIB DD statement in JCL and procedures of all IMS jobs that access databases.

Method 2: Merging Integrity Checker load modules into the IMS resident library (SDFSRESL)
If all the Integrity Checker modules (FABL* members) in the LIU load module library are merged into the IMS SDFSRESL data set, when a database is accessed, the FABLRTR0 module, which is link-edited to the DBRC module DSPCRTR0 module, starts and activates DMB verification. Before the modules are merged, the FABLRTR0 module and DBRC module DSPCRTR0 must be link-edited to create one DSPCRTR0 module, and the module entry point must be set to FABLRTR0. Before the modules are merged, a backup of the IMS SDFSRESL data set must be created.

- Advantage: You do not need to modify the STEPLIB DD statement in each JCL and procedure of IMS jobs that access databases.
- Disadvantage: When you apply PTFs for IMS Library Integrity Utilities and IMS, the following additional steps are required:
  - For IMS Library Integrity Utilities, after you apply a PTF, you must merge the members again.
  - For IMS, if the PTF updates the DSPCRTR0 module, you must restore the DSPCRTR0 module before you apply the PTF. After you apply the PTF, create a backup of the SDFSRESL data set, and then link-edit FABLRTR0 and DSPCRTR0 again.

Integrity Checker supports multiple versions of IMS with one module. Therefore, with either method, if the version of IMS is supported by IMS Library Integrity Utilities, the same LIU load module data set can be used.

LICON data sets and global option modules

LICON data sets and global option modules are unique resources that are required by Integrity Checker.

LICON data sets

LIU Integrity control data sets (LICON data sets) are KSDS data sets that are the repositories for *registered DMB entries* (RDEs). RDEs contain database management block (DMB) information that is used in DMB verification. In the LICON data sets, RDEs are stored as KSDS records. At least one LICON data set must be defined and initialized.

Integrity Checker obtains the DMB information that IMS used to load the database, stores the information in the RDE as the correct DMB information, and refers to that DMB information to verify the DMB information obtained from the DBDLIB, ACBLIB, or IMS directory that is referenced by IMS jobs.

In addition to the DMB information, RDEs contain the following information:

- Runtime options that are applied to DMB verification.
- Time stamp of the last database access made by load utility programs, unload utility programs, and utilities or application programs with an update intent in the PSB.

When a DBD is updated, Integrity Checker updates the corresponding RDE accordingly.

In addition to the RDE that contains the latest DMB information, Integrity Checker also maintains historical copies of RDEs. You can use those copies to review the DBD information before a DBD change, or when you restore DBDs, you can use the copies to restore RDEs.

Global option modules

Global option modules are the configuration definition modules of Integrity Checker. Each global option module defines the LICON data set and the runtime options that are applied to DMB verification. IMS Library Integrity Checker does not provide global option modules, so you must create at least one global option module.

IMS Library Integrity Utilities provides the FABLPGEN program for creating global option modules. The LICON data set name and runtime options are defined through FABLPGIN macro control statements of the FABLPGEN program.

After running the FABLPGEN program, the source code must be assembled and link-edited to a load module. The created global option module must be stored in the LIU load module library.

Integrity Checker configuration requirements

Because DMB verification works within DBRC authorization processing, DBRC must be active in your IMS environment. The number of LICON data sets and global option modules in an Integrity Checker configuration depends on how the DBRC environment is configured.

The basic rules for designing an Integrity Checker configuration are as follows:

- One LICON data set must be created for each set of RECON data sets. (A set consists of RECON1, RECON2, and RECON3.)
- Global option modules contain the names of the LICON data sets. Therefore, if your environment requires multiple LICON data sets, you must prepare the same number of global option modules.
- Global option modules must be named using the following naming convention:

    LIU@INST
    The default name for the global option module. If your environment requires only one LICON data set, use this name.

    LIU@*imsid*
    If your environment requires multiple LICON data sets, use this format. For *imsid*, use the IMS ID of each IMS subsystem.

    When multiple global option modules exist, Integrity Checker determines the global option module to use from the IMS ID. If no corresponding global option module is found for an IMS ID, Integrity Checker uses the default module LIU@INST. Even if the corresponding global option module is found, if some options are not defined in that module, the options that are defined in LIU@INST are applied. If LIU@INST is not found or some options are not defined in the LIU@INST, the system default values are used.

    Tip: At sign (@) is a code-page-dependent character. If you are working in an environment where you cannot use the at sign (@), name the modules LIUGINST or LIUG*imsid*.

Use the following examples to design an Integrity Checker configuration for your environment.

As a rule, the same runtime options must be defined for each LICON data set. To simplify the explanations, these examples are cases where one global option module refers to one LICON data set.

Tip: Certain runtime options can have different values within one LICON data set. To define different runtime option values, instead of assigning alias names for global option modules as shown in the examples, create one global option module for each IMS ID. For more information about the runtime options, see "Global option module generation macro" on page 68.

For IMS environments that are not explained in these examples, such as IMSplex, the basic rules are the same. For any IMS environment, you can follow these basic rules to design Integrity Checker configuration:

- One LICON data set for one set of RECON data sets
- One global option module for one LICON data set

Single IMS subsystem configuration example

The following figure shows the simplest configuration.

In this environment, one IMS subsystem is used and one set of RECON data sets is used, so the Integrity Checker resources that are required in this environment are as follows:

- Number of LICON data sets: 1
- Number of global option modules: 1

Because only one global option module is required, the name of the global option module is LIU@INST.

IMSA
DBRC=Y

IMS SDFSRESL

LIU load
module library

RECON
data sets

IMS databases

LICON data set

Global option module

LIU@INST

1. Create a LICON data set.

2. Define the LICON data set
   name in LIU@INST.

Figure 2. Integrity Checker configuration for a single IMS subsystem

Multiple IMS subsystems configuration example: Data-sharing environment

The following figure illustrates a configuration where two IMS subsystems share databases.

In this environment, two IMS subsystems are used and one set of RECON data sets is used, so the Integrity Checker resources that are required in this environment are as follows:

- Number of LICON data sets: 1
- Number of global option modules: 1

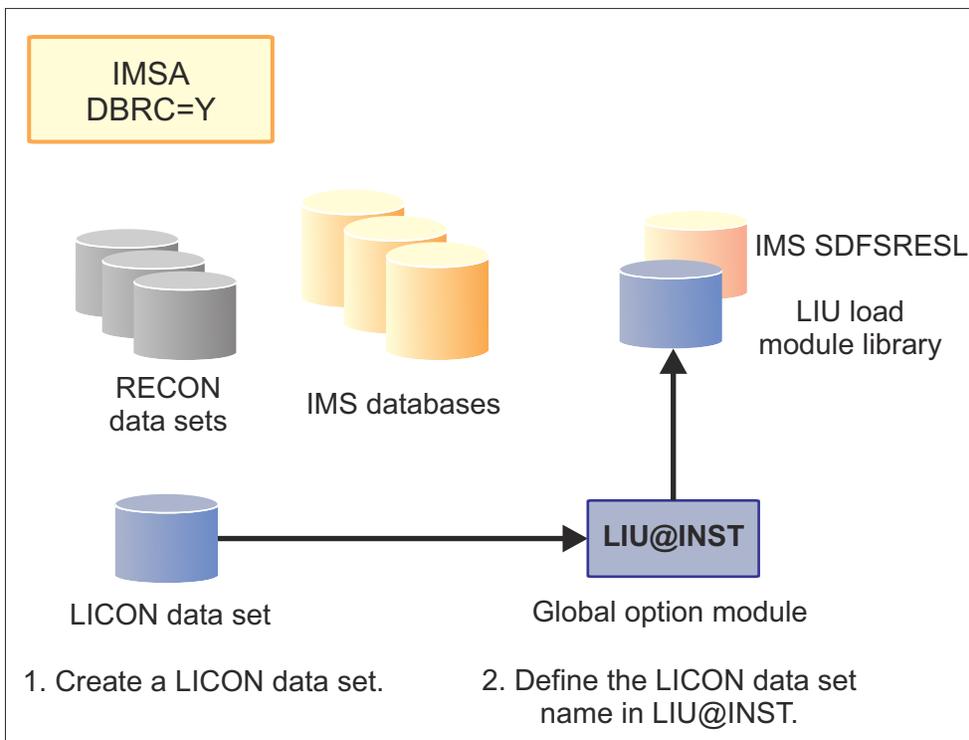Because only one global option module is required, the name of the global option module is LIU@INST.
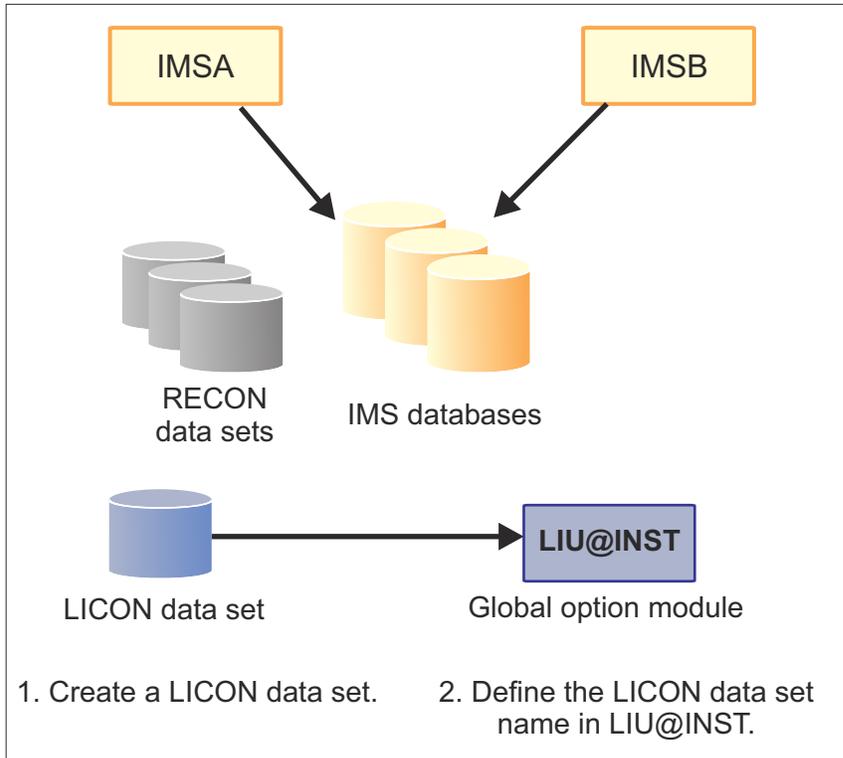

Figure 3. Integrity Checker configuration in a data-sharing environment

Multiple IMS subsystems configuration example: Non-data-sharing environment

The following figure illustrates a configuration where two IMS subsystems use a different set of databases. In such an environment, a LICON data set must be created for each IMS subsystem. Each LICON data set requires one global option module.

In this environment, two IMS subsystems are used and two sets of RECON data sets are used, so the Integrity Checker resources that are required in this environment are as follows:

- Number of LICON data sets: 2
- Number of global option modules: 2

Because two global option modules are required, the names of the global option modules are LIU@IMSA for IMS subsystem IMSA, and LIU@IMST for IMS subsystem IMST.
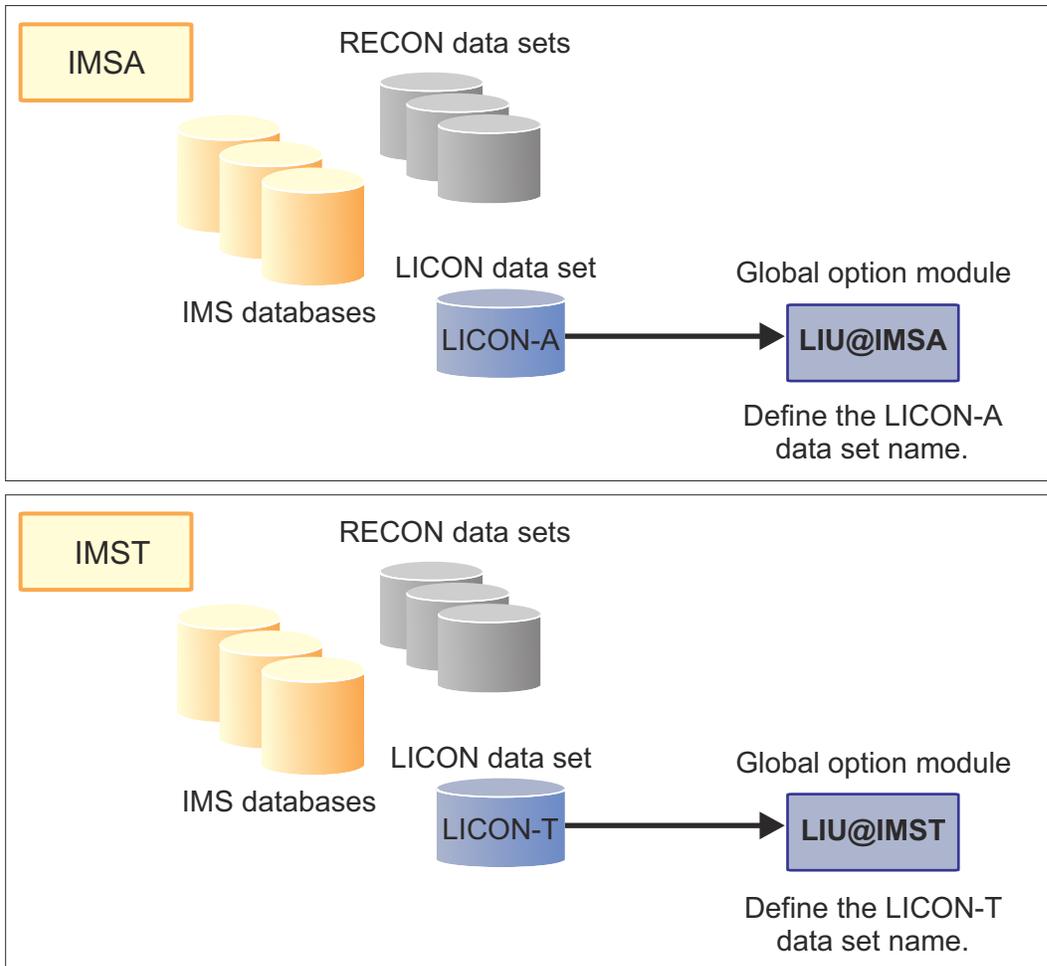
Figure 4. Integrity Checker configuration in a non-data-sharing environment

Multiple IMS subsystems configuration example: Multiple data-sharing environments

The following figure illustrates a configuration for multiple data-sharing environments. In each data-sharing environment, two IMS subsystems use the same set of databases. In such environments, a LICON data set must be created for each data-sharing environment.

Generally, when four IMS subsystems exist, four global option modules are required. However, as shown in the figure, if you create a global option module for each LICON data set and define an alias name for each global option module, you can design the Integrity checker configuration with two global option modules.

Create global option module LIU@IMSA for IMS subsystem IMSA, and then define alias name LIU@IMSB for LIU@IMSA so that IMS subsystem IMSB can also use LIU@IMSA. Do the same for IMS subsystems IMSC and IMSD.

Tip: If you create two global option modules (LIU@IMSA and LIU@IMSB), the LICON data set names and the runtime options must be the same between the two global option modules. By assigning an alias name and sharing a global option module, the same values are automatically applied; you can prevent errors that might be caused by inconsistency.
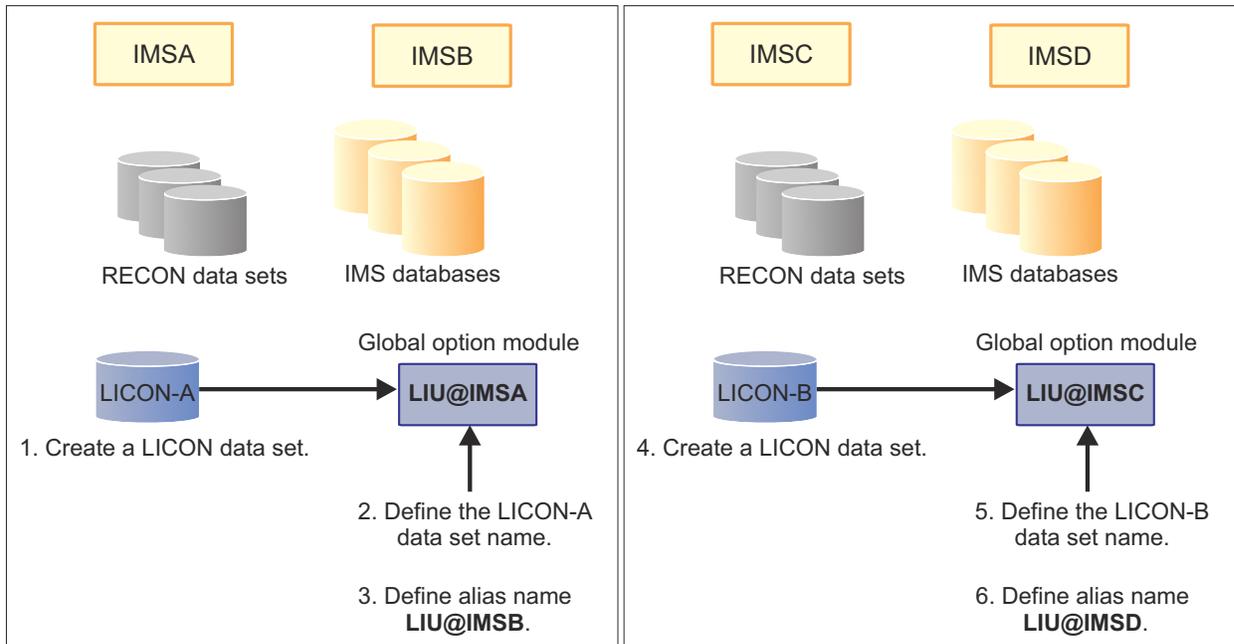
Figure 5. Integrity Checker configuration in multiple data-sharing environments

Multiple IMS subsystems configuration example: XRF complex

To use Integrity Checker in an XRF complex, the active IMS subsystems and the alternate IMS subsystems must use the same LICON data set. If they use different LICON data sets, the change log of the DMB is not inherited during takeover, and Integrity Checker might not detect the DMB inconsistency or might deny DBRC authorization even though the correct DBDLIB or ACBLIB is used.

If the number of active IMS subsystems is one, the number of required LICON data set is also one. The name of the global option module is LIU@INST.

If multiple active IMS subsystems exist, define a LICON data set for each set of RECON data sets, and create one global option module for each LICON data set.

For example, in an XRF complex shown in the following figure, create a global option module for each active IMS subsystem. Create LIU@IMS1 and LIU@IMS3, and then assign alias name LIU@IMS4 for LIU@IMS3.

Then, to apply the same runtime options to the active IMS subsystems and the alternate IMS subsystems, assign alias to the global option modules so that the alternate IMS subsystems can also use the same global option modules.

Specifically, IMS subsystems IMS1 and IMS2 are in a same XRF complex, so the two subsystems must use the same LICON data set. Therefore, assign alias name LIU@IMS2 for LIU@IMS1. IMS subsystems IMS5 and IMS6 are alternate IMS subsystems for IMS3 and IMS4, so all these subsystems must use the same LICON data set. Therefore, assign alias names LIU@IMS5 and LIU@IMS6 for LIU@IMS3.
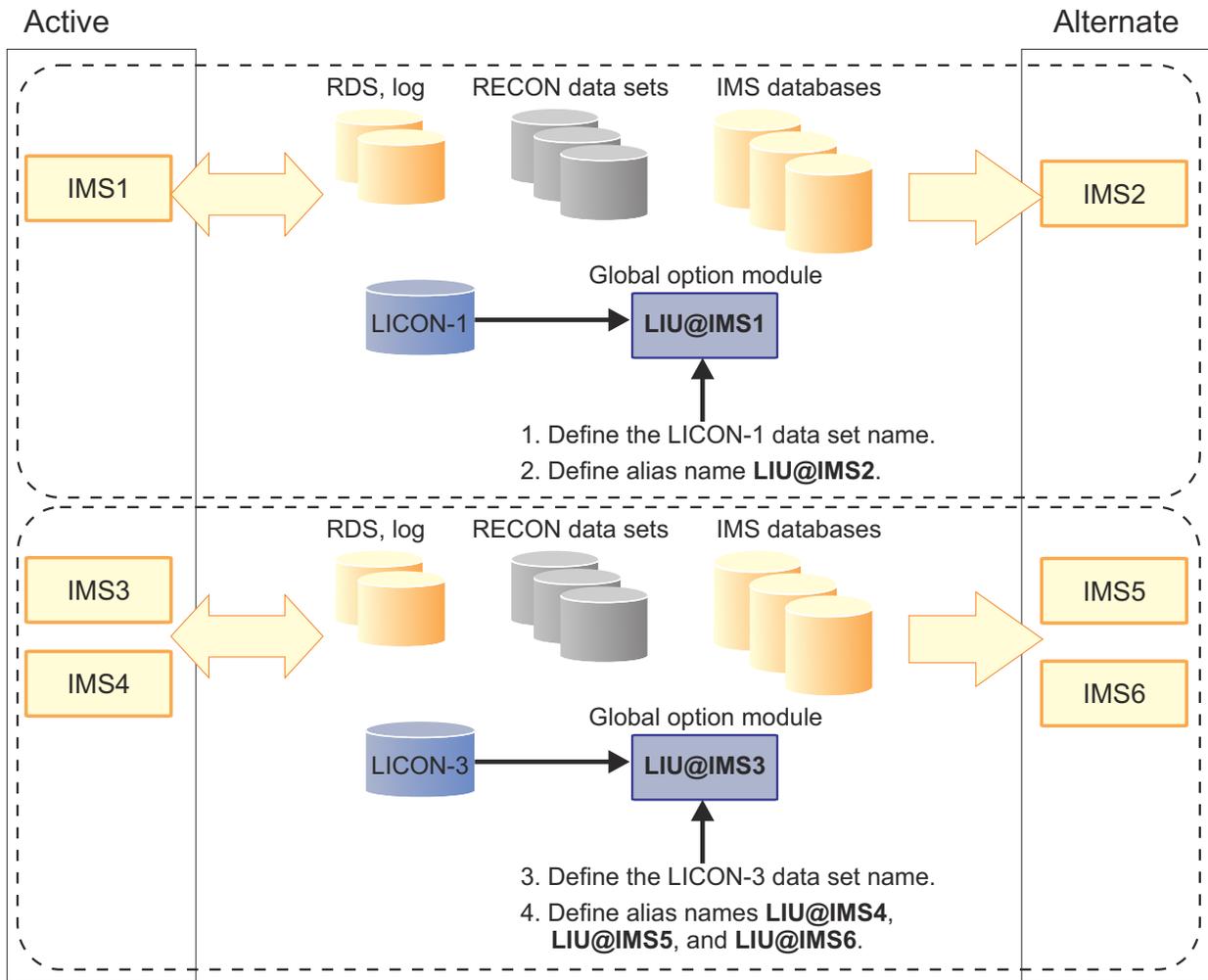
Figure 6. Integrity Checker configuration in an XRF complex

Runtime options and environments

RDEs contain runtime options for DMB verification. Some runtime options can be defined differently for each IMS processing environment.

How the runtime options are specified

The runtime options are commonly used across the environment that uses the global option module. The global default values that are applied to the options when creating new RDEs can be specified for each database.

The runtime options are defined in the global option module. You can specify the following runtime options:

- The data set name of the LICON data set
- DMB verification method (single-step or double-step)
- Routing codes and descriptor codes of Integrity Checker messages
- Option values to control the processing of Integrity Checker
- Global default values for the options used in creating new RDEs:
  - DMB verification option
  - Database access record option

– Number of expired RDEs to maintain

You specify the option values in the global option module by using the global option module generation macro. The option values that you specify are copied to the RDE when an RDE is created.

Generally, you can use the option values without changing them, but by using the LICON utility, you can change the option values in RDEs after they are created. By changing the values, you can apply different options for each database. For more information about these options, see .

How the runtime options are applied

Some runtime options can be set differently for each environment that Integrity Checker operates. Such options are referred to as *DMB verification options*.

DMB verification options control the processing and the action taken when a DMB mismatch is found. You can specify the options for each of the following IMS processing environments:

- Online IMS subsystem
- Batch program
- User load program
- Batch image copy utility
- Database recovery utility

For each environment, you can specify:

- Whether to activate DMB verification.
- Whether DMB verification sets a nonzero return code for the database authorization request to deny it, or only issues a warning message and registers the DMB to the LICON data set.

Using these options, you can control the behavior of DMB verification differently in each environment.

Recommendation: In all environments, set the option to deny the database authorization request when a DMB mismatch is found. By setting this option, you can prevent database corruption when an incorrect DBDLIB or ACBLIB is used.

Attention: Use the option *Only issue a warning message and register the DMB to the LICON data set* with caution. Use this option only when you intend to change the DBD, and you want the updated DBD reflected in the RDE so that the new DMB information is used in future DMB verifications. When this option is used, database corruption caused by use of an incorrect DBDLIB or ACBLIB cannot be prevented.

For example, if you want to use the DBD that is specified by a user load program for future DMB verifications, but you want to restrict DBD changes in other environments, you can have Integrity Checker issue a warning message and update the RDE in the user load program jobs, but deny database authorization when a DMB mismatch is found in other environments.

Historical data maintained in LICON data sets

Integrity Checker stores historical copies of RDEs in the LICON data sets. RDEs record database access logs for database update, load, and unload applications.

Historical copies of DBD definition

Integrity Checker maintains historical copies of RDEs. Each of these copies contains a part of the database description (DBD) that is used in DMB verifications. Whenever a DBD is changed, Integrity Checker creates a new RDE that contains the latest DMB information. The historical copies of RDEs can be used to track the changes made to DBDs.

The content of the current RDE and historical copies of RDEs can be printed in reports. You can use the reports to examine the information in DBDs, review the changes made to DBDs, or to compare the difference between the current DBD and the DBDs used in the past.

To print the content of an RDE in a report, use the LIST.DB command of the LICON utility. For more information, see the following topics:

- To print a report, see the topic "LIST.DB command" in the *IMS Library Integrity Utilities User's Guide*.
- For report field descriptions, see the topic "Output from Integrity Checker" in the *IMS Library Integrity Utilities User's Guide*.

Database access recording option

Integrity Checker records the time stamp of the last database access that is made by update, load, and unload applications in the RDE together with the IMS subsystem name. Such information can be printed in a report, which can be used as evidence in database auditing.

To record database accesses, make the specifications in the RDE by using the global option module or the LICON utility. To use the database access recording option, specify the option for each of the following database access types:

Database access	Access type identified by Integrity Checker
Load utility	Load access
Reorganization utility	Load access
Unload utility	Unload access
Recovery utility	Update access
Utility or application program with the PCB processing option (PROCOPT) of A, I, R, or D	Update access

For instructions to specify these options, see the following topics:

- To set the options in the global option module, see "Global option module generation macro" on page 68.
- To update the options in RDEs, see the topic "INIT.DB command" in the *IMS Library Integrity Utilities User's Guide*.

When Integrity Checker is activated in an IMS online environment, Integrity Checker records access information only for the first database authorization request, and the recorded time is not updated while the IMS online environment is active. To have the recorded information updated while the IMS online environment is active, the ACCESS parameter of the DATABASE macro statement must specify UP or EX (during the system definition stage), or the parameter for the /START DB ACCESS= command must specify UP or EX.

Restrictions: The database access recording option can be used for the utilities or the application programs that are supported by Integrity Checker. However, the following restrictions apply:

- This option is not effective for online reorganization functions or online reorganization utilities that are provided by IMS or IMS Tools.
- For a database access through the High-Speed DEDB Direct Reorganization utility, Integrity Checker identifies the database access type as a database update.
- Database access for load or unload operations is recorded only when the operations are done by IMS standard utilities or IMS Tools utilities.
- If a utility or an application program ends with an error after the Integrity Checker DMB verification process ends successfully, the database access information that is recorded in the RDE is updated in the same way as when the utility or the application program ended successfully, even though the utility or application program ended with an error.
- For index databases or secondary index databases that have no PCBs, Integrity Checker does not record database access information in the RDEs except when the database operation is done by IMS Database Reorganization Expert.

Considerations for activating Integrity Checker

Before you activate Integrity Checker, review these considerations.

Cases where DMB verification is not done
   DMB verification is not done in certain environments or for certain application jobs.

   For more information, see the topic "Restrictions: Cases where DMB verification is not done" in the *IMS Library Integrity Utilities User's Guide*.

Size of the LICON data sets
   Before creating LICON data sets, you can estimate the required storage for the LICON data sets.

   For more information, see "Estimating the size of the LICON data set" on page 56.

LICON data set serialization consideration
   If you want more than one MVS system to access the LICON data set, you must serialize the LICON data set. To access the LICON data set from more than one MVS system, Global Resource Serialization (GRS) or a similar global enqueue product must be installed.

   For more information, see "Serializing the LICON data set" on page 61.

RACF security considerations
   If you want to protect the LICON data sets with RACF or if you plan to use Integrity Checker in IMS Database Recovery Facility jobs, you must modify RACF security.

   For more information, see "Setting up RACF security" on page 61.

Considerations when you alter the definition of an online HALDB or an online DEDB
   When you alter the definition of an online HALDB database or an online DEDB database with the following IMS command or IMS utility, you must temporarily stop the DMB verification process.

   - Use the INITIATE OLREORG command (with the ALTER option) and the online change (OLC) function to alter the definition of an online HALDB database.
   - Use the DEDB Alter utility (DBFUDA00) to alter the definition of an online DEDB database.

   After you alter the definition, restart the DMB verification process with new RDEs. Without these steps, the DMB verification process uses the old definitions to verify the DMBs. Therefore, the DMB verification process might deny database authorization requests that use correct IMS control blocks.

   For detailed instructions, see the following topics in the *IMS Library Integrity Utilities User's Guide*:

   - "Altering the definition of a DMB verification-enabled online HALDB by using the HALDB alter function"
   - "Altering the definition of a DMB verification-enabled online DEDB by using the DEDB Alter utility (DBFUDA00)"

Consideration when you change the maximum size of OSAM data sets for a HALDB
   When you change the maximum size of OSAM data sets for a HALDB from 4 GB to 8 GB or 8 GB to 4 GB, you must use the LICON utility and manually create RDEs to reflect the change for the DMB verification process.

   For more information, see the topic "Changing the maximum OSAM data set size for a DMB verification-enabled HALDB" in the *IMS Library Integrity Utilities User's Guide*.

Estimating the size of the LICON data set

Before you create LICON data sets, you can estimate the required storage for the LICON data sets.

About this task

Two methods are available for estimating the LICON data set size. The first method is more complicated but results in a precise data set size. The other method is simpler, but results in only an approximate data set size. You can use the simpler estimation method if the number of database segments is less than 20.

Tips:

- To accommodate any future increase in the number of databases, partitions, or DEDB areas, make the LICON data set size larger than the value that you calculate. For example, when you allocate a LICON data set, increase the calculated primary allocation size by 10% to 20%, and increase the secondary allocation size by approximately 10% of the primary allocation size.
- When you create multiple LICON data sets, you can create them on the same volume or on different volumes.

Procedure

For each LICON data set, use one of the following methods to estimate the data set size:

Estimating the precise LICON data set size

Use the following formula to calculate the precise LICON data set size:

```
LICON data set size = total_size_required_for_full-function_databases
                    + total_size_required_for_HALDBs
                    + total_size_required_for_DEDBs
```

total_size_required_for_full-function_databases
The total size of all the full-function databases that are to be processed.

For each full-function database, calculate the required size by using the following formula:

```
(450 + 32 * number_of_DSGs + 52 * number_of_segments
 + 180 * access_info_size) * 16
```

Note: DSG stands for data set group.

total_size_required_for_HALDBs
The total size of all the HALDB partitions that are to be processed.

For each HALDB partition, calculate the required size by using the following formula:

```
(450 + 32 * number_of_DSGs + 52 * number_of_segments
 + 180 * access_info_size) * 16
```

total_size_required_for_DEDBs
The total size of all the DEDB areas that are to be processed.

For each DEDB area, calculate the required size by using the following formula:

```
(220 + 48 * number_of_segments + 180 * access_info_size) * 16
```

Notes:

- If you do not enable the database access recording option (activated by the RECUPD, RECLD, or RECUL keyword in the global option module), *access_info_size* is 0. When this option is enabled, *access_info_size* increases by 1 for each keyword that you specify. For example, if you enable the RECUPD option (one keyword), *access_info_size* is 1. If you enable RECUPD and RECLD options (two keywords), *access_info_size* is 2.

  Related reading: For instructions to activate this option, see "Global option module generation macro" on page 68.
- For these formulas, the units are in bytes.
- Each formula is multiplied by 16 for storing histories of database definitions.

Estimating the approximate LICON data set size

Use the following formula to calculate the approximate LICON data set size:

```
LICON data set size = (32 * number_of_full-function_databases
                    + 32 * number_of_HALDB_partitions
                    + 16 * number_of_DEDB_areas) KB
```

number_of_full-function databases
  The number of full-function databases (excluding HALDBs) that are to be processed.

number_of_HALDB_partitions
  The number of HALDB partitions that are to be processed.

number_of_DEDB_areas
  The number of DEDB areas that are to be processed.

Activating Integrity Checker

Activating Integrity Checker involves preparing the global option modules, LICON data sets, and LIU load modules, and other steps.

Before you begin

Before you activate Integrity Checker, plan an Integrity Checker configuration for your environment, as described in "Planning for Integrity Checker configuration" on page 46.

About this task

To activate Integrity Checker, you first prepare the global option modules and the LICON data sets. Then, by using the DBD library, ACB library, or IMS directory, register the DMB information for your databases in the LICON data set. When this registration is done, customize the LIU load modules by using either of the following methods:

- Create an alias of the DSPCRTR0 module and add DD statements to JCL or procedures of DBRC, IMS batch application, IMS utility, and IMS Tools (recommended method).
- Merge load modules into the IMS SDFSRESL library.

Finally, activate Integrity Checker by restarting IMS online and running IMS batch applications, IMS utilities, and IMS Tools jobs.

In both IMS batch and online environments, the steps to activate Integrity Checker are the same. These steps can be applied when you reactivate Integrity Checker after Integrity Checker is deactivated.

Procedure

To activate Integrity Checker, complete the following steps:

a. "Setting up the global option modules" on page 58
b. "Setting up the LICON data sets" on page 59
c. "Setting up RACF security" on page 61
d. "Customizing LIU load modules" on page 63
e. "Configuring for a BPE-based DBRC environment" on page 67
f. "Verifying that Integrity Checker is activated" on page 67
g. "Restarting IMS online and running IMS batch application, IMS utility, and IMS Tools jobs" on page 68

Setting up the global option modules

Create global option modules and, optionally, assign alias names to the global option modules. At least one global option module must be present when Integrity Checker is started because Integrity Checker uses it to obtain the name of the LICON data set that it uses.

Before you begin

See "Integrity Checker configuration requirements" on page 48 and determine the number of global option modules to create. Also, determine whether to share a global option module across multiple IMS subsystems.

Procedure

1. Create global option modules.

   To create global option modules, use the FABLPGEN procedure and provide the control statements by using the SYSIN input stream. The FABLPGEN procedure is in the SHPSSAMP data set. For an instruction, see "Global option module generation macro" on page 68.

   Tip: To avoid activating Integrity Checker until all of your installation activities are complete, have no global option modules created.

   Related reading: The topic "Options applied to RDEs when multiple global option modules exist with different effective ranges" in the *IMS Library Integrity Utilities User's Guide*.

2. If you want more than one IMS subsystems to use a set of options that are defined in a single global option module, use the linkage editor to assign an alias name to the global option module.

   In a database sharing environment where multiple IMS subsystems share databases, the LICON data set and option values that are defined in the global option module must be the same across the IMS subsystems. Assigning aliases is beneficial in such an environment as well as in XRF environments.

   Use the following job to assign an alias name to the global option module.

```
//LKED    JOB
//L       EXEC PGM=IEWL,PARM='XREF,LIST'
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),
//            SPACE=(1024,(100,10),RLSE),DISP=(,DELETE)
//SYSLMOD  DD DISP=SHR,DSN=HPS.SHPSLMD0
//SYSLIN   DD *
 INCLUDE  SYSLMOD(LIU@IMSA)  /* Global option module for IMSID=IMSA */
          ALIAS  LIU@IMSB    /* Alias for IMSID=IMSB              */
          NAME   LIU@IMSA    /* Original name for IMSID=IMSA      */
/*
```

Setting up the LICON data sets

Define and initialize LICON data sets, create RDEs to register the correct DMB information, and serialize the LICON data sets. At least one LICON data set must be present to activate Integrity Checker.

Before you begin

See "Integrity Checker configuration requirements" on page 48 to determine the number of LICON data sets to create.

Procedure

The following steps describe how to set up a single LICON data set. If more than one LICON data set is required for your environment, complete the following steps for each LICON data set.

a. "Defining and initializing the LICON data set" on page 59
b. "Creating an RDE to register DMB information" on page 60
c. "Serializing the LICON data set" on page 61

Defining and initializing the LICON data set
Define the LICON data set by using the DEFINE cluster command, and initialize it by using the INIT.LICON command of the LICON utility.

Procedure

1. Define the LICON data set by using the DEFINE CLUSTER command.

   Sample JCL is in the SHPSJCL0 library, member FABLINIT. The following figure shows the DEFLICON step of the sample JCL, which defines the LICON data set.

Specify the values for the following parameters. For the CYL parameter, specify the size that you calculated in "Estimating the size of the LICON data set" on page 56.

```
//DEFLICON EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE  (imshlq.licondsn) CLUSTER ERASE PURGE
  SET MAXCC=0
  DEFINE CLUSTER (NAME(imshlq.licondsn) -
         INDEXED -
         KEY (44 0)  -
         SHR(3 3) -
         NOREUSE -
         VOL (liconvol) -
         CYL (pri sec) -
         RECSZ (4096 32760) -
         FREESPACE (xx xx)) -
         DATA(NAME(imshlq.licondsn.DATA)) -
         INDEX(NAME(imshlq.licondsn.INDEX))
  /*
```

2. Initialize the LICON data set by using the INIT.LICON command of the LICON utility.

Sample JCL is in the SHPSJCL0 library, member FABLINIT. The following figure shows the INILICON step of the sample JCL, which initializes the LICON data set.

Add the load module data sets that contain the global option module and the LIU load module library to the STEPLIB concatenation.

```
//INILICON EXEC PGM=FABLIU00
//STEPLIB  DD DISP=SHR,DSN=HPS.SHPSLMD0
//         DD DISP=SHR,DSN=IMSVS.SDFSRESL
//FABLPRNT DD SYSOUT=*
//FABLIN   DD *
  INIT.LICON
  /*
```

Creating an RDE to register DMB information
Create an RDE to register the correct DMB information by using the LICON utility.

About this task

This task is optional because if you do not create an RDE with the LICON utility, Integrity Checker creates an RDE automatically when the database is accessed for the first time after Integrity Checker is activated. In this case, Integrity Checker creates an RDE for each database that is accessed while IMS is online. Therefore, the performance of IMS online processing might decline depending on the number of databases. Consider creating RDEs manually by using the LICON utility to avoid performance degradation.

If you want Integrity Checker to create RDEs automatically, you can skip this task.

Procedure

Use the LICON utility to create an RDE.

Ensure that you provide the following information with the JCL statements:

- Specify the DBD library, the ACB library, or the IMS directory that contains the DMB information to be used in DMB verifications.
- To verify the user exit routine, add the data set that contains the user exit routine to be used in DMB verifications to the STEPLIB concatenation.

Provide the INIT.DB command in the FABLIN input stream. The RDE is created in the LICON data set that is specified by the global option module found in the STEPLIB concatenation.

The following JCL example is for the LICON utility. `INIT.DB DBD(*)` specifies that an RDE is created for every DBD member in the DBD library IMSVS.DBDLIB.

```
//LICJOB    JOB
//          EXEC PGM=FABLIU00
//STEPLIB   DD DISP=SHR,DSN=HPS.SHPSLMD0
//          DD DISP=SHR,DSN=IMSVS.SDFSRESL
//DBDLIB    DD DISP=SHR,DSN=IMSVS.DBDLIB
//FABLPRNT  DD SYSOUT=*
//FABLIN    DD *
  INIT.DB DBD(*)
/*
```

Serializing the LICON data set
This step is optional. If you want more than one MVS system to access the LICON data set, you must serialize the LICON data set.

Procedure

To access the LICON data set from more than one MVS system, you must install Global Resource Serialization (GRS) or a similar global enqueue product. GRS processes the resource as a global resource. The global enqueue product must propagate the enqueues to all MVS hosts. If the enqueues are not propagated to all hosts that have access to the LICON data set, the data set becomes corrupted and unusable.

The LICON data set is enqueued with the following parameters at the SYSTEMS level:

Parameter	Value
QNAME	'FABLICON'
RNAME	The name of the LICON data set

Setting up RACF security

You can optionally set up RACF security to protect LICON data sets. If you plan to activate Integrity Checker in IMS Database Recovery Facility jobs and if DBRC command and API request authorization support is enabled with RACF, ensure that appropriate permission is given to users.

Procedure

Complete the following steps to set up RACF security:

- "Setting up security for LICON data sets" on page 61
- "Setting up security for IMS Database Recovery Facility jobs" on page 62

Setting up security for LICON data sets
This step is optional. If you want to protect the LICON data sets with RACF, complete this step.

About this task

The following procedure provides a brief overview of the security setting for LICON data sets.

For more information about the security setting of the started tasks, see the topics that describe how to associate started procedures and jobs with user IDs in the *z/OS Security Server RACF System Programmers Guide* and in the *z/OS Security Server RACF Security Administrators Guide*.

Procedure

To use RACF to protect the LICON data sets, define data set profiles for the LICON data sets. Also, because LICON data sets are accessed in each environment, consider the following requirements in both IMS online environment and batch environment.

IMS online environment

For IMS online processing, access to the LICON data set is done by the DBRC region. In this case, the DBRC region must be assigned a user ID, preferably by using the RACF STARTED class. This assignment can also be done by using the RACF started task table (ICHRIN03) or the USER= and PASSWORD= values on the DBRC job or the task JCL. Give UPDATE access authority to the LICON data set for the user ID assigned to the DBRC region.

The following list contains an example of the statements that are used to create and enable LICON data set protection by using RACF.

```
AG licongrp
AU liconusr DFLTGRP(licongrp)
RDEF STARTED dbrcrgn.* STDATA( USER(liconusr) GROUP(licongrp))
ADDSD liuhlq.licon UACC(NONE)
PE liuhlq.licon ID(licongrp) ACCESS(UPDATE)
```

AG *licongrp*
   Creates a RACF group named *licongrp*.

AU *liconusr* DFLTGRP(*licongrp*)
   Creates a RACF user ID of *liconusr* and assigns the default group *licongrp*.

RDEF STARTED *dbrcrgn.* STDATA(USER(*liconusr*) GROUP(*licongrp*))
   Defines the STARTED class profile, which will assign the user ID to the *dbrcrgn* procedure. This statement assumes that the PDS member name of the procedure that is started is *dbrcrgn*. This statement assigns user ID *liconusr* to the started procedure.

ADDSD *liuhlq.licon* UACC(NONE)
   Defines a data set profile for the LICON data set.

PE *liuhlq.licon* ID(*licongrp*) ACCESS(UPDATE)
   Gives the *licongrp* group UPDATE access to the LICON data set.

Batch environment

The user ID created for the DBRC online region cannot be used because jobs can be submitted by many different users. The user ID will be used if the ID is correctly assigned through the use of the RACF STARTED class by coding entries for job names in the STARTED class and associating those entries with the user ID. However, the STARTED class is used only when the batch jobs are started with the MVS START command. Jobs submitted to a JES reader will not invoke a call to the STARTED class. Therefore, they might not be assigned a user ID that allows appropriate access to the LICON data set. In this case, all users authorized to run batch jobs must be identified and connected to a group that has UPDATE access to the LICON data set.

Setting up security for IMS Database Recovery Facility jobs

If you plan to activate Integrity Checker in IMS Database Recovery Facility jobs, complete this step.

Procedure

When you activate Integrity Checker in an IMS Database Recovery Facility job, Integrity Checker uses the DBRC command utility and the DBRC API. If you protect the DBRC command and DBRC API request authorization support with RACF by permitting appropriate user access to the profiles, you must permit appropriate users of jobs to use the following DBRC commands and DBRC API requests:

DBRC command
   LIST.RECON

DBRC API requests

- STARTDBRC
- STOPDBRC
- RELBUF
- QUERY,TYPE=DB
- QUERY,TYPE=PART

For more information about the RACF settings for DBRC commands and DBRC API requests, see the topic "DBRC security" in *IMS System Administration*.

Customizing LIU load modules

Customize the LIU load modules to activate Integrity Checker.

Before you begin

See <u>"LIU load module library customization" on page 46</u> to determine which method you use to customize the LIU load modules.

Procedure

Customize the LIU load modules by using either of the following methods:

- <u>"Method 1. Customizing LIU load modules by creating alias name DSPCRTR0" on page 63</u>
- <u>"Method 2. Customizing LIU load modules by merging into the IMS SDFSRESL library" on page 65</u>

Method 1. Customizing LIU load modules by creating alias name DSPCRTR0

Create an alias name DSPCRTR0, APF-authorize the LIU load module library, and add DD statements to JCL and procedures of DBRC, IMS batch applications, IMS utilities, and IMS tools.

Procedure

1. Create alias name DSPCRTR0 by link-editing the FABLRTR0 load module.

   Complete this step if either of the following conditions apply:

   - You are activating Integrity Checker for the first time.
   - You removed the alias name DSPCRTR0 by completing the steps in the topic "<u>Deactivating Integrity Checker when IMS Library Integrity Utilities is installed as a component of an IMS solution pack</u>" in the *IMS Library Integrity Utilities User's Guide* and you want to reactivate Integrity Checker.

   If you are reactivating Integrity Checker and the DSPCRTR0 alias that was created in the last activation still exists, you can skip this step.

   You can use the following JCL example to create alias name DSPCRTR0. This JCL is in the SHPSJCL0 library, member FABLALSC.

   When the job ends, confirm that the return code is 0.

   ```
   //FABLALSC JOB
   //*------------------------------------------------------------------
   //*   STEP1: Add the alias DSPCRTR0
   //*------------------------------------------------------------------
   //LINK    EXEC PGM=IEWL,REGION=0M,
   //   PARM='SIZE=(880K,64K),LET,LIST,NCAL,RENT,REFR,XREF'
   //* IEWL = IEWBLINK
   //*
   //SYSPRINT DD  SYSOUT=*
   //SYSLMOD  DD  DISP=OLD,DSN=LIU.SHPSLMD0  LIU target load module lib
   //SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(10,1))
   //* CAUTION!!!
   //*   SPECIFY SHPSLMD0 TO THE INCLUDE STATEMENT.
   //*   IF YOU SPECIFY AHPSMOD0 TO THE INCLUDE STATEMENT,
   //*   THE NON-ACCEPTED CHANGES WILL BE DELETED.
   //SYSLIN   DD  *
     ENTRY    FABLRTR0
     INCLUDE  SYSLMOD(FABLRTR0)
     ALIAS    DSPCRTR0
     NAME     FABLRTR0(R)
   /*
   ```

2. APF-authorize the LIU load module library.

The LIU load module library and the load module data sets that contain the global option modules must be APF-authorized.

3. Add DD statements to JCL and procedures of DBRC, IMS batch applications, IMS utilities, and IMS tools.

You must add DD statements to JCL and cataloged procedures for all the jobs that you want to activate Integrity Checker in. These JCL and procedures include those for DBRC, IMS batch applications, IMS utilities, and IMS Tools jobs that update IMS databases.

Important: For integrity, ensure that all JCL and procedures that might change databases meet the following STEPLIB DD requirements.

Add the following DD statements:

STEPLIB DD
Add the load module data sets that contain the global option module and the LIU load module library to the STEPLIB concatenation.

The LIU load module library must be concatenated before the IMS load module library and must be APF-authorized.

If you want Integrity Checker to detect changes in the logic of IMS user exits, which include randomizing routines, segment edit/compression exit routines, and HALDB or DEDB partition selection exit routines, also include the exit load modules in the STEPLIB concatenation. Integrity Checker does not check the user exit load modules in the LPA, ELPA, or LNKLST.

FABLPRNT DD
Optionally, you can specify the FABLPRNT DD statement in your procedures. This statement causes Integrity Checker to generate messages in the DD.

If this statement is specified, Integrity Checker writes messages into this DD in addition to issuing the WTO macro. Each message contains a time stamp in its prefix, and you can easily identify the messages in relation to the authorization request from your application programs.

FABLSNAP DD
Optionally, you can specify the FABLSNAP DD statement in your online DBRC procedure. This statement causes Integrity Checker to generate diagnostic information for the VSAM control blocks when Integrity Checker gets a VSAM error. For DL/I batch jobs, you do not need to specify this DD statement because the jobs issue an abend dump when they get a VSAM error.

Example

The following figures show examples of the procedures.

```
//         PROC RGN=64M,DPTY='(14,15)',SOUT=A,
//              IMSID=SYS3,SYS2=,IMSPLEX=
//IEFPROC EXEC PGM=DFSMVRC0,REGION=&RGN,
//              DPRTY=&DPTY,PARM='DRC,&IMSID,IMSPLEX=&IMSPLEX'
//****************************************************
//*
//STEPLIB  DD DSN=HPS.SHPSLMD0,DISP=SHR      <<---- STEPLIB DD
//         DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//PROCLIB  DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
//JCLOUT   DD SYSOUT=(A,INTRDR)
//JCLPDS   DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
//SYSUDUMP DD SYSOUT=&SOUT
//SYSABEND DD SYSOUT=&SOUT
//FABLPRNT DD SYSOUT=&SOUT                   <<---- FABLPRNT DD
//FABLSNAP DD SYSOUT=&SOUT                   <<---- FABLSNAP DD
```

Figure 7. Example of DBRC procedure for a non-BPE-based DBRC region

```
//DBRC     PROC RGN=0M,SOUT=A,
//              RESLIB='IMS.SDFSRESL',
//              BPECFG=BPECONFG,
//              DBRCINIT=000,
//              IMSID=IMS1,
//              PARM1='BPEINIT=DSPBINI0'
//*
//DBRCPROC   EXEC PGM=BPEINI00,REGION=&RGN,
// PARM='BPECFG=&BPECFG,DBRCINIT=&DBRCINIT,IMSID=&IMSID,&PARM1'
//*
//STEPLIB  DD DSN=HPS.SHPSLMD0,DISP=SHR      <<---- STEPLIB DD
//           DD DSN=&RESLIB,DISP=SHR
//           DD DSN=SYS1.CSSLIB,DISP=SHR
//PROCLIB  DD DSN=IMS.PROCLIB,DISP=SHR
//SYSPRINT DD SYSOUT=&SOUT
//SYSUDUMP DD SYSOUT=&SOUT
//JCLOUT   DD SYSOUT=(A,INTRDR)
//JCLPDS   DD DSN=IMS.PROCLIB,DISP=SHR
//SYSABEND DD SYSOUT=&SOUT
//FABLPRNT DD SYSOUT=&SOUT                   <<---- FABLPRNT DD
//FABLSNAP DD SYSOUT=&SOUT                   <<---- FABLSNAP DD
```

Figure 8. Example of DBRC procedure for a BPE-based DBRC region

```
//        PROC MBR=TEMPNAME,PSB=,BUF=7,
//              SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,
//              SRCH=0,CKPTID=,MON=N,LOGA=0,FMTO=T,
//              IMSID=,SWAP=,DBRC=,IRLM=,IRLMNM=,
//               BKO=N,IOB=,SSM=,APARM=,
//              RGN=4M,
//              SOUT=A,LOGT=2400,SYS2=,
//              LOCKMAX=,GSGNAME=,TMINAME=,
//              IMSPLEX=
//G       EXEC PGM=DFSRRC00,REGION=&RGN,
//              PARM=(DLI,&MBR,&PSB,&BUF,
//              &SPIE&TEST&EXCPVR&RST,&PRLD,
//              &SRCH,&CKPTID,&MON,&LOGA,&FMTO,
//              &IMSID,&SWAP,&DBRC,&IRLM,&IRLMNM,
//              &BKO,&IOB,&SSM,'&APARM',
//              &LOCKMAX,&GSGNAME,&TMINAME,
//              &IMSPLEX)
//STEPLIB  DD DSN=HPS.SHPSLMD0,DISP=SHR      <<---- STEPLIB DD
//           DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//           DD DSN=IMS.&SYS2.PGMLIB,DISP=SHR
//DFSRESLB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//IMS      DD DSN=IMS.&SYS2.PSBLIB,DISP=SHR
//           DD DSN=IMS.&SYS2.DBDLIB,DISP=SHR
//PROCLIB  DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
//IEFRDER  DD DSN=IMSLOG,DISP=(,KEEP),VOL=(,,,99),
//           UNIT=(&LOGT,,DEFER),
//           DCB=(RECFM=VB,BLKSIZE=4096,
//           LRECL=4092,BUFNO=2)
//IEFRDER2 DD DSN=IMSLOG2,DISP=(,KEEP),VOL=(,,,99),
//           UNIT=(&LOGT,,DEFER,SEP=IEFRDER),
//           DCB=(RECFM=VB,BLKSIZE=4096,
//           LRECL=4092,BUFNO=2)
//SYSUDUMP DD SYSOUT=&SOUT,
//           DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
//           SPACE=(605,(500,500),RLSE,,ROUND)
//IMSMON   DD DUMMY
//FABLPRNT DD SYSOUT=&SOUT                   <<---- FABLPRNT DD
```

Figure 9. Example of DLIBATCH procedure

Method 2. Customizing LIU load modules by merging into the IMS SDFSRESL library

Back up the IMS SDFSRESL library, and then merge the LIU load modules into the IMS SDFSRESL library.

Procedure

1. Back up the IMS SDFSRESL library.

   When program temporary fixes (PTFs) are released for the DSPCRTR0 module, you must restore the DSPCRTR0 module from the backup to apply the PTFs. Therefore, before merging LIU load modules,

you must create a backup of the IMS SDFSRESL library. The backup is also required to deactivate Integrity Checker.

2. Use SMP/E to apply and accept IMS and IMS Library Integrity Utilities maintenance, and ensure that both are at the latest maintenance level.

3. Run the FABLUMD1 job that is in the SHPSJCL0 JCL library.

   This job updates the SMP/E CSI of IMS. It runs SMP/E RECEIVE/APPLY of USERMOD to install the FABLRTR0 module into the IMS SDFSRESL library. The FABLUMD1 job is shown in "JCL example to install the FABLRTR0 module into the IMS SDFSRESL library" on page 66.

4. Merge the LIU load modules (FABL* members) in the target library SHPSLMD0 into the IMS SDFSRESL library.

   If this step is not done, when Integrity Checker is activated, an ABENDU0109 load failure occurs for the required LIU load modules.

JCL example to install the FABLRTR0 module into the IMS SDFSRESL library

```
//FABLUMD1 JOB
//*-------------------------------------------------------------------
//*   STEP1: SMP/E RECEIVE/APPLY usermod ZZLIU01 to IMS CSI
//*-------------------------------------------------------------------
//STEP1    EXEC procedure name of IMS SMP/E job
//*
//*
//AHPSMOD0 DD DISP=SHR,DSN=LIU.AHPSMOD0
//*
//SMPPTFIN DD DATA,DLM=@@
++USERMOD(ZZLIU01)                       /*  LIU R2 USERMOD         */
      REWORK(2004058)                    /*                         */
 /*                                                                 */
 /* OPTIONAL LIU USERMOD FOR IMS-DBRC FMIDS.                        */
 /* APPLY THIS USERMOD TO IMS-DBRC SMP/E CSI ONLY IF YOU WANT TO    */
 /* INSTALL LIU-MODIFIED VERSION OF DSPCRTR0 INTO YOUR IMS SMP/E    */
 /* CSI.                                                            */
 /*                                                                 */
 /* BEFORE APPLYING THIS USERMOD, ALL MAINTENANCE FOR IMS-DBRC      */
 /* DSPCRTR0 MUST BE ACCEPTED OR RESTORED.                          */
 /*                                                                 */.
++VER(P115)                              /*  IMS SYSTEM ID/FMID     */
      FMID(FMID of IMS)
                                         /*                         */.
++JCLIN CALLLIBS                         /*  JCLIN FOR LIU MODULES  */
                                         /*                         */.
//LINK    EXEC PGM=IEWL,REGION=0M,
//    PARM='SIZE=(880K,64K),LET,LIST,NCAL,RENT,REFR,XREF'
//SYSPRINT DD  SYSOUT=A
//AHPSMOD0 DD  DISP=OLD,DSN=LIU.AHPSMOD0
//ADFSLOAD DD  DISP=OLD,DSN=IMS.ADFSLOAD
//SYSLMOD  DD  DISP=OLD,DSN=IMS.SDFSRESL
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(10,1))
//SYSLIN   DD  *
  INCLUDE AHPSMOD0(FABLRTR0)
  INCLUDE ADFSLOAD(DSPCRTR0)
  ENTRY   FABLRTR0
  NAME    DSPCRTR0(R)
/*
++MOD (FABLRTR0) LKLIB(AHPSMOD0)     /*  LIU MODULE MOD ENTRY    */.
@@
//SMPCNTL  DD *
  SET     BDY (GLOBAL).
  RECEIVE S   (ZZLIU01) SYSMODS.
  SET     BDY (TZONE name of IMS).
  APPLY   S   (ZZLIU01).
/*
```

Configuring for a BPE-based DBRC environment

To activate Integrity Checker in an IMS online environment that has a BPE-based DBRC region, you must configure the members of the IMS PROCLIB data set. This step is required only when a BPE-based DBRC is used.

Procedure

1. Set up the BPE configuration parameter member.

   Specify the DBRC user exit list member by using the EXITMBR statement in the BPE configuration parameter member. The BPE configuration parameter member is specified by the BPECFG= keyword in the DBRC procedure that is used for the BPE-based DBRC region in which you want to activate Integrity Checker.

   You can skip this step if the EXITMBR statement for the DBRC user exit list member already exists.

   The following example specifies the EXITMBR statement in a BPE configuration parameter member:

   ```
   #
   # User exit list PROCLIB member specification
   #
   EXITMBR=(member_name,DBRC) /* DBRC user exit list member */
   ```

   For more information about the EXITMBR statement, see the topic "BPE configuration parameter member of the IMS PROCLIB data set" in *IMS System Definition*.

2. Set up the DBRC user exit list member.

   Specify the Integrity Checker load module FABLBIN0 on the EXITDEF statement. The FABLBIN0 module must be specified as a DBRC Request exit in the DBRC user exit list member. The DBRC user exit list member is specified by the EXITMBR statement for DBRC in the BPE configuration parameter member.

   • When you have two or more DBRC user exits, the FABLBIN0 module must be specified as the first member on the EXITDEF statement.

   • Do not specify the ABLIM parameter on the EXITDEF statement.

   The following example specifies the Integrity Checker load module on the EXITDEF statement:

   ```
   **********************************************************************
   * DBRC USER EXIT LIST PROCLIB MEMBER                                 *
   **********************************************************************
   #----------------------------------------------------------------#
   # DEFINE a DBRC request user exit.                               #
   #----------------------------------------------------------------#
   EXITDEF(TYPE=REQUEST,EXITS=(FABLBIN0),COMP=DBRC)
   ```

   For information about the EXITDEF statement, see the topic "BPE exit list members of the IMS PROCLIB data set" in *IMS System Definition*.

Verifying that Integrity Checker is activated

IMS Library Integrity Utilities provides sample JCL for verifying successful activation of Integrity Checker. You can modify the sample JCL and then use it to ensure that Integrity Checker is running correctly.

About this task

This task is optional. Complete this task only if you want to ensure that Integrity Checker is activated.

Procedure

In the SHPSJCL0 library, locate sample JCL member FABLIVP3. Modify the sample JCL by following the instructions in the sample JCL and submit the JCL. Ensure that the job ends without errors.

Restarting IMS online and running IMS batch application, IMS utility, and IMS Tools jobs

When you have done all the steps, you are ready to restart IMS online to activate the DMB verification process.

Procedure

1. Restart IMS online and run IMS batch application jobs, IMS utility jobs, and IMS Tools jobs.
2. Confirm that the DMB verification process is activated by locating the following WTO message:

```
FABL0114I LIU INTEGRITY CHECKER ACTIVATED. IMS VERSION IS version
```

Global option module generation macro

Use the global option module generation macro to create global option modules.

Creating global option modules

To activate Integrity Checker, create at least one global option module that contains the name of the LICON data set. If you want to change the default options of Integrity Checker globally, specify them when you create this module.

About this task

IBM does not supply global option modules. You must create at least one global option module before invoking Integrity Checker.

Procedure

To create a global option module, determine the type of the global option module, then use the FABLPGEN procedure (provided in the SHPSSAMP data set) to create it. Runtime options can be defined by using the SYSIN control statements.

There are two levels of global option modules that can be categorized by their effective range: installation level and IMS subsystem level.

Installation level
> To set values that will be effective at the installation level, create a global option module named LIU@INST. The values in this module apply to all the databases that are defined in the IMS environment.

IMS subsystem level
> To set values that will be effective at the IMS subsystem level, create a global option module named LIU@*imsid*, where *imsid* is the 4-character ID of the IMS subsystem. The values that you set in this module apply to all the databases that are defined to that IMS subsystem. In accordance with the options assignment rule, they override the values that are set in the LIU@INST module.
>
> **Tip:** If you want more than one IMS subsystems to use a set of options that are defined in a single global option module, create a global option module for the IMS subsystem level and use the linkage editor to assign an alias to that global option module.
>
> In a database sharing environment where more than one IMS subsystem shares databases, the LICON data set and option values defined in the global option module must be the same across the IMS subsystems. Assigning an alias is beneficial in such a case as well as in XRF environments. For

information about how to assign an alias name, see "Setting up the global option modules" on page 58.

In environments where a LICON data set is used across multiple IMS subsystems, assigning an alias name for the global option module to apply the same runtime options for all IMS IDs is a good practice. However, if you want to set runtime options for each IMS ID, instead of assigning an alias name to the global option module, you can create one global option module for each IMS ID. In such a case, except for certain control statement keywords, the keyword parameters must be the same. You can set different parameters for the following control statement keywords:

- VERIFY=
- MSGROUT=
- MSGDESC=
- VERIFYLMT=
- RDEBUILD=
- INITERR=

When you create global option modules, name the modules LIU@*xxxx*. At sign (@) is a code-page-dependent character. If you are working in an environment where you cannot use the at sign (@), name the modules LIUG*xxxx*. Use either format for all the global option modules consistently because maintaining both LIU@*xxxx* and LIUG*xxxx* modules can cause confusion. If Integrity Checker finds both LIU@*imsid* and LIUG*imsid* in the same effective range level, Integrity Checker ignores LIUG*imsid* and uses LIU@*imsid*.

JCL requirements for the FABLPGEN program

The following JCL requirements must be met to create a global option module with the FABLPGEN program.

Subsections:

- "EXEC statement" on page 69
- "DD statements" on page 69
- "Control statement keywords" on page 70
- "Example" on page 74

EXEC statement

The EXEC statement must be in the following form.

```
//stepname   EXEC FABLPGEN,MBR=module,SOUT=x
```

MBR=
Specifies the name of the global option module. *module* is LIU@INST, LIUGINST, LIU@*imsid*, or LIUG*imsid*.

SOUT=
Specifies the SYSOUT class to be used for SYSPRINT DD.

DD statements

Code the following DD statements to identify the source of input and the placement of output information:

SYSLIB DD
This statement specifies the macro library (SHPSMAC0) provided by IMS Library Integrity Utilities, or one of IMS tools solution packs. This library contains the FABLPGIN macro.

SYSIN DD
This statement specifies the input control statement stream.

SYSLMOD DD
This statement specifies the output data set for global option modules. If you merge the Integrity Checker load modules into the IMS SDFSRESL library, specify the IMS SDFSRESL library for this DD statement. Otherwise, specify the Integrity Checker load module library for this DD statement.

You can create global option modules in a different library. If you do so, concatenate that library to the STEPLIB DD in JCL and cataloged procedures for all the jobs from which you want to activate Integrity Checker.

Control statement keywords

The control statement formats are as follows:

FABLPGIN
The IBM supplied macro for use in defining the global option module. The syntax of the parameter specifications of this macro is the same as the syntax of an ordinary assembler macro statement.

You must specify the statement label for the FABLPGIN macro. For the statement label, specify the name of the global option module, which is LIU@INST, LIUGINST, LIU@*imsid*, or LIUG*imsid* (in the example in Figure 10 on page 74, LIU@INST beginning at column 1.)

LICON=
Specifies the name of the LICON data set. No system default value is provided for this parameter.

VERIFY=
Specifies the method for verifying the DMBs; either SNGL or DBLE. SNGL specifies single-step verification and DBLE specifies double-step verification. The system default value for this parameter is SNGL.

You can choose either of the following two options for how Integrity Checker verifies the DMB of a full-function database against the DMB information registered in the RDE. Specify your choice in the global option module.

Single-step verification
Integrity Checker verifies all elements at once. This method is more reliable than the double-step method, but slower.

Double-step verification
Integrity Checker verifies the version ID of the DMB in the first step. Version ID is the 13-character time stamp of when the DBD was created or the character string that is specified on the VERSION= keyword of the DBD statement that was supplied for DBDGEN. If the version IDs are not the same, Integrity Checker proceeds to the next step to verify other elements. This method is faster than the single-step method, but less reliable.

For DEDBs, single-step verification is always applied.

Recommendation: Specify single-step verification. If you experience a performance problem, consider using double-step verification.

MSGROUT=
Specifies the message routing codes for write-to-operator (WTO) messages issued by Integrity Checker. You can specify values in the range of 1 - 16. The system default value for this parameter is (2,7,11).

MSGDESC=
Specifies the message descriptor codes for write-to-operator (WTO) messages issued by Integrity Checker. The system default value for this parameter is (7).

VERIFYLMT=
Specifies the maximum number of the mismatch messages to be issued for a DMB. For example, specifying 3 means Integrity Checker does not issue more than three mismatch messages for a DMB.

You can specify any number in the range of 0 - 99. 0 specifies that verification is to be done but no mismatch message issued. 99 specifies that the number of messages is unlimited. The system default value for this parameter is 10.

RDEBUILD=

Specifies whether Integrity Checker automatically creates an RDE. The system default value for this parameter is Y.

Y

If no current RDE exists for a DEDB area, a non-HALDB full-function database, or a HALDB partition, Integrity Checker automatically creates an RDE during the first access to it.

N

Even when no current RDE exists for a DEDB area, a non-HALDB full-function database, or a HALDB partition, Integrity Checker does not create an RDE during the first access to it.

INITERR=

Specifies whether Integrity Checker abnormally ends, or issues a warning message and stops its processing, when the initialization of Integrity Checker fails. The system default value for this parameter is A.

This option is not effective in an IMS online subsystem that has a BPE-based DBRC region. If you specify this option for such an environment, Integrity Checker stops processing, and the IMS online subsystem continues processing.

A

If the initialization of Integrity Checker fails, it ends abnormally together with the IMS online subsystem that has a non-BPE-based DBRC region or the IMS batch job.

W

If the initialization of Integrity Checker fails, it issues a warning message and stops its processing. The IMS online subsystem or the IMS batch job continues processing without the Integrity Checker function. However, Integrity Checker ends abnormally if errors occur before the effective value for this option is decided. Such errors are load failures of the following modules:

- FABLRTR*x* (*x*: 8, 9, A, B, C, or D)
- FABLWM0
- FABLAI0
- Global option module

CHECKON=

Specifies the verification option for online IMS subsystems. The system default value for this parameter is (Y,D). You can specify the following options:

Y

Check. If you specify Y, you can specify either of the following parameters:

D

If a mismatch is found, deny authorization to use the database.

W

Issue a warning message and create a new RDE.

N

Do not check.

CHECKBAT=

Specifies the verification option for batch jobs. The system default value for this parameter is (Y,D). You can specify the following options:

Y

Check. If you specify Y, you can specify either of the following parameters:

D

If a mismatch is found, deny authorization to use the database.

W

Issue a warning message and create a new RDE.

N
>> Do not check.

CHECKLD=
> Specifies the verification option for user load program jobs. The system default value for this parameter is (Y,D). You can specify the following options:

> **Y**
>> Check. If you specify Y, you can specify either of the following parameters:

>> **D**
>>> If a mismatch is found, deny authorization to use the database.

>> **W**
>>> Issue a warning message and create a new RDE.

> **N**
>> Do not check.

CHECKIC=
> Specifies the verification option for batch image copy jobs. The system default value for this parameter is (Y,D). You can specify the following options:

> **Y**
>> Check. If you specify Y, you can specify either of the following parameters:

>> **D**
>>> If a mismatch is found, deny authorization to use the database.

>> **W**
>>> Issue a warning message and create a new RDE.

> **N**
>> Do not check.

CHECKRV=
> Specifies the verification option for database recovery jobs. The system default value for this parameter is (Y,D). You can specify the following options:

> **Y**
>> Check. If you specify Y, you can specify either of the following parameters:

>> **D**
>>> If a mismatch is found, deny authorization to use the database.

>> **W**
>>> Issue a warning message and create a new RDE.

> **N**
>> Do not check.

CHKRAND=
> Specifies whether to verify changes in randomizing routines by checksum. The system default value for this parameter is N. You can specify the following options:

> **Y**
>> Check.

> **N**
>> Do not check.

> If CHECKON, CHECKBAT, CHECKLD, CHECKIC, or CHECKRV is set to Y, this specification is effective in each IMS environment.

CHKCOMP=
> Specifies whether to verify changes in segment edit/compression routines by checksum. The system default value for this parameter is N. You can specify the following options:

> **Y**
>> Check.

N

    Do not check.

If CHECKON, CHECKBAT, CHECKLD, CHECKIC, or CHECKRV is set to Y, this specification is effective in each IMS environment.

CHKPSEL=

Specifies whether to verify changes in HALDB partition selection exit routines by checksum. The system default value for this parameter is N. You can specify the following options:

Y

    Check.

N

    Do not check.

If CHECKON, CHECKBAT, CHECKLD, CHECKIC, or CHECKRV is set to Y, this specification is effective in each IMS environment.

CHKFPSEL=

Specifies whether to verify changes in DEDB partition selection exit routines by checksum. The system default value for this parameter is N. You can specify the following options:

Y

    Check.

N

    Do not check.

If CHECKON, CHECKBAT, CHECKLD, CHECKIC, or CHECKRV is set to Y, this specification is effective in each IMS environment.

GENMAX=

Specifies the maximum number of expired RDEs to be kept in the LICON data set for use in recoveries. The system default value for this parameter is 0 (do not keep expired RDEs). A maximum of 15 RDE copies can be kept.

RECUPD=

Specifies whether to record database update access information. The system default value for this parameter is N. You can specify the following options:

Y

    Record.

N

    Do not record.

RECLD=

Specifies whether to record database load access information. The system default value for this parameter is N. You can specify the following options:

Y

    Record.

N

    Do not record.

RECUL=

Specifies whether to record database unload access information. The system default value for this parameter is N. You can specify the following options:

Y

    Record.

N

    Do not record.

Example

Sample JCL is in the SHPSJCL0 library, member FABLINIT. The following figure shows the CREGOM step of the sample JCL, which creates global option module LIU@INST.

The name of a global option module can be LIU@INST, LIU@*imsid*, LIUGINST, or LIUG*imsid*. Replace LIU@INST in the JCL example to create a module with one of these names.

```
//CREGOM  EXEC FABLPGEN,MBR=LIU@INST,SOUT=A
//C.SYSLIB  DD DISP=SHR,DSN=HPS.SHPSMAC0
//C.SYSIN   DD *
LIU@INST FABLPGIN VERIFY=SNGL,      single step verification        X
                  MSGROUT=(2,7,11),  WTO message rouing codes        X
                  MSGDESC=(7),       WTO message descritor codes     X
                  VERIFYLMT=5,       max number of mismatch messages X
                  CHECKON=(Y,D),     verification option - online    X
                  CHECKBAT=(Y,D),    verification option - batch     X
                  CHECKLD=(Y,D),     verification option - load      X
                  CHECKIC=(Y,D),     verification option - image copy X
                  CHECKRV=(Y,D),     verification option - recovery  X
                  GENMAX=3,          max number of expired RDE kept  X
                  LICON=imshlq.licondsn
        END
/*
//L.SYSLMOD DD DISP=SHR,DSN=HPS.USERLIB(&MBR)
```

Figure 10. JCL for creating a global option module LIU@INST

Setting up the ISPF interfaces

Three components of IMS Database Utility Solution provide ISPF interfaces: IMS HALDB Toolkit, IMS HP Pointer Checker (DB Historical Data Analyzer utility), and Database Repair Facility.

About this task

- If you want to use IMS HALDB Toolkit as an ISPF application or use the ISPF interface of IMS HALDB Toolkit, you must set up the ISPF interface. For instructions, see "Setting up the ISPF interface for IMS HALDB Toolkit" on page 74.
- If you want to use the DB Historical Data Analyzer utility in the TSO/ISPF environment, you must set up the ISPF interface of IMS HP Pointer Checker. For instructions, see "Setting up the ISPF interface for DB Historical Data Analyzer (IMS HP Pointer Checker)" on page 76.

The ISPF interface of Database Repair Facility can be used without customization.

Setting up the ISPF interface for IMS HALDB Toolkit

If you want to run IMS HALDB Toolkit as an ISPF application, you must define a VSAM KSDS data set to prepare the ISPF environment definition data set, and then configure your CLIST file to start the ISPF panel of IMS HALDB Toolkit.

Procedure

1. Locate the sample JCL, IHCCKSDS, in the SHICSAMP data set and modify the JCL statements.

   This sample JCL creates and initializes a VSAM KSDS data set for the ISPF environment definition data set, and creates an ISPF base environment module that defines the name of the ISPF environment definition data set.

   a) To create and initialize the VSAM KSDS data set, change *your_ksds_data_set* to the VSAM KSDS data set name for your environment **1**.

```
//*=================================================================
//*      Define Environment KSDS data set
//*=================================================================
//S1       EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DELETE your_ksds_data_set                                        1
 SET MAXCC=0
 DEFINE CLUSTER(NAME(your_ksds_data_set) -                        1
        INDEXED KEYS(16,0) RECSZ(2026,4080) -
        CISZ(4096) VOL(vvvvvv) CYLINDER(3 1) )
//*=================================================================
//*       Initialize Environment KSDS data set
//*=================================================================
//S2       EXEC PGM=IHCYUTIL
//STEPLIB  DD DISP=SHR,DSN=$dbsp.SIHCLOAD
//TRACE    DD DUMMY
//MSGPRINT DD SYSOUT=*
//KSDSDD   DD DISP=SHR,DSN=your_ksds_data_set                     1
//IHCSYSIN DD *
  RUN    PGM(IHCYKSDS)
//
```

b) To create an ISPF base environment module that defines the name of the ISPF environment definition data set, modify the JCL statements as follows:

- Change *ISPF_environment_definition_data_set* to your ISPF environment definition data set name, which is the name of the VSAM KSDS data set that you specified in Step a 2.
- Change *your_program_library* to the program library that you want the ISPF base environment module to be created in 3.

```
//C       EXEC  PGM=ASMA90,
//              PARM='LIST,RENT,DECK,NOOBJ'
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1   DD  SPACE=(CYL,(2,2)),UNIT=SYSDA
//SYSPUNCH DD  DSN=&&OBJ,SPACE=(CYL,(1,1)),
//              UNIT=SYSDA,DISP=(MOD,PASS),
//              DCB=(BLKSIZE=2400,LRECL=80,RECFM=FB)
//SYSIN    DD  *
*=================================================================
*        Environment definition dataset name
*=================================================================
         MACRO
         ENVIR &DSN=
IHCWENVR CSECT
IHCWENVR AMODE 31
IHCWENVR RMODE ANY
         DC    CL44'&DSN'
         MEND
*              +-------------------------------------------------------
*              | Environment KSDS dataset name
*              +-------------------------------------------------------
         ENVIR DSN=ISPF_environment_definition_data_set            2
         END
/*
//L       EXEC  PGM=HEWL,PARM='XREF,LIST',COND=(4,LE,C)
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DISP=SHR,DSN=$dbsp.SIHCLOAD
//SYSLMOD  DD  DISP=SHR,DSN=your_program_library                   3
//SYSUT1   DD  SPACE=(CYL,(5,2)),
//              UNIT=SYSDA
//SYSLIN   DD  DSN=&&OBJ,DISP=(OLD,DELETE)
//         DD  *
  NAME   IHCWENVR(R)
//
```

c) Submit the job.

2. Make sure that all users who work with IMS HALDB Toolkit have update authority for the ISPF environment definition data set.

3. Configure the CLIST file so that it starts the ISPF panel interface of IMS HALDB Toolkit.

The sample CLIST file is provided as IHCCHAL in SIHSSAMP.

Copy the sample CLIST file to your ISPF CLIST library and modify it.

```
PROC 0
  ISPEXEC CONTROL ERRORS RETURN
/*-------------------------------------------------------------------*/
/*        Allocate ISPF libraries                                    */
/*-------------------------------------------------------------------*/
  ISPEXEC LIBDEF ISPLLIB DATASET ID('your_ISPF_clist_library'), +
                                 ('$dbsp.SIHCLOAD')
  ISPEXEC LIBDEF ISPPLIB DATASET ID('$dbsp.SIHCPANL')
  ISPEXEC LIBDEF ISPMLIB DATASET ID('$dbsp.SIHCMESG')
  ISPEXEC LIBDEF ISPSLIB DATASET ID('$dbsp.SIHCSKEL')
/*-------------------------------------------------------------------*/
/*        Run DFSHALDB EXEC                                           */
/*-------------------------------------------------------------------*/
  ISPEXEC SELECT CMD(IHCWISPF) NEWAPPL(IHC) PASSLIB
  SET &RC = &LASTCC
  IF &RC /= 0 THEN DO
     WRITE 'RETURN CODE IS &RC '
     END
/*-------------------------------------------------------------------*/
/*        Deallocate ISPF libraries                                  */
/*-------------------------------------------------------------------*/
  ISPEXEC LIBDEF ISPSLIB
  ISPEXEC LIBDEF ISPMLIB
  ISPEXEC LIBDEF ISPPLIB
  ISPEXEC LIBDEF ISPLLIB
 END
//
```

4. Issue the following command to ensure that the IMS HALDB Toolkit ISPF main panel starts:

```
ex 'your_ISPF_CLIST_data_set_name(IHCCHAL)'
```

Setting up the ISPF interface for DB Historical Data Analyzer (IMS HP Pointer Checker)

The ISPF interface of the DB Historical Data Analyzer utility must be customized before it can be used. To use the ISPF interface, complete the following steps.

Procedure

Complete these steps to set up the DB Historical Data Analyzer ISPF interface:

1. Modify the logon procedure.

   a) Ensure that the Graphical Data Display Manager (GDDM) library can be accessed as a PDS library by the TSO terminal user.

      Unless the GDDM target library is specified as a link library, you must change the existing, or define a new, TSO logon procedure to contain a STEPLIB DD statement that refers to the GDDM program library (GDDMLOAD). For more information, see the GDDM products documentation.

   b) Concatenate the IMS HP Pointer Checker data sets for panels and messages with the corresponding ISPF/PDF data sets. Also, allocate the IMS HP Pointer Checker data set for programs in the ISPF link library (ddname ISPLLIB) in your TSO logon procedure.

      For example, specify as follows:

```
//ISPMLIB   DD DSN=HPS.SHPSMLIB,DISP=SHR
//             DSN=ISP.SISPMENU,DISP=SHR
//ISPPLIB   DD DSN=HPS.SHPSPLIB,DISP=SHR
//             DSN=ISP.SISPMENU,DISP=SHR
//ISPLLIB   DD DSN=HPS.SHPSLMD0,DISP=SHR
```

   c) If you use ISPPALT and ISPMALT for DBCS, allocate the IMS HP Pointer Checker data set to ISPPALT and ISPMALT.

   For more information about ISPF, see the *z/OS ISPF User's Guide, Volume 1*.

You can also allocate these data sets by coding appropriate TSO ALLOCATE commands. By using this method, you do not need to modify your TSO logon procedure. In this case, you must allocate the data sets before you invoke ISPF.

2. Modify the command list (CLIST).

   DB Historical Data Analyzer provides a sample command list (CLIST) to allocate data sets and to invoke DB Historical Data Analyzer. This sample CLIST is named FABGCMD0 and is provided in the SHPSCLIB data set. You might need to copy this sample CLIST to your command procedure data set and modify it to meet your installation requirements.

3. Modify the ISPF/PDF Primary Option Menu panel.

   You can modify the ISPF Primary Option Menu panel (ISR@PRIM) to add an entry so that you can invoke DB Historical Data Analyzer by a selection code.

   The following sample ISPF Primary Option Menu has been modified to invoke DB Historical Data Analyzer by selecting option D. This option starts processing by invoking a command procedure. The FABGCMD0 CLIST can be used to start the dialog processing.

```
%---------------------- ISPF/PDF PRIMARY OPTION MENU ----------------------
%OPTION ===>_ZCMD                                                          +
%                                                    +USERID   - &ZUSER
%   0 +ISPF PARMS  - Specify terminal and user parameters   +TIME     - &ZTIME
%   1 +BROWSE      - Display source data or output listings +TERMINAL - &ZTERM
%   2 +EDIT        - Create or change source data           +PF KEYS  - &ZKEYS
%   3 +UTILITIES   - Perform utility functions
%   4 +FOREGROUND  - Invoke language processors in foreground
%   5 +BATCH       - Submit job for language processing
%   6 +COMMAND     - Enter TSO command or CLIST
%   7 +DIALOG TEST - Perform dialog testing
%   8 +LM UTILITIES- Perform library administrator utility functions
%   9 +IBM PRODUCTS- Additional IBM program development products
%   C +CHANGES     - Display summary of changes for this release
%   D +DBHDA       - Invoke DB Historical Data Analyzer dialog
%   T +TUTORIAL    - Display information about ISPF/PDF
%   X +EXIT        - Terminate ISPF using log and list defaults
%
+Enter%END+command to terminate ISPF.
%
)INIT
  .HELP = ISR00003
  &ZPRIM = YES        /* ALWAYS A PRIMARY OPTION MENU     */
  &ZHTOP = ISR00003   /* TUTORIAL TABLE OF CONTENTS       */
  &ZHINDEX = ISR91000 /* TUTORIAL INDEX - 1ST PAGE        */
  VPUT (ZHTOP,ZHINDEX) PROFILE
)PROC
&ZQ = &Z
  IF (&ZCMD ¬= ' '
    &ZQ = TRUNC(&ZCMD,'.')
    IF (&ZQ = ' '
      .MSG = ISRU000
  &ZSEL = TRANS( &ZQ
                0,'PANEL(ISPOPTA)'
                1,'PGM(ISRBRO) PARM(ISRBRO01)'
                2,'PGM(ISREDIT) PARM(P,ISREDM01)'
                3,'PANEL(ISRUTIL)'
                4,'PANEL(ISRFPA)'
                5,'PGM(ISRJB1) PARM(ISRJPA) NOCHECK'
                6,'PGM(ISRPTC)'
                7,'PGM(ISRYXDR) NOCHECK'
                8,'PANEL(ISRLPRIM)'
                9,'PANEL(ISRDIIS)'
                C,'PGM(ISPTUTOR) PARM(ISR00005)'
                D,'CMD(FABGCMD0)'
                T,'PGM(ISPTUTOR) PARM(ISR00000)'
              ' ',' '
                X,'EXIT'
                *,'?' )
  &ZTRAIL = .TRAIL
)END
```

Chapter 6. Troubleshooting

Use these topics to diagnose and correct problems that you experience with IMS Database Utility Solution tools.

While you customize IMS Database Utility Solution, IMS Database Utility Solution components issue messages to help you track progress and to inform you when errors occur. For an explanation of the messages, refer to the corresponding product's user guide. See "Product documentation and updates" on page 9.

Topics:

- "How to look up message explanations" on page 79
- "Gathering diagnostic information" on page 79

How to look up message explanations

You can use several methods to search for messages and codes.

Searching for messages on the web

You can use any of the popular search engines that are available on the web to search for message explanations. When you type the specific message number or code into the search engine, you are presented with links to the message information in IBM Documentation.

Gathering diagnostic information

Before you report a problem with IMS Database Utility Solution tools to IBM Software Support, you need to gather the appropriate diagnostic information.

Procedure

Provide the following information for all IMS Database Utility Solution problems:

- A clear description of the problem and the steps that are required to re-create the problem.
- The version of IMS and the version of the operating system that you are using.
- A complete log of the job.
- The maintenance level of the product or the component.

  If the product in error supports the Diagnostics Aid program, run the Diagnostics Aid program to generate a Load Module/Macro APAR Status report. Provide the report when you contact IBM Software Support. For information about creating a Load Module/Macro APAR Status report, see the documentation of the individual products.

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.

Licensees 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

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® 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.html.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the Unites States, other countries, or both.

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

IBM®

Product Number: 5698-DUL