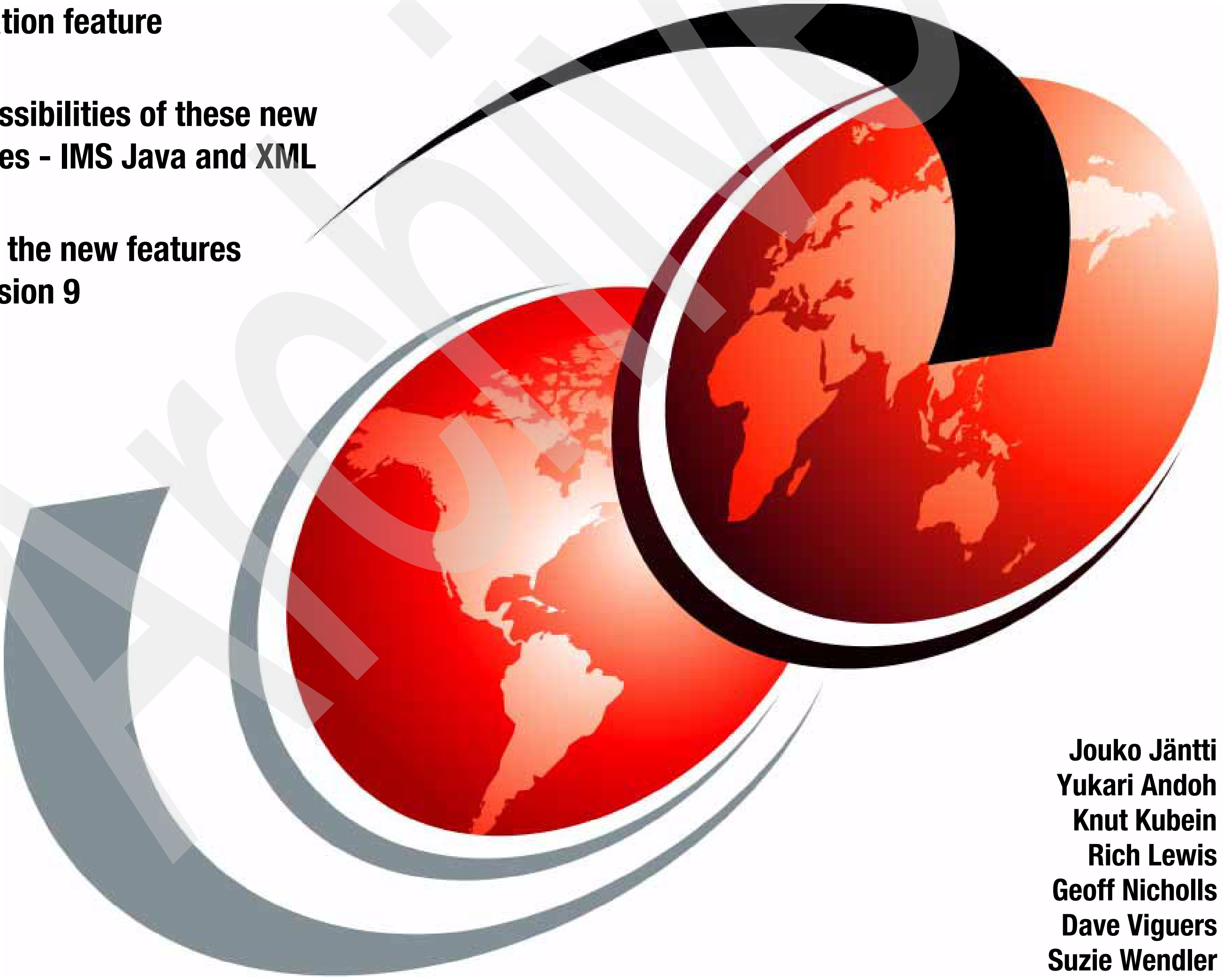
IMS Version 9 Implementation Guide:

A Technical Overview

Get familiar with the HALDB online reorganization feature

See the possibilities of these new technologies - IMS Java and XML

Explore all the new features of IMS Version 9



Jouko Jäntti
Yukari Andoh
Knut Kubein
Rich Lewis
Geoff Nicholls
Dave Viguers
Suzie Wendler

Redbooks

ibm.com/redbooks

International Technical Support Organization

IMS Version 9 Implementation Guide: A Technical Overview

December 2004

SG24-6398-00

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

First Edition (December 2004)

This edition applies to IMS Version 9 or later (program number 5655-J38) for use with the z/OS operating system.

Note: This book is based on a pre-GA version of a product and may not apply when the product becomes generally available. We recommend that you consult the product documentation or follow-on versions of this redbook for more current information.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

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.

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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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 illustrates 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. 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 IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

CICS/ESA®
CICS®
DataPropagator™
DB2 Universal Database™
DB2®
Enterprise Storage Server®
ESCON®
@server®
FICON®
IBM®
ibm.com®
IMS™
Language Environment®
MQSeries®
MVS™
Nways®
OS/390®
Parallel Sysplex®
ProductPac®
Redbooks™
Redbooks (logo) ™
RACF®
RMF™
SystemPac®
TotalStorage®
VTAM®
WebSphere®
z/OS®
zSeries®

The following terms are trademarks of other companies:

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Summary of changes

This section describes the technical changes made in this edition of the book and in previous editions. This edition may also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-6398-00
for IMS Version 9 Implementation Guide: A Technical Overview
as created or updated on October 27, 2005.

November 2004, First Edition

The revision of this First Edition, first published on December 14 2004, reflect the changes and additions described below.

November 2005, First Update

Changed information

Change bars identify the corrected area.

- Corrected the ICHERCDE macro in Example 6-6 on page 110 in order to facilitate migration matching the definitions used in RACF 1.6 and later.
- Removed the former Chapter 11, “MNPS replacement of XRF uservar’ and related references.

Preface

This IBM® Redbook provides an overview of the new functions and enhancements in IBM Information Management System (IMS™) Version 9. IMS Version 9 contains over 50 enhancements, in all areas of the product, that address availability, scalability, capacity, usability, manageability, operations, on demand business, and application development requirements from its huge customer base. All of the enhancements in this new version resulted from specific customer requirements. The major enhancements include:

- Improved database availability for HALDB databases is provided by the long-awaited online reorganization (OLR) enhancement.
- The functionality of the IMS Connect product is included as an integrated function of IMS.
- DBRC provides a new API that allows assembler programs to query the RECON.
- IMS Java™ users will have new JDBC SQL calls and DLIModel Utility support for retrieving existing IMS data in XML format as well as storing, indexing, searching, and retrieving valid XML documents into new or existing IMS databases.

We describe these enhancements and all other major changes in IMS Version 9. This redbook gives you the essential information that you need when you are migrating your IMS systems to IMS Version 9.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Jouko Jäntti is a Senior IT Specialist at IBM Global Services in Finland. Previously he was an IMS Specialist and a Project Leader with the IBM International Technical Support Organization, San Jose Center. He is the lead author of the 12 IMS related Redbooks, listed in the “Related publications” on page 313. Jouko has been working with IMS since 1986 starting with IMS Version 1.3 and has working experience in all IMS versions thereafter.

Yukari Andoh is an IT Specialist at the IMS group for IBM Japan Systems Engineering Co., Ltd., in Chiba, Japan. She provides technical support for IMS and IMS data management tools including three years of experience with the reorganization tools suite.

Knut Kubein is a Senior Advisory I/T Specialist with IBM Global Services in Germany. He has 30 years experience with IBM Large Systems Products working as a Technical Support Specialist, with 20 of those years devoted to IMS. He leads the IMS Support Team in EMEA. His areas of expertise include IMS data sharing, shared queues, and the Parallel Sysplex® architecture. He supports large bank accounts as well as other industrial IMS users. He has written extensively about shared queues.

Rich Lewis is a Senior Consulting IT Specialist with IBM Advanced Technical Support, for the Americas, located in Dallas, Texas. He has provided IMS technical support and consulting services as a member of the Dallas Systems Center for over 20 years. His areas of expertise include IMS database design, performance, and availability. He has developed and taught IMS classes for IBM and is a frequent speaker at IMS technical conferences, user group meetings, road shows, and seminars.

Geoff Nicholls is a Certified Senior IT Specialist, working in the IBM IMS Advocate team for the Silicon Valley Laboratory, and is based in Melbourne Australia. Geoff has 15 years of experience supporting IMS customers throughout the world. Previously, Geoff was an IMS Specialist with the International Technical Support Organization, San Jose Center. Geoff has a degree in Computer Science from the University of Melbourne, Australia. He has worked as an Application Programmer and Database Administrator for several insurance companies before specializing in database and transaction management systems with Unisys and IBM.

Dave Viguers is a member of the IBM IMS Development Team at Silicon Valley Lab working with performance, test, development, and customer support. Dave also develops and delivers education on IMS Parallel Sysplex. He has spent over 30 years working with IMS in various functions and organizations.

Suzie Wendler is a Consulting IT Specialist with IBM Advanced Technical Support, for the Americas, located in Dallas, Texas. Over the past several years her major responsibilities have included customer support, critical situation analyses, IMS e-business and connectivity projects, and design reviews, with an emphasis towards performance and DC communications. She has been involved in several on-site consulting projects including implementations of client/server environments, connectivity issues, and Web access. Additionally, she has made presentations for several user groups, including Guide and Share. She has also been involved in IBM-sponsored activities such as road shows and technical conferences, and has provided both technical and teaching support in IMS for IBM Education.

Thanks to the following people for their contributions to this project:

Paolo Bruni
Deanna Polm
International Technical Support Organization, San Jose Center

Rich Conway
Bob Haimowitz
International Technical Support Organization, Poughkeepsie Center

Rose Levin
Merle Beghtel
Jim Campbell
Kiran Challapalli
Kyle Charlet
Richard Cooper
Dario D'Angelo
Jeff Fontaine
Mike Gonzales
Judi Hill
Claudia Ho
Shyh-mei Ho
Chris Holtz
Tawei Hu
Jenny Hung
Terry Krein
Dorothy Lin
Tom Morrison
Betty Patterson
Jack Pinto
Frank Ricchio
Pat Schroeck
Alan Smith

Marc Spooner
Clyde Tahara
Judy Tse
Judi Vera
Jack Wiedlin
Jerry Zentner
IBM Silicon Valley Laboratory, San Jose, USA

Alonia Lonnie Coleman
IBM Advanced Technical Support, Americas, Dallas Center

Alan Cooper
IBM UK

Rober Hain
IBM Australia

Philippe Sueur
IBM France

Gen Sasaki
IBM Japan

Special thanks for help during the project and for her marvelous contribution to reviewing:

Alison Coughtrie
IBM Product Introduction Centre, Hursley Park, UK

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. QXXE Building 80-E2
  650 Harry Road
  San Jose, California 95120-6099

1

Introduction

IBM Information Management System (IMS) is unsurpassed in availability and speed for database and transaction processing. With the demands of the evolving on demand business environment and a marketplace that works in Web time, IMS continues to deliver the integrity, capability, and performance that customers have come to expect from IBM.

IBM is focused on strengthening IMS leadership, helping customers in on-demand business enablement, growth, availability, and systems management that newer environments and cost measures require. The enhancements in IMS Version 9 help you:

- Transform the way you do business with integrated information
- Build on demand applications that tolerate the rigors of doing business on the Internet
- Run a scalable, available, safe, and easily managed environment
- Leverage your information to make more informed decisions

This chapter provides the introduction and highlights of IMS Version 9.

1.1 Leading the industry

Industries worldwide rely on IMS to run their businesses. IMS is a part of everyday life. Chances are you use IMS when you turn on a light, make a telephone call, get a business loan, process accounting records, use your ATM card, put money in a bank, rent a car, purchase insurance, travel, send a package, track in-transit packages, trade stocks, control inventories, process payroll, update personnel records, control an assembly line, control a railroad, use a corporate database, run a government agency, conduct international business or banking, and many more tasks.

- More than 95% of the Fortune 1000 companies use IMS.
- IMS serves 200 million end users, managing over 15 million gigabytes of production data.
- IMS processes over 50 billion transactions every day.

IMS still owns the high-volume online transaction and database management environment. IMS customers have been driving their own growth with IMS.

IMS customers report:

- More than 100 million transactions were handled by one customer in a single day on a single sysplex system.
- 7 million transactions per hour and 120 million transactions per day were handled by another customer.
- One large customer has reached over 3000 days without an outage and is still going strong.
- Another large customer has transferred more than $2.5 trillion through IMS in a single day.

IMS, the IBM premier hierarchical transaction and database management system, is the product of choice for critical online operational applications and data where support for high availability, performance, capacity, integrity, and low cost are key factors. Today, IMS manages the world's mission-critical data and has been at the forefront of the swing back to mainframe usage.

1.1.1 Unsurpassed performance

The IBM performance group achieves extremely high transaction rates with IMS Version 9 running in a high stress sysplex environment using four IMSs on a single IBM eserver® zSeries® 990 model B16 processor.

IMS demonstrated 21,396 transactions per second with all database updates using the IMS Fast Paths shared expedited message handler queue capability, 4-way data sharing, shared virtual storage option areas, and new IMS Version 9 capabilities.

Using the new processor and the IBM TotalStorage™ Enterprise Storage Server™ model 800, IMS was able to reach 28.5 MB per second logging bandwidth. This shows that IMS scales up to large logging requirements for high stress IMS activity, such as integrated online reorganization for HALDBs, multiple systems coupling, and shared queues.

These performance tests demonstrate that nothing can match the performance of IMS and the zSeries in transaction and database access, making IMS the industry leader for on demand business and all of your business needs.

1.2 IMS Version 9 highlights

IMS Version 9 is evolving with on demand business by providing product integration openness with tools for application development and connectivity. IMS Version 9 offers manageability with autonomic computing to ease use, eliminate or reduce outages, and minimize the education curve for users. IMS scalability enhancements provide flexibility for growth and expansion.

With IMS Version 9, new hardware and software facilities, combined with many data and application sources, optimize IMS performance, capacity, availability, and recovery. IMS DB Version 9 offers enhancements in:

- Integration and openness

  Broadened access and storage of XML data in IMS databases, with XML and Java™ technology.

- Manageability

  Expanded, autonomic, user friendly commands and interfaces that are accessible across environments. Eased installation and system generation, as well as enhanced security and serviceability.

- Scalability

  Improved availability and recovery, with fully-integrated online reorganization for High Availability Large Databases (HALDB), which provide concurrent online update and availability of data.

  Improved performance, capacity and virtual storage constraint relief (VSCR) for Database Recovery Control (DBRC) and Fast Path.

IMS TM Version 9 offers enhancements in:

- Integration and Openness

  Broadened application development and execution tools, with XML and Java technology and with integrated IMS Connect function.

- Manageability

  Extended autonomic network switchover capability, and eased installation and system generation, as well as enhanced security and serviceability.

- Scalability

  Improved system availability, performance, and capacity for workload balancing.

IBM also provides a robust portfolio of tools and utilities to help you manage IMS efficiently and gain the best possible performance. To learn more about the IBM IMS products and tools, visit the Web site at:

http://www.ibm.com/ims

1.3 IMS Database Manager enhancements

In the areas of integration and open access with new application development and connectivity, IMS Version 9 includes the following enhancements:

- XML storage in IMS databases

  XML enablement provides support for storage and retrieval of XML documents into and from IMS databases. The XML enablement extends the IMS Java JDBC interface to allow the composition of XML documents from pre-existing non-XML IMS data. Additionally, IMS can store XML documents into IMS either wholly intact or decompose them into standard IMS segments and fields that can be used by existing or new non-XML enabled applications. XML enhancements are described later in this book in Chapter 9, “XML storage in IMS databases” on page 145.

- Tooling for XML storage

  The DLIModel Utility enhancement generates XML schemas from database definitions (DBDs) and program specification blocks (PSBs). The DLIModel Utility supports XML storage and retrieval at run time. This enhancement automates schema generation to accelerate IMS application development.

- Distributed database access

  IMS Java remote database services enable IMS DB access from an Enterprise Java Bean (EJB) that runs on a distributed J2EE application server, without requiring additional z/OS® application programming.

- Additional enhancements including the following:

  - Symbolic checkpoint and restart support for the Java batch region
  - SQL enhancements for new SQL keywords and aggregate functions
  - JDBC™ 2.0 support, including the ability to obtain scroll insensitive result sets
  - IMS-DB2 interoperability from within a Java dependent region
  - Provides a Java interface to access GSAM
  - The DLIModel Utility graphical user interface (GUI) provides a graphical version of the DLIModel Utility built as an Eclipse plug-in

We describe these features more in detail in Chapter 9, “XML storage in IMS databases” on page 145 and in Chapter 10, “IMS Java enhancements” on page 159 in this book.

IMS Version 9 Database Manager enhancements in the areas of scalability in high performance, capacity, availability, and recovery include the following:

- HALDB online reorganization enablement

  integrated HALDB online reorganization (OLR) is designed to address customer requirements for enhanced IMS data availability. OLR provides online reorganization by partition with concurrent online update and availability during reorganization. OLR is designed to be totally nondisruptive, there is no outage. You can adjust the pace of OLR to further minimize online impact. Multiple partitions can be reorganized in parallel. Coordination is provided through IMS Database Recovery Control (DBRC).

  We describe the HALDB OLR feature in more detail in Chapter 3, “Integrated HALDB online reorganization” on page 29 in this book.

- Database Recovery Control (DBRC) enhancements

  The new DBRC application programming interface (API) enables customer-written application programs to obtain services from DBRC. The application obtains these services by issuing DBRC API requests; DBRC returns results to an area in storage where the application can retrieve them.

  Command authorization support, initially provided in IMS Version 8 for DBRC batch commands, is now also provided for the online DBRC /RMxxxx commands. DBRC modules have been moved above the 16 MB line for enhanced capacity.

  DBRC can reassign and reuse currently unused database management block (DMB) numbers to reduce the potential for outages.

  We describe the DBRC enhancements in IMS Version 9 in more detail Chapter 5, “DBRC enhancements” on page 73 in this book.

- Fast Path enhancements

  Enhancements provide improved performance of Fast Path data entry database (DEDB) area open/close processing. Increased parallelism is obtained by exploiting multiple task control blocks (TCBs) to process multiple area open/close requests simultaneously. This can provide a performance advantage when many areas are open. Additional usability enhancements improve the handling of DEDB area open/close processing during IMS emergency restarts and other system error recovery scenarios.

  Fast Path shared virtual storage option (SVSO) multi-area structure support provides for housing multiple DEDB areas in a single Coupling Facility structure. Instead of having one Coupling Facility structure per area, multiple areas can reside in a single Coupling Facility structure. This reduces the total number of Coupling Facility structures that must be defined in a system.

  A number of additional enhancements increase the serviceability and usability of Fast Path. These enhancements include, among other items, additional log record information for sequential dependent (SDEP) segments that can be exploited by the IMS Performance Analyzer.

  We describe the DBRC enhancements in IMS Version 9 in more detail Chapter 4, “Fast Path enhancements in IMS Version 9” on page 57 in this book.

IMS Version 9 Database Manager includes also the following database utilities and other scalability enhancements:

- Support for tape block size greater than 32 KB provides a performance advantage in the image copy and recovery utilities.
- Enhanced recoverability of external subsystem indoubt units of work enables you to resolve indoubt units of work before IMS restart.
- IMS dynamic allocation provides virtual storage constraint relief (VSCR) by ensuring that generated data set association blocks (DSABs) are allocated above the 16 MB line. For example, a customer with 20,000 full function or Fast Path data sets that are dynamically allocated could have more than a megabyte of storage allocated above the 16 MB line.
- Logger enhancements improve the availability of online log data sets (OLDSs) for restart and the integrity of log data that is obtained for write ahead data sets (WADSs).

1.4 IMS Transaction Manager enhancements

IMS Transaction Manager offers many enhancements and improvements in the areas of integration and open access with new application development and connectivity. As of IMS Version 9, the functionality of the IMS Connect product (program number 5655-K52), Version 2.2, is included in IMS as an integrated feature. Refer to Chapter 8, “Integrated IMS Connect support” on page 135 for details.

Open transaction manager access (OTMA) security and serviceability enhancements provide the following performance advantages:

- OTMA clients can now set the access control environment element (ACEE) security aging value for user IDs in the OTMA message prefix without requiring the client to reconnect to OTMA.
- The OTMA trace table entries for the OTMA user exit routines (DFSYIOE0, DFSYPRX0, and DFSYDRU0) are standardized so that you can view the return code set by the exit routines. The DFSYPRX0 user exit routine for OTMA can set the OTMA destination TPIPE name.

- The Z2 field of IMS application data can now be set or changed for OTMA translation input and output.

OTMA and many other IMS Transaction Manager enhancements are discussed later in this book in Chapter 12, “Miscellaneous enhancements in IMS Version 9” on page 199.

Other IMS Version 9 Transaction Manager enhancements in this area include the following:

- IMS Version 9 has enhancements to support RACF® (or an equivalent product) replacement of the security maintenance utility (SMU) security. IMS now consolidates implementation of all security under RACF. SMU security is available in IMS Version 9 to enable migration to RACF. This and other security enhancements are discussed later in this book in Chapter 6, “Security considerations with IMS Version 9” on page 85.
- The allowable number of named classes of transactions is increased from 255 to 999 to enhance usability and customization of the system.
- An added option allows a type 3 logical unit device to log on as an Extended Terminal Option (ETO) type 1 secondary logical unit (SLU1) or 3270P device, making IMS available for these devices.
- COBOL XML support through WebSphere® Studio Enterprise Edition Version 5-generated XML converters allows IMS COBOL applications to parse and transform XML documents. You can now send and receive XML documents directly to and from IMS COBOL applications that run inside IMS regions.
- Message Format Service (MFS) Web Services support through WebSphere® products enables you to publish existing MFS transactions on the Internet as Web Services and connect to IMS using simple object access protocol (SOAP) and Enterprise Java Bean (EJB) bindings.

In the areas of scalability in high performance, capacity, availability, and recovery IMS Version 9 offers the following:

- Fast Path enhancements

  The Fast Path optional expedited message handler queue (EMHQ) structure eases manageability, enabling you to bypass allocating an EMHQ structure and its associated data sets when the shared EMH is not being used. This and other Fast Path enhancements are discussed further in Chapter 4, “Fast Path enhancements in IMS Version 9” on page 57.
- Database Recovery Control (DBRC) enhancements

  The new DBRC application programming interface (API) enables customer-written application programs to obtain services from DBRC. The application obtains these services by issuing DBRC API requests; DBRC returns results to an area in storage where the application can retrieve them.

  Command authorization support, initially provided in IMS Version 8 for DBRC batch commands, is now also provided for the online DBRC /RMxxxx commands.

  DBRC modules have been moved above the 16 MB line for enhanced capacity. DBRC enhancements are described in more detail in Chapter 5, “DBRC enhancements” on page 73.
- Utilities and other scalability enhancements:

  Enhanced recoverability of external subsystem indoubt units of work enables you to resolve indoubt units of work before IMS restart.

  Logger enhancements improve the availability of online log data sets (OLDSs) for restart and the integrity of log data that is obtained for write ahead data sets (WADSs).

1.5 IMS system manageability enhancements

With IMS Version 9, IMS systems manageability continues to evolve. New enhancements for IMS TM and IMS DB enable you to manage operations more effectively, while reducing system generation time and effort and various miscellaneous enhancements to fulfill customer requirements. These enhancements include the following:

- Operations management enhancements, which include:
  - Command environment enhancements simplify the Common Service Layer (CSL), which enables you to use IMS enhanced format commands and the IMS single point of control (SPOC) without requiring the Resource Manager.
  - Sysplex-wide database commands expand operations management single point of control to handle database commands. In IMS Version 8, an Operations Manager (OM) application programming interface (API) was provided to enable you to issue IMS commands from the OM. In IMS Version 9, commands for database and area resources are added to provide you with the ability to better manage the IMS sysplex and to provide a single-system image.

  These enhancements are described in more detail in Chapter 7, “Type-2 commands and enhanced command environment” on page 115.

  > **Note:** IMS Version 9 publications use new terminology to distinguish the types of IMS commands. Classic commands are called type-1 commands. IMSplex or enhanced commands are now called type-2 commands. Type-1 commands are generally preceded by a leading slash character, and they may be entered from any valid IMS command source. Type-2 commands may be entered only through the Operations Manager (OM) API.

- System generation time and effort enhancements, which include:
  - System generation enhancements stage the removal of conditional binds currently done by system generation. This removes the restriction of requiring separate execution libraries for IMS environments. These enhancements eliminate the conditional link of composite modules, thus eliminating a step and reducing the impact of the system generation process.
  - Online change modules enhancements remove most online change modules from the nucleus link-edit step and place them in their own load module. The value to the customer is saving space below the 16 MB line private storage. This is another step toward limiting the impact of IMS system generation, easing the process for defining IMS resources.
  - Extended Terminal Option (ETO) feature checking has been moved to the initialization phase, eliminating the need for an IMS system generation to add this feature.
  - DBRC type 4 SVC module enhancements enable you to apply maintenance to the module without having to restart IBM z/OS.
  - Dynamic add of the resource cleanup module enhances availability and serviceability.
  - Enhancements to the Syntax Checker, which provides detailed assistance with the tailoring of IMS, include support for the definition and maintenance of additional IMS PROCLIB members, and the addition of usability features.
  - Installation verification program (IVP) enhancements provide new sample applications that support the Common Queue Server (CQS) and the enhanced command environment. In addition, a number of usability improvements are provided, including the ability to import IVP variables from previous versions of IMS.

These enhancements are described in more detail in Chapter 2, “Product packaging, installation and resource definition enhancements” on page 9.

- Miscellaneous enhancements

  Miscellaneous enhancements in the system manageability area include the following improvements to fulfill customer requirements:

  - High Availability Large Database (HALDB) usability eases partition initialization without requiring the use of DBRC commands.
  - The new **/DIAGNOSE** command eases IMS serviceability.
  - Command recognition character registration eases operations management by providing unique subsystem registration so that an operator can enter a command from any system in a sysplex and have it routed to the correct subsystem. With unique subsystem registration, the operating system can detect collisions between subsystems, and can inform operators or system programmers which prefixes are currently in use.
  - A new write-to-operator (WTO) message replaces the WTO-reply (WTOR) message for Fast Database Recovery (FDBR). The new message notifies the operator when FDBR is started before the active IMS, without having to wait for a reply. FDBR waits for the active IMS to start up. This eases operations usability in this environment.
  - Enhancements to the online change copy utility support the IEBCOPY utility data set parameters WORK, SIZE, and LIST. The online change copy utility passes the values for these parameters to the IEBCOPY utility. These enhancements reduce outages because they enable you to override default values.

  These enhancements are described in more detail in Chapter 12, “Miscellaneous enhancements in IMS Version 9” on page 199.

2

Product packaging, installation and resource definition enhancements

This chapter describes the new features and changes from the previous version in the areas that you will be dealing with first, after you have ordered and received IMS Version 9 software and begin to install it.

This chapter covers the following topics:

- Product packaging
- Resource definition and installation changes
- IMS Application Menu
- Installation verification program (IVP) enhancements
- Syntax Checker enhancements

When you start planning the migration of your existing environment, please also refer to Chapter 13, “Migration and coexistence considerations” on page 223, which provides information about the coexistence between different IMS versions and some migration considerations.

2.1 Product packaging

The IMS Version 9 product number is 5655-J38. IMS products are ordered by Function Modification Identifiers (FMIDs). Table 2-1 lists the FMIDs for IMS Version 9.

Table 2-1 FMIDs for IMS Version 9.

Function modification identifier	Function description
HMK9900	System Services
JMK9901	Database Manager
JMK9902	Transaction Manager
JMK9903	Extended Terminal Option (ETO)
JMK9904	Recovery Level Tracking (RSR RLT)
JMK9905	Database Level Tracking (RSR DLT)
JMK9906	IMS Java
HIR2101	IRLM 2.1
HIR2102	IRLM 2.2

Note that both IRLM releases 2.1 and 2.2. can be used with IMS Version 9, and it is a customer's decision which one to use. Both releases can coexist with each other and you can have members in a data sharing environment on either release. The release/s of IRLM distributed with IMS Version 9 depend on the order type as below:

- CBPDO, ProductPac® or Standalone ==> both IRLM 2.1 and IRLM 2.2 are included
- ServerPac, FunctionPac or SystemPac® ==> IRLM 2.2 is included
- CustomPac ==> CustomPac is made up of three offerings: FunctionPac, ProductPac, and SystemPac. Which IRLM version you get with CustomPac depends on which offering you order.

Integrated IMS Connect function

The IMS Connector for Java 9.1.0.1 runtime component for z/OS is delivered with IMS. The IMS Connector for Java 9.1.0.1 development component which ships with WebSphere Studio Application Developer Integration Edition (WSADIE) and is delivered as an update to WSADIE 5.1.1.

IMS Java remote database services

To use IMS Java remote database services (RDS) to access IMS databases from applications that run on WebSphere Application Server on a non-z/OS, you must download IMS Java files from the IMS Java Web site. These files are required in addition to the files that are installed as part of the SMP/E installation of the IMS Java FMID. To download the required IMS Java files, go to the IMS Web site and link to the IMS Java Web page for more information:

`http://www.ibm.com/ims`

2.2 Resource definition and installation changes

It has been a customer requirement for a long time to reduce IMS system generation time and effort. IMS has responded to that requirement by providing enhancements over the several versions. The major steps in this area have been the following:

- IMS Version 4 stopped using system generation to support major new functions by introducing ETO.
- IMS Version 5 and 6 removed conditional assembly modules.
- IMS Version 7 put non-conditional link-edit modules under SMP/E control.
- IMS Version 8 provided the Resource Manager, coordinated global online change, Syntax Checker, packaging, installation and IVP enhancements, and removed the RSR RLT/DLT features install checking.

There are substantial changes to IMS system definition in IMS Version 9 to further simplify this process. In the short term, we are continuing to reduce the IMS system generation effort by:

- Removing the restriction of maintaining SDFSRESL libraries for different system definitions.
- Removing online change modules from the nucleus link-edit step.
- Removing the ETO feature install checking.
- Allowing the dynamic install of DBRC type 4 SVC.
- Replacing DFSMRCL0 by new resource cleanup services.

2.2.1 Shared SDFSRESL for different configurations and system definitions

With IMS Version 9, you do not need separate IMS SDFSRESL libraries for system generation of any of the following environments: DB/DC, DBCTL, DCCTL or Fast Path. You can use separate IMS SDFSRESLs or combine them, as needed for your environment. This is accomplished through two capabilities: the removal of system generation conditional binds and support for a generic DFSVC000 module through the ability to override required values through new IMS Control Region parameters. Three new control region parameters, SVC2 (type 2 SVC number), MCS (IMS route code), DESC (IMS descriptor code), replace the equivalent IMSCTF and IMSCTRL system definition parameters. The IMSGROUP (defines the IMSID) control region parameter must be added to each IMS/DBC/DCC procedure.

A new batch region parameter RGSUF for DFSPBxxx member provides batch specific overrides for the parameters.

Removal of system generation conditional binds

In previous releases, the mechanism to bind (link edit) required modules to support a specific IMS environment was done through conditional bind controls in the system generation process. With this enhancement, all non-suffixed modules for all possible configurations are always bound during SMP/E install. During IMS initialization, the appropriate modules for the specific configuration are loaded into SDFSRESL as standalone modules. The nucleus which is a suffixed module is still conditionally bound.

This capability removes the need to maintain separate resource libraries for different configurations.

Generic DFSVC000 module

DFSVC000 is a module that IMS uses to maintain information unique to a specific execution environment such as the IMSID, subsystem type, SVC numbers, and so on. In releases prior to IMS Version 9, when users wanted to share SDFSRESL, a common practice was to place the DFSVC000 module unique to an execution environment into a library that was concatenated in front of the shared SDFSRESL.

With IMS Version 9, an enhancement has been introduced that allows a generic DFSVC000 to be used for all execution environments. Module DFSVC000 continues to be built during system generation into the SDFSRESL, and it is based on most recent system generation that used the library. If a system requires a type 2 SVC number, route code or descriptor code that is different than what is in the DFSVC000 member, the following new parameters are available to provide the values:

- SVC2

  SVC2 specifies the type 2 SVC number reserved for use by the IMS subsystem. The value provided can range from 200 to 255. If not specified, then the value defined in the IMSCTF macro during system generation is used or the default of 254.

- MCS

  MCS specifies the z/OS routing code or codes to be assigned to the IMS system console and is applicable when multiple console support is included in the operating system. If not defined, the value from the IMSCTRL macro is used or the default of (2,7).

- DESC

  DESC specifies the descriptor code to be assigned to the IMS system console messages when multiple console support is included. It defaults to the IMSCTRL macro specification or the default of 7.

Considerations for sharing DFSVC000

The generic DFSVC000 member can be created to contain whatever default values you choose as the most appropriate. These can be overridden by execution parameters.

- SVCNO (IMSCTF)

  SVCs are downward compatible, so the recommendation is to use the highest level for all releases.

- MCS and DESC (IMSCTRL)

  Recommendation is to use the most common set of parameters.

- IMSID (IMSCTRL)

  Specify the value you want for the default.

The existing techniques to provide override values for IMSID continue to be used:

- IMSID which is always unique to an IMS environment continues to be overridden with the IMSID= specification on the control region.

  For associated dependent regions, in other words, MPPs, BMPs, IFPs, JMPs, JBPs (and even any DLISAS/DBRC regions that are manually started), the IMSID value can match that of the target control region's IMSID or IMSGROUP value.

- The IMSGROUP parameter is a technique that allows a dependent region to use MVS™ name token services to find the correct IMSID to use.

  This facilitates creating generic dependent region JCL that can be moved from one system to another.

RGSUF parameter for batch

The RGSUF parameter is now available in IMS Version 9 systems for batch regions. This allows batch to use a DFSPBxxx member of PROCLIB for overriding the execution parameter values that are applicable to batch (DBRC=, IRLM=, IMSID=, SVC2=, MCS=, DESC=). RGSUF has no default value and must be specified in the batch region if access to DFSPBxxx is desired.

Note that as in the past, an override value for DBRC is honored only if the installation default is NO or YES. If the installation default is defined as FORCE then the value in this override parameter is ignored. The IMSID= value in a batch environment is used as an identifier to relate messages that are routed to the z/OS system console with the batch job. This is the value that is displayed along with the message on the console. To avoid confusion as to which region issued the console message, each batch region can specify a unique value.

Figure 2-1 on page 14 shows how the IMS1 control region JCL overrides the default values that were generated in DFSVC000.

Since the control region JCL includes a specification of IMSGROUP=IMSA, MVS name token services are invoked to create a token called IMSA that is associated with the control region IMSID value of IMS1. Dependent regions that need to connect to IMS1 can specify one of the following in the execution JCL:

- IMSID

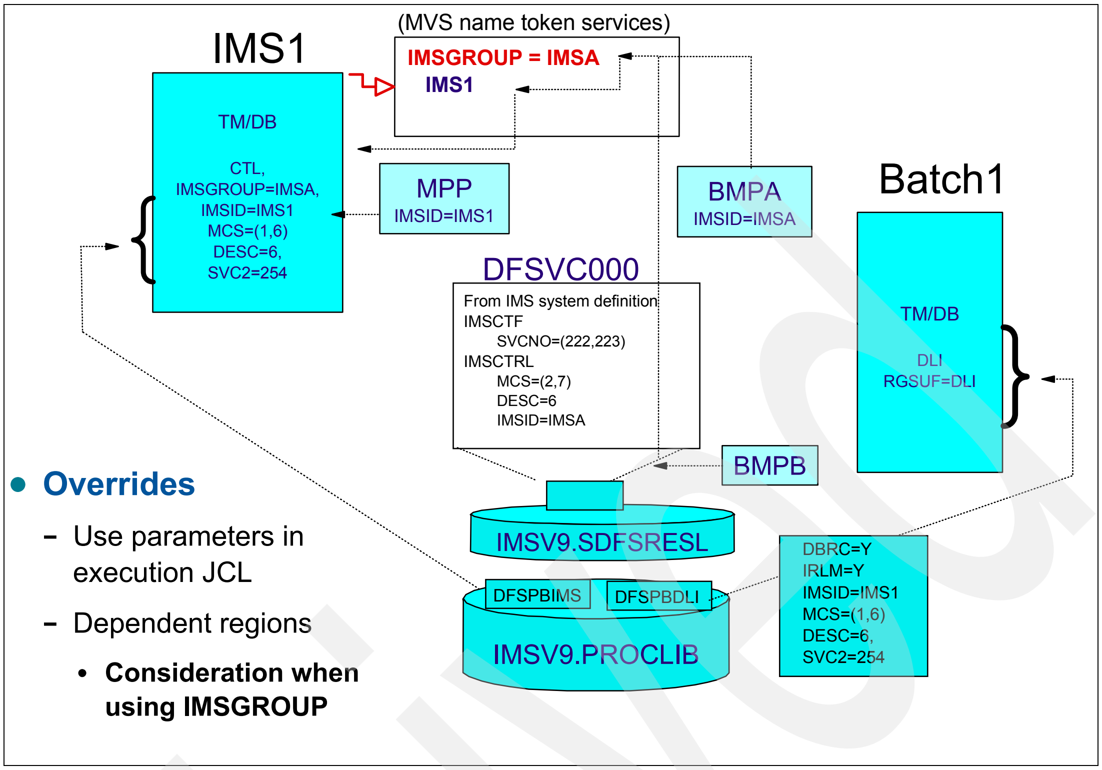
  The target IMSID value, for example an MPP which directly specifies IMSID=IMS1.

- The IMSGROUP name value

  BMPA which specifies IMSID=IMSA and results in MVS name token services equating IMSA with IMS1.

- No IMSID specification

  BMPB which results in using the default IMSID value in DFSVC000. Since the default IMSID value is not equal to an actual control region IMSID, MVS name token services in this case also provide the translation from the name IMSA to the actual IMSID value of IMS1.



- **Overrides**
  - Use parameters in execution JCL
  - Dependent regions
    - **Consideration when using IMSGROUP**

Figure 2-1 DFSVC000 override

2.2.2 DBRC keyword ignored in the system definition

Starting in IMS Version 9 the default for DBRC in batch is no longer set in the system definition. The default is DBRC=YES in batch and utility regions. The IMSCTRL macro specification for DBRC= in the system definition is ignored, and if coded, the following informational message will be printed:

```
G011 WARNING: THE DBRC | MAXIO KEYWORD IS NO LONGER USED AND IS IGNORED.
```

A new module, DFSIDEF0, is used by batch and utilities. DFSIDEF0 contains the batch and utilities installation default for DBRC. A sample of DFSIDEF0 is provided by IMS. If DFSIDEF0 cannot be loaded at initialization time, DBRC defaults to YES as the initial value. It can be overridden by JCL or in PROCLIB member DFSPBxxx.

If it is not desired to have DBRC used in batch and utility regions, this can be changed in one of three ways:

- Assemble and bind DFSIDEF0 into IMS execution library.
- Override the value in IMS.PROCLIB DFSPBxxx.
- Override the value in JCL.

Batch and utility regions use the DFSIDEF0 module during initialization. A sample DFSIDEF0 module is provided with IMS in IMS.ADFSSMPL library. The provided sample module contains a batch and utility region default of DBRC=YES. This value is coded on the DFSIDEF macro. If DBRC is to be used for batch and utilities, creating the DFSIDEF0 module is not required.

A sample job for creating the DFSIDEF0 module is also provided as a new IVP job, IV_E207J. The JOB can be found in IMS.INSTALIB library and it can be run as part of IVP.

Example 2-1 shows the contents of the JOB provided by IVP. Without modifications the module has the default value of "DBRC=YES" coded. If you want to change the DBRC default to NO or FORCE, make the changes by commenting out and uncommenting the appropriate lines, and then assemble and bind the module into the SDFSRESL.

Example 2-1   Job for assembling and binding the module DFSIDEF0

```
//ASSEMBLE EXEC PGM=ASMA90,PARM='NOOBJ,DECK'
//SYSLIB   DD DSN=IMS910G.SDFSMAC,DISP=SHR
//SYSPUNCH DD DISP=OLD,DSN=IMS910G.OBJDSET(DFSIDEF0)
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(15,15))
//SYSIN    DD *
IDEF0    TITLE 'DFSIDEF0 - IMS INSTALLATION DEFAULTS BLOCK'
DFSIDEF0 CSECT
         SPACE 1
         DFSIDEF TYPE=BEGIN
*-----------------------------------------------------------------*
*  DBRC=  DBRC FOR BATCH                                          *
*    DBRC=YES -  USE DBRC.                                        *
*    DBRC=NO  -  DO NOT USE DBRC                                  *
*    DBRC=FORCE - USE DBRC. CAN NOT BE OVERRIDDEN WITH EXEC PARM  *
*-----------------------------------------------------------------*
         DFSIDEF TYPE=PARM,DBRC=YES
*****    DFSIDEF TYPE=PARM,DBRC=NO
*****    DFSIDEF TYPE=PARM,DBRC=FORCE
         DFSIDEF TYPE=END
         END   DFSIDEF0
/*
//*
//BIND     EXEC PGM=IEWL,
//            PARM='SIZE=(880K,64K),NCAL,LET,REUS,XREF,LIST',
//            COND=(0,LT,ASSEMBLE)
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD DSN=IMS910G.OBJDSET,DISP=SHR
//SYSLMOD  DD DSN=IMS910G.SDFSRESL,DISP=SHR
//SYSUT1   DD UNIT=(3390,SEP=(SYSLMOD,SYSLIN)),
//            SPACE=(CYL,(10,1))
//SYSLIN   DD *
   INCLUDE SYSPUNCH(DFSIDEF0)
   NAME  DFSIDEF0(R)
/*
//*
```

2.2.3 Removing conditional link-edit for online change modules

Approximately 33 online change modules are removed from the nucleus link-edit step of IMS system definition and they become standalone modules. Approximately 20 of these modules are moved to above-the-line private storage. This is another step towards simplifying system definition and allows you to save below-the-line private storage.

2.2.4 ETO feature checking

The Extended Terminal Option (ETO) feature installs completely with the SMP/E install process and no longer requires system generation to validate the install. The validation will occur during IMS initialization.

ETO installation in IMS Version 9 continues to be done through the SMP/E install process. There is no longer a system generation requirement to validate the installation. The system generation process, however, will still continue to build ETO descriptors, if requested. In the stage 1 system definition, the IMSCTRL macro has been changed to ignore the first two subparameters of the ETOFEAT parameter. These were used, in previous releases, to determine whether or not validation of the ETO feature was to take place during system generation. If specified in IMS Version 9, an informational system generation G1209 message will be produced to inform the user that the subparameters are no longer valid and are ignored. The third subparameter is still valid and used to request ETO descriptor generation, with the following values:

- NO is the default and produces a normal system definition without descriptors.
- ALL requests that descriptors be generated as part of normal system definition.
- ONLY requests that this invocation of system definition produce only descriptors.

NO is the new default value in IMS Version 9.

During IMS startup if the ETO=Y parameter is specified, then the IMS initialization process will attempt validation of the install. If the feature is not installed, IMS will issue a user abend U3476 and will print message DFS3676I. The response to this abend is to turn off the ETO feature request with ETO=N or go back and review the SMP/E install of the feature.

2.2.5 Dynamic update of DBRC type 4 SVC

The dynamic SVC update (DFSUSVC0) utility has been enhanced to allow dynamic update of the DBRC type 4 SVC module. Thus, you can apply maintenance to the DBRC type 4 SVC module without having to re-IPL z/OS after each update. The SVC module, however, must still be installed into the z/OS system and placed in LPALIB or an MLPA library.

In previous releases, the utility only worked for type 2 SVCs. When it ran, there could not be any active IMS images (control regions or batch regions) using the SVC number. The code searched the IMS SVC directory table to determine if there was an active IMS using the SVC number and failed if one was found. Any updates to the DBRC type 4 SVC required re-IPLing z/OS.

As in previous releases, IMS system definition still creates the SVC routines using the IMSCTF macro-defined user-specified numbers, or the IMS-provided default numbers. The name created for the type 4 SVC is IGC00xxx where, xxx indicates the signed decimal type 4 SVC number. IMS system definition copies the load modules representing the SVC routines into the SDFSRESL library.

IMS dynamic SVC update utility (DFSUSVC0) continues to support dynamic updates of the type 2 SVC and adds support for type 4 SVC. As in previous releases, for updates to the type 2 SVC, the utility checks to see if there is an active IMS using the SVC number. If so, the update fails. The same does not apply to the DBRC type 4 SVC. DBRC regions are not required to be down when the type 4 SVC is updated. The utility determines the SVCs to be updated and dynamically changes the z/OS SVC table to point to the new SVC modules.

The SVCs that are to be updated are specified by a new parameter in DFSUSVC0 JCL. The EXEC statement must be in one of the following forms:

```
//STEP001 EXEC PGM=DFSUSVC0
//STEP001 EXEC PGM=DFSUSVC0,PARM='SVCTYPE=(2)'
//STEP001 EXEC PGM=DFSUSVC0,PARM='SVCTYPE=(4)'
//STEP001 EXEC PGM=DFSUSVC0,PARM='SVCTYPE=(2,4)'
```

The EXEC statement allows you to specify whether the IMS type 2 SVC module, the DBRC type 4 SVC module, or both are to be updated. When SVCTYPE=(2) is specified, the IMS type 2 SVC module is updated. When SVCTYPE=(4) is specified, the DBRC type 4 SVC is updated. When SVCTYPE=(2,4) is specified, both the IMS type 2 SVC module and the DBRC type 4 SVC module are updated. If a value is not specified for the SVCTYPE= parameter, the IMS type 2 SVC module is updated by default.

For more information about DFSUSVC0 utility, refer to the manual *IMS Version 9: Utilities Reference: System,* SC18-7834.

2.2.6 Replacing DFSMRCL0 by new resource cleanup services

IMS dynamically installs its resource manager cleanup routine; you do not need to install the DFSMRCL0 module as part of the IMS installation. Registration of the IMS resource manager cleanup routine with the operating system is done automatically during IMS startup and it is established as a dynamic rather than static resource manager.

With the invocation of the z/OS RESMGR service to dynamically register the new IMS resource cleanup routine during IMS initialization, DFSMRCL0 no longer needs to be bound into LPALIB nor does it need to be defined as a static z/OS resource manager in CSECT IEAVTRML of module IGC0001C.

The RESMGR service actually calls a stub module DFSMRC00 which calls the new IMS resource cleanup module DFSMRC20. This allows for transparent application maintenance. DFSMRC20 also contains a version number. If a dynamic update is done, the old copy will remain until the next IPL.

As long you are using earlier versions of IMS than IMS Version 9 in your environment, those prior releases still require DFSMRCL0 to be installed as part of z/OS. IMS Version 9 resource cleanup can coexist with DFSMRCL0 from previous releases. If a pre-IMS Version 9 and IMS Version 9 run on the same LPAR, use the highest version pre-IMS Version 9 DFSMRCL0 or DFSMRCL0 that is shipped with IMS Version 9 for binding it into LPALIB. The name "DFSMRCL0" still must be zapped into IEAVTRML CSECT of the z/OS module IGC0001.

When migration to IMS Version 9 is complete, in other words, there is no pre-IMS Version 9 systems (online or batch) in the same LPAR, DFSMRCL0 can be removed from LPALIB and IEAVTRML.

> **Note:** Name DFSMRCL0 must be removed from IEAVTRML before the module is removed from LPALIB. If the name is still in IEAVTRML and the module is not in LPALIB, IPL will fail.

2.3 IMS Application Menu

Over several releases, IMS has provided a number of applications that run on TSO and ISPF. This is a trend that will continue for several more releases to come. As such, the growing challenge for a user is remembering how to start each application using its own unique command name and parameters. The IMS Application Menu addresses this issue by providing a common interface to the applications. This allows a quick way to view and invoke the available applications.

There are alternative ways to set up IMS Application Menu. First, the IMS.SDFSEXEC data set may be included in the SYSPROC DD concatenation. This is the library that contains the DFSAPPL application. The IMS menu can then be displayed by typing the following on an ISPF command line:

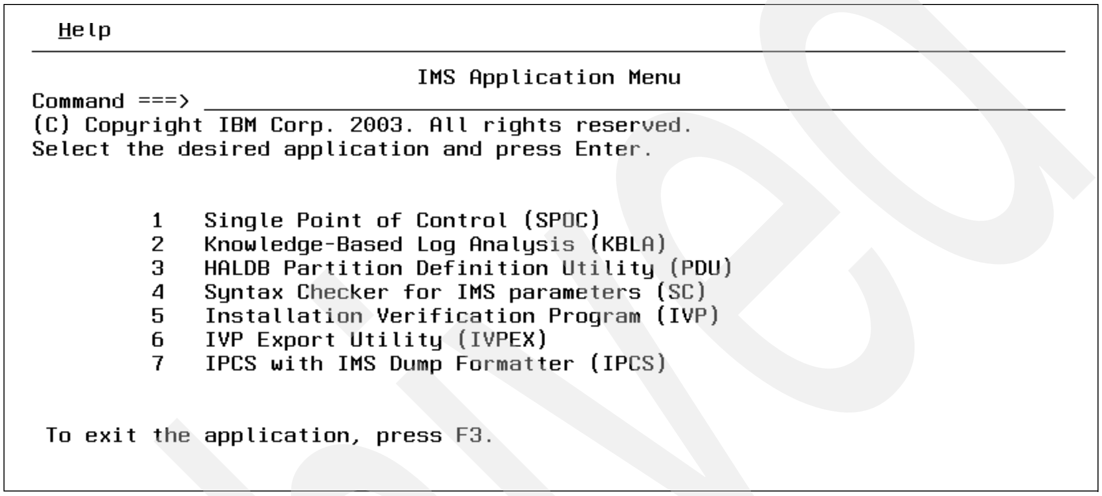
```
TSO %DFSAPPL HLQ(IMS910).
```

Another way is to call the exec directly:

```
EXEC 'IMS910.SDFSEXEC(DFSAPPL)' 'HLQ(IMS910)'
```

The HLQ parameter value must be set to the high level qualifier for IMS Version 9 libraries and it is required on the first use. If it is not specified on subsequent uses, the most recently specified HLQ is used. In our examples, we are having the value IMS910 for the HLQ. The user may want to add a link to their own ISPF menu where the IMS Application Menu can then be invoked as an option.

Figure 2-2 shows the IMS Application Menu in IMS Version 9.



```
  Help
------------------------------------------------------------------------------
                             IMS Application Menu
Command ===> __________________________________________________________________
(C) Copyright IBM Corp. 2003. All rights reserved.
Select the desired application and press Enter.


         1   Single Point of Control (SPOC)
         2   Knowledge-Based Log Analysis (KBLA)
         3   HALDB Partition Definition Utility (PDU)
         4   Syntax Checker for IMS parameters (SC)
         5   Installation Verification Program (IVP)
         6   IVP Export Utility (IVPEX)
         7   IPCS with IMS Dump Formatter (IPCS)


 To exit the application, press F3.
```

Figure 2-2 IMS Application Menu

To select an application from the IMS Application Menu, type the number of the application and press Enter or position the cursor on the desired application and press Enter. The IMS Application menu allows an easy invocation of the following applications:

- TSO Single Point of Control (TSO SPOC)

  By selecting this option you can invoke a TSO SPOC session. TSO SPOC was introduced with IMS Version 8, refer to IBM Redbook *IMS Version 8 Implementation Guide - A Technical Overview of the New Features,* SG24-6954 for more information.

- Knowledge Based Log Analysis (KBLA)

  KBLA is a new feature introduced with IMS Version 9. Refer to Chapter 11, “Knowledge Based Log Analysis (KBLA)” on page 179 for more information about KBLA.

- HALDB Partition Definition Utility (PDU)

  PDU was introduced with High Availability Large Database feature in IMS Version 7. Refer to IBM Redbook *The Complete IMS HALDB Guide, All You Need to Know to Manage HALDBs,* SG24-6945 for more information about HALDB PDU.

- Syntax Checker

  Syntax Checker was introduced with IMS Version 8, refer to IBM Redbook *IMS Version 8 Implementation Guide - A Technical Overview of the New Features,* SG24-6954 for more information. IMS Version 9 provides enhancements to Syntax Checker, and we describe them later in this chapter.

- Installation verification program (IVP)

  You can also access the IVP dialog from the IMS Application Menu. IVP has some changes and enhancements in IMS Version 9. They are described later in this chapter.

- IVP Export Utility

  IVP Export Utility is new with IMS Version 9 and is used to export IVP variables pre-version 9 releases. These variables can be imported into IMS Version 9, easing migration.

- IPCS with IMS Dump Formatter

  Access to IPCS with IMS Dump Formatter is available from the IMS Application Menu. IPCS panels have some enhancements for IMS Version 9. They are described in 12.3.8, “IPCS enhancement” on page 219.

2.4 Installation verification program (IVP) enhancements

For IMS Version 9, the IMS installation verification program (IVP) provides the following enhancements:

- New sub-option for CSL
- The “E” IVP steps support creating of the DBRC default module DFSIDEF0.
- The “O” IVP steps support for a Common Queue Server (CQS) sample application.
- The “P” IVP steps provide an Enhanced Command Environment sample application.
- The IVP help text for the IVP steps:
  - Uses the variables from variable gathering.
  - Uses color to emphasize important information.
- Support for JES3 job classes is improved, and problems encountered by the IVP jobs that cannot run in a JES3 environment have been eliminated.
- SMS storage class and management class parameters are available for allocating all data sets.
- A new and separate high level qualifier variable, IXUVSMHQ, for VSAM data sets is introduced.
- New steps are introduced to complete and test the install of the IPCS for IMS dumps.
- IVP variables can be exported and imported between IMS releases for easier migration.
- The IVP for Java has been enhanced to include an example of running the IMS Java IVP for remote database services. See *IMS Version 9: IMS Java Guide and Reference,* SC18-7821 for details.

2.4.1 IVP sub-options with IMS Version 9

The supported sub-options in IMS Version 9 include the following:

- Use IRLM in IVP Applications

  The default is to use this sub-option (marked by a slash) for DB batch, DBCTL, DB/DC, and DB/DC with XRF. This sub-option is not available for DCCTL. If you select this sub-option, the IRLM is defined during system definition and the IVP is run using the IRLM for the single-lock manager. If you do not select this sub-option, the IRLM is not used and program isolation (PI) is used as the single-lock manager. Use of IRLM is only required if you plan to use block-level data sharing. Optionally, you can elect to use the IRLM, instead of PI, as the single-lock manager. If you select IRLM, IVP creates a simulated inter-CPC block-level data sharing configuration using two IRLMs.

- Use Fast Path in IVP Applications

  The default is to use this sub-option (marked by a slash) for DBCTL, DCCTL, DB/DC, and DB/DC with XRF. This sub-option is not available for DB batch.

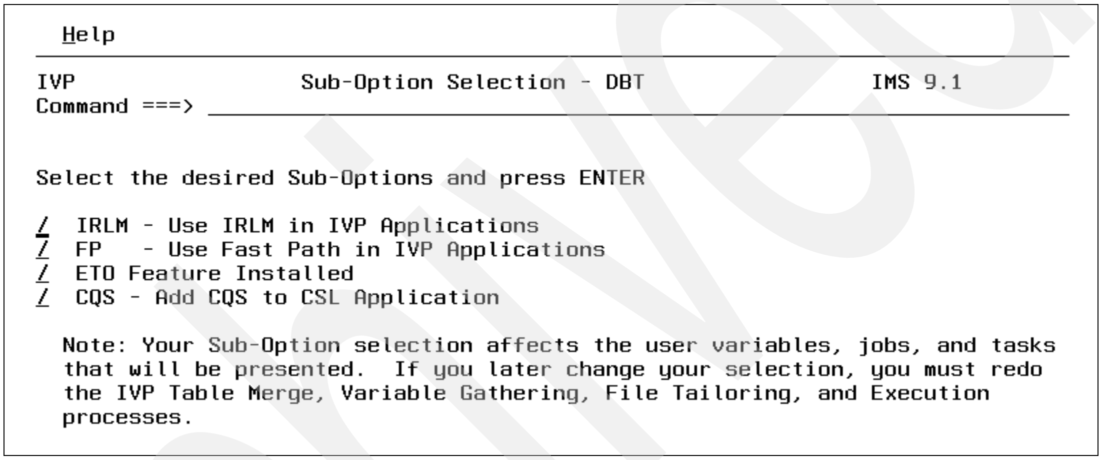
- ETO Feature Installed

  For DCCTL, the default is not to use this sub-option (no slash). For DB/DC and DB/DC with XRF, the default is to use this sub-option (marked by a slash). This sub-option is not available for DB batch or DBCTL.

- Add CQS to CSL Application

  The default is not to use this sub-option (no slash). If you select this option, the IVP adds the necessary jobs and tasks to the CSL sample application to use CQS. This option is new in IMS Version 9.

Figure 2-3 shows the sub option selection panel in IVP.



```
  Help
 ------------------------------------------------------------------------------
 IVP                Sub-Option Selection - DBT                 IMS 9.1
 Command ===> __________________________________________________________________


 Select the desired Sub-Options and press ENTER

 /  IRLM - Use IRLM in IVP Applications
 /  FP   - Use Fast Path in IVP Applications
 /  ETO Feature Installed
 /  CQS - Add CQS to CSL Application

   Note: Your Sub-Option selection affects the user variables, jobs, and tasks
   that will be presented.  If you later change your selection, you must redo
   the IVP Table Merge, Variable Gathering, File Tailoring, and Execution
   processes.
```

Figure 2-3 Sub option selection panel in IVP

2.4.2 The new “E” IVP steps

The new “E” IVP steps include:

- IV_E207J JOB

  A job for creating the DFSIDEF0 module for DBRC default value.

- IV_E304J JOB

  Add CQS Members to IMS.PROCLIB. This adds the following PROCLIB members:

DFSSQ000	Shared queues PROCLIB member
CQSIP000	CQS initialization parameters
CQSSL000	CQS local structure definition
CQSSG000	CQS global structure definition

- IV_E307J JOB

  Define CFRM policy for CQS to z/OS. Look at the information rather than run the job. Your installation may already have a CFRM policy set up and will only need to add the information pertaining to CQS.

2.4.3 The new “O” IVP steps

The new “O” IVP steps include:

- IV_O102J JOB

  Allocate CQS Execution Data Sets. This step defines the allocation of the CQS execution data sets:

  - System checkpoint data set
  - Structure recovery data sets

- IV_O203J JOB

  Start CQS.

- IV_O205T: TASK

  SPOC Sample I. This updates the help text to show the CQS as part of the IMSplex.

- IV_O215J: JOB

  Start DB/DC Region IVP1. This brings up shared queues.

2.4.4 The new “P” IVP steps

The enhanced command environment removes the requirement for a Resource Manager and also supports a non-Parallel Sysplex configuration. The use of type-2 commands still requires the Operations Manager (OM) and Structured Call Interface (SCI) address spaces to be enabled. The IVP introduces a new set of “P” steps to verify the installation and demonstrate this new environment. Two SPOC samples are also included.

For the DB/DC environment, the full set of “P” steps are as follows:

IV_P001T: NOTE	Introduction - Enhanced Command Environment
IV_P101J: JOB	Allocate Data Sets
IV_P102J: JOB	Initialize RECON / Register Data Bases
IV_P103J: JOB	Data Base Initial Load
IV_P104J: JOB	Batch Image Copy
IV_P215J: JOB	Start DB/DC Region IVP1
IV_P217T: TASK	SPOC Sample I
IV_P218T: IVP1	Cold Start IMS
IV_P221T: SPOC	SPOC Sample II
IV_P230T: IVP1	Stop IMS with a /CHE FREEZE
IV_P232T: MVS	Shut Down SCI - Shut down OM and SCI using z/OS command
	`F IVP91SCI,SHUTDOWN CSLPLEX`
IV_P401J: JOB	Scratch Data Sets

Additionally, IMS PROCLIB members can be displayed by issuing “SPR” or “R” command on the IVP Task that Starts IMS (IV_P215J for DB/DC). “SPR” is an acronym for Special Processing Routine.

Refer to Chapter 7, “Type-2 commands and enhanced command environment” on page 115 for detailed information about the enhanced command environment.

2.4.5 DFSIVPEX IVP utility to export and import the variables

There are over 280 variables that can be set by a user in the IVP variable gathering phase during installation. Specifying these values can be a tedious and time consuming task. IMS

Version 9 introduces a new IVP utility, DFSIVPEX, that can save the variables from one release and reload them into the IMS Version 9 IVP. The utility uses the table data set IMS.INSTATBL which contains the ISPF tables that are read and updated by the IVP dialog.

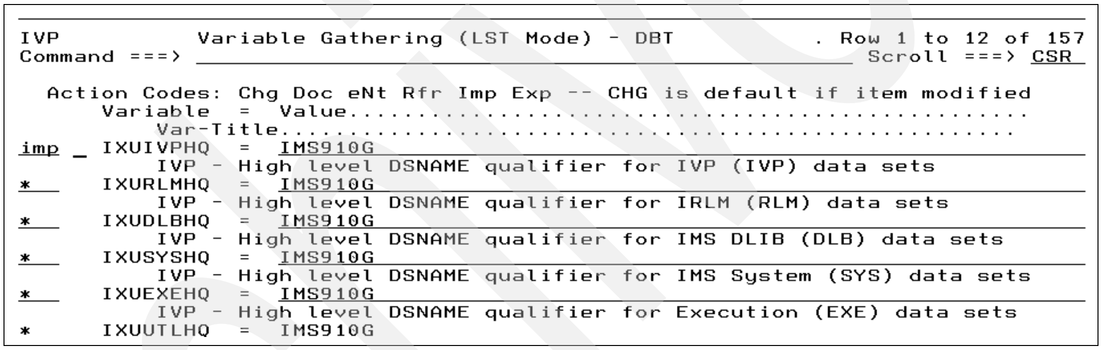
When using the export capability, the IVP variables are copied from the INSTATBL data set associated with the high level qualifier specified in the panel and written to the export data set name that is also identified. The export option will work on any supported IMS IVP release.

When using the import capability, the variables are written from the export data set name to the table data set associated with the specified high level qualifier. This option only reloads the variables into the IMS Version 9 IVP.

In the Variable Gathering phase of the IVP, the panel provides an action code option to either Import or Export the variables. The Figure 2-4 shows that an Import is being requested to a target IMS Version 9 IVP environment.

You can invoke the DFSIVPEX either from IMS Application Menu by selecting option 6 or by issuing the following command from ISPF option 6. Replace hlq in the example by your high level qualifier of IMS system libraries.

```
EXEC ‘hlq.SDFSEXEC(DFSIVPEX)’ ‘HLQ(hlq)’
```



```
IVP           Variable Gathering (LST Mode) - DBT          . Row 1 to 12 of 157
Command ===> ________________________________________________ Scroll ===> CSR
 
  Action Codes: Chg Doc eNt Rfr Imp Exp -- CHG is default if item modified
      Variable  =  Value............................................
          Var-Title.............................................
imp _ IXUIVPHQ  =  IMS910G
          IVP - High level DSNAME qualifier for IVP (IVP) data sets
*     IXURLMHQ  =  IMS910G
          IVP - High level DSNAME qualifier for IRLM (RLM) data sets
*     IXUDLBHQ  =  IMS910G
          IVP - High level DSNAME qualifier for IMS DLIB (DLB) data sets
*     IXUSYSHQ  =  IMS910G
          IVP - High level DSNAME qualifier for IMS System (SYS) data sets
*     IXUEXEHQ  =  IMS910G
          IVP - High level DSNAME qualifier for Execution (EXE) data sets
*     IXUUTLHQ  =  IMS910G
```

Figure 2-4 Import function

The dialog requests the name of the IVP export data set that was used to save the variables from another system and then compares the export versus import environments. If they do not match, an option is provided to cancel the import or to continue.

2.4.6 IVP help text enhancements

IMS Version 9 provides a customized help text for the IVP steps. In prior releases, generic names (for example, IMS.SDFSRESL) were always used to represent data set names and other items. This sometimes caused confusion and errors because the names in the help text did not necessarily match the names in the user environment. IMS Version 9 resolves this ambiguity by displaying the actual user-defined high level qualifiers (for example, IMSPSA.IM0B.SDFSRESL) as specified in the variable gathering phase.

The addition of color to the help text is also designed to emphasis important information that may otherwise be missed.

2.4.7 JES3 improvements in IVP

The IMS IVP process has historically experienced problems when running in a JES3 environment due to the IVP support of a single character job class where JES3 supports job classes that can be from one to eight characters.

In IMS Version 9, the job class variables IXUJCLAS (JOB statement CLASS parameter for IVP jobs) and IXUJCLS2 (JOB statement CLASS parameter for SYSDEF STAGE2 jobs) have been modified based on the variable IXUJESTY (JES version) for a particular environment.

If IXUJESTY is set to JES2 then the IVP will continue to function as before:

- The job class variables will accept only one character.
- The “CLASS=” parameter will be added to the JOB card.

If IXUJESTY is set to JES3 then:

- The job class variables will accept up to 8 characters.
- The “CLASS=” parameter will not be added to the JOB card, rather a //*MAIN CLASS= statement will be added to the JCL following the JOB card.

Another problem for the JES3 users has been that the IVP jobs use IDCAMS to define data sets in one step and reference these data sets in later steps with DISP=SHR. This can cause problems in a JES3 environment. In IMS Version 9, the DISP processing for all jobs that allocate data sets will be changed as follows:

- The data sets will be allocated using IEFBR14 with a DISP=(,PASS).
- Any references to the data set in subsequent job steps will also use a DISP of PASS.
- The last DD statement to reference the data set will catalog the data set as needed.

This change applies to both JES2 and JES3 environments.

2.4.8 IVP data set allocation enhancements

IMS Version 9 introduces some enhancements to facilitate data set allocations.

The first change deals with the use of SMS. In prior releases, SMS allocation of data sets was only done for the IVP sample application data sets. This has now been extended to allow SMS storage and management class parameters to be used when allocating all data sets.

The second enhancement affects VSAM allocations. In many customer shops, VSAM data sets have specific high level qualifiers that differ from those of non-VSAM data sets. For these environments, the IVP jobs had to be converted to accommodate the change. This task was often frustrating, time consuming and error prone. IMS Version 9 addresses the issue by introducing a new variable IXUVSMHQ as part of the variable gathering phase. This is a required variable and will be used when allocating and referencing VSAM data sets.

2.4.9 IVP enhanced dump formatter setup

There are additional steps to complete and test the installation of IPCS for IMS dumps. The new steps are the following:

IV_D216T	Example for IPCS ISPF data set concatenation.
IV_D217T	Example for defining MVS dump options.
IV_H311T	Task for bringing up IPCS session for sample dump in DB/DC and XRF environments.
IV_J309T	Task for bringing up IPCS session for sample dump in DCCTL environment.
IV_G309T	Task for bringing up IPCS session for sample dump in DBCTL environment.

2.4.10 Miscellaneous IVP changes

The following changes in IVP are also introduced with IMS Version 9:

- Remove TSO VTAM definitions from VTAM APPL definition samples

  The IVP dialog member IV_D301T now contains VTAM APPL definition samples only for IMS. The VTAM definitions for TSO have been removed because they are not needed. For the IVP to work, TSO had to have already been installed. Additionally, for DBCTL environments where IMS VTAM definitions are not needed, IV2D301T will not be presented.

- Allow a blank value for variable IXUUSRID

  In prior releases, when IXUUSRID was defined in the variable gathering phase without a value, the IVP dialog automatically assigned it the value of the submitter's TSO id. This caused the dialog to include USER= parameters in the job cards of all the JCL members of INSTALIB. To explicitly prevent the USER= parameter from being included in the job card, the variable IXUUSRID had to be explicitly set to blank every time the variable gathering phase was run. In IMS Version 9, the IVP dialog recognizes and accepts a null value.

2.5 Syntax Checker enhancements

The Syntax Checker provides an ISPF user interface that enables you to define, verify, and validate parameters and their values in the DFSPBxxx member of IMS.PROCLIB. With IMS Version 9, new IMS Version 9 parameters are supported, as well as IMS Versions 7 and 8. Additionally, checking is now provided also for the following two PROCLIB members:

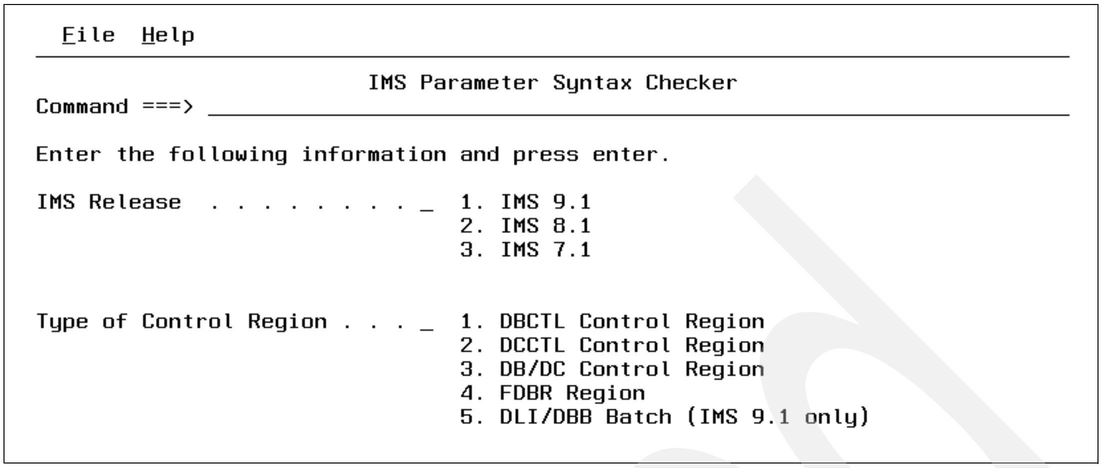
- Data communications options member, DFSDCxxx
- CQS address space member for shared queues, DFSSQxxx.

With Syntax Checker, you can:

- View default values for parameters.
- Use the new select command “P” for process which enables you to interrupt the processing of the current member and begin processing the selected member.
- Update the DFSPBxxx members for new IMS Version 9 parameters.
- Process the DFSPBxxx, DFSDCxxx and DFSSQxxx members.

Depending on the parameter member being processed, additional information such as the release of IMS and the type of control region may be necessary to correctly process the parameters. If the Syntax Checker cannot determine the information required, the user will be prompted for the information through additional panels. Syntax Checker can be used to assist with IMS release to release migrations when converting PROCLIB members.

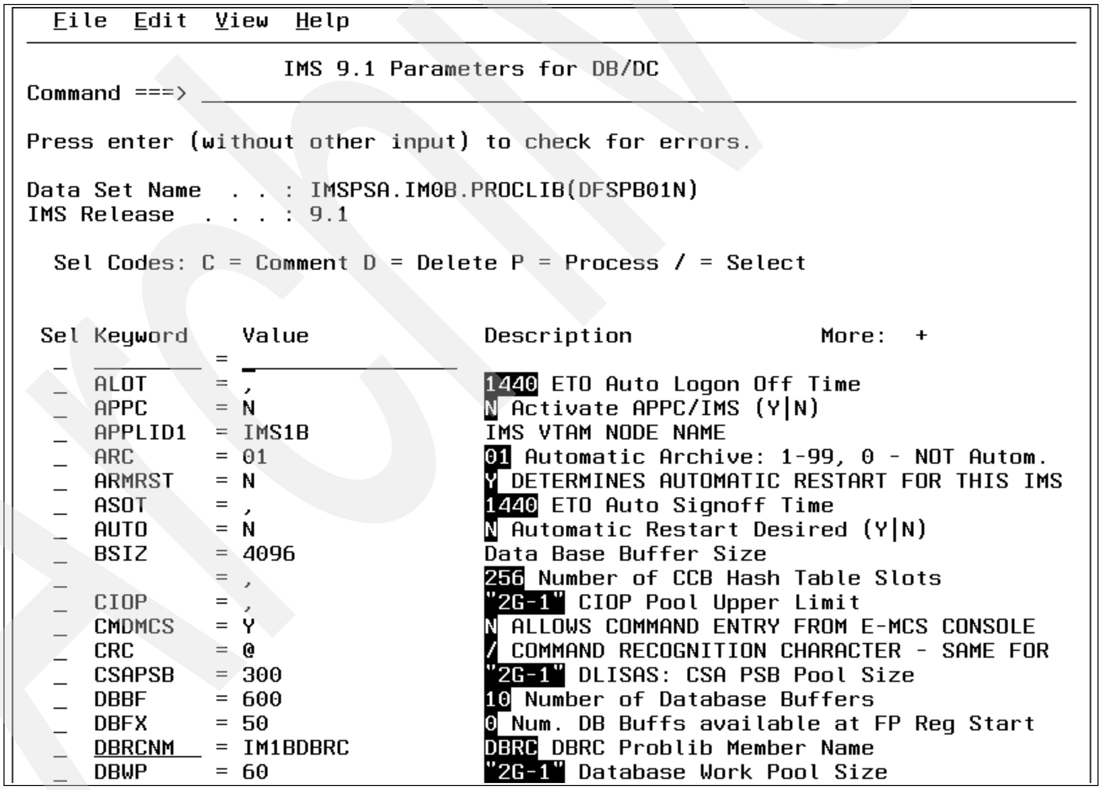
New with IMS Version 9 is that you can process the DFSPBxxx members for FDBR and DLI/DBB environments. Figure 2-5 on page 25 shows the selection panel with new options 4 and 5.



```
  File  Help
─────────────────────────────────────────────────────────────────────────────
                        IMS Parameter Syntax Checker
Command ===> ________________________________________________________________

Enter the following information and press enter.

IMS Release  . . . . . . . . _  1. IMS 9.1
                                2. IMS 8.1
                                3. IMS 7.1


Type of Control Region . . . _  1. DBCTL Control Region
                                2. DCCTL Control Region
                                3. DB/DC Control Region
                                4. FDBR Region
                                5. DLI/DBB Batch (IMS 9.1 only)
```

Figure 2-5 Environment selection panel for Syntax Checker

Figure 2-6 is an example of the Syntax Checker keyword display panel. It displays the keywords and values and indicates any errors. IMS Version 9 adds the option to show the default values for keywords. F6 key provides the toggling capability which changes the display back and forth from displaying the defaults to not displaying them.



```
  File  Edit  View  Help
──────────────────────────────────────────────────────────────────────────────
                   IMS 9.1 Parameters for DB/DC
Command ===> _________________________________________________________________

Press enter (without other input) to check for errors.

Data Set Name  . . : IMSPSA.IM0B.PROCLIB(DFSPB01N)
IMS Release  . . . : 9.1

  Sel Codes: C = Comment D = Delete P = Process / = Select


 Sel Keyword    Value              Description              More:  +
  _  ________ = _________________
  _  ALOT     = ,                  1440 ETO Auto Logon Off Time
  _  APPC     = N                  N Activate APPC/IMS (Y|N)
  _  APPLID1  = IMS1B              IMS VTAM NODE NAME
  _  ARC      = 01                 01 Automatic Archive: 1-99, 0 - NOT Autom.
  _  ARMRST   = N                  Y DETERMINES AUTOMATIC RESTART FOR THIS IMS
  _  ASOT     = ,                  1440 ETO Auto Signoff Time
  _  AUTO     = N                  N Automatic Restart Desired (Y|N)
  _  BSIZ     = 4096               Data Base Buffer Size
  _           = ,                  256 Number of CCB Hash Table Slots
  _  CIOP     = ,                  "2G-1" CIOP Pool Upper Limit
  _  CMDMCS   = Y                  N ALLOWS COMMAND ENTRY FROM E-MCS CONSOLE
  _  CRC      = @                  / COMMAND RECOGNITION CHARACTER - SAME FOR
  _  CSAPSB   = 300                "2G-1" DLISAS: CSA PSB Pool Size
  _  DBBF     = 600                10 Number of Database Buffers
  _  DBFX     = 50                 0 Num. DB Buffs available at FP Reg Start
  _  DBRCNM   = IM1BDBRC           DBRC DBRC Problib Member Name
  _  DBWP     = 60                 "2G-1" Database Work Pool Size
```

Figure 2-6 Syntax Checker sample

The new "P" line command option stands for process and is valid on all keywords whose value is used to identify another member in the IMS.PROCLIB data set. This causes Syntax Checker to interrupt the processing of the current member and begin processing the member selected. At the completion of the processing of the selected member, the user can select F3 key to return processing to the original member that was being processed prior to the interruption.

If the member selected is not processed by Syntax Checker a panel will be displayed asking if the member should be edited using the ISPF editor (a blank member will be created if the member doesn't exist). The member can be edited in the current data set or in a different data set.

2.6 Documentation changes

The following changes and additions have been made to IMS manuals:

- New books:
  - *IMS Version 9: IMS Connect Guide and Reference,* SC18-7821
  - To complement the IMS Version 9 library, a new book will be available in January 2005:

    An Introduction to IMS: Your Complete Guide to IBM's Information Management System, by Dean Meltz, et. al. Prentice Hall. ISBN 0131856715.
- New chapters:
  - Integrated HALDB online reorganization is described in *IMS Version 9: Administration Guide: Database Manager,* SC18-7806.
  - New commands are in command reference manual, *IMS Version 9: Command Reference,* SC18-7814
  - New part in DBRC manual, *IMS Version 9: Database Recovery Control (DBRC) Guide and Reference,* SC18-7818
  - New exits in customization guide, *IMS Version 9: Customization Guide,* SC18-7817
- Changed title (new name):
  - *IMS Version 9: IMS Java Guide and Reference,* SC18-7821
- Reorganization:
  - *IMS Version 9: Installation Volume 1: Installation Verification,* GC18-7822
  - The chapter "DLIModel Utility" moved from *IMS Java Guide and Reference* to *IMS Version 9: Utilities Reference: System,* SC18-7834
  - The DLIModel Utility messages that were in *IMS Java Guide and Reference* have moved to *IMS Version 9: Messages and Codes Volume 1,* GC18-7827

When ordering IMS Version 9, you are entitled to the following books (one hardcopy included per order):

- *IMS Version 9: Installation Volume 1: Installation Verification,* GC18-7822
- *IMS Version 9: Installation Volume 2: System Definition and Tailoring,* GC18-7823
- *Licensed Programming Specification,* GC18-7825
- *IMS Version 9: Release Planning Guide,* GC17-7831
- *Licensed Product Kit,* LK3T-7213 (CD)

The following manuals are available as a soft copy only:

- *IMS Version 9: Base Primitive Environment Guide and Reference,* SC18-7813
- *IMS Version 9: Common Queue Server Guide and Reference,* SC18-7815
- *IMS Version 9: Common Service Layer Guide and Reference,* SC18-7816
- *IMS Version 9: IMS Java Guide and Reference,* SC18-7821

All remaining books and additional entitled books are available for a fee.

2.6.1 IMS Information Center

DB2® Information Management Software Information Center for z/OS Solutions contains documentation for IMS and IMS Tools. It can be found in the Internet at:

```
http://publib.boulder.ibm.com/infocenter/dzichelp/index.jsp
```

3

Integrated HALDB online reorganization

It has been the goal of most database administrators to avoid having to work an overnight or weekend shift to perform database reorganization. It has also been the goal of many Chief Information Officers to achieve the same thing, due to an outage that must be taken during a reorganization and its potential impact on business operations. Availability of data in a database should not be affected by reorganization. All of the data should be available all the time.

IMS Version 9 helps both groups of people achieve their goal — the ability to reorganize a database without taking it offline. Integrated HALDB online reorganization (OLR) provides the capability to reorganize your HALDB databases as a background process, while they continue to be read and updated by online and batch programs. OLR is provided as part of the base of IMS Version 9, and is therefore available for immediate use in your organization.

The benefits of using integrated HALDB online reorganization include:

- Elimination of database outages for reorganization
- 100% availability with zero outages of your database during reorganization
- Applications are unaffected; they never get data unavailable conditions due to online reorganization
- Full integrity and recoverability are maintained

3.1 Overview of integrated HALDB online reorganization

In the beginning, when data is initially loaded into an IMS database, the physical location of each segment is chosen by IMS to minimize the number of I/Os needed to access records and segments in the database. As programs modify, insert, and delete segments over time, the cost of accessing the data in the database tends to increase, leading to longer processing times for application programs accessing these databases.

Database reorganization is the process of relocating segments in a database, with the goal of optimizing the I/O cost for accessing the data in the database. In the past, an offline reorganization process has been necessary for full function databases, including HALDB. This process reads every segment in the database, writes them to a sequential data set, re-initializes the database data sets, and reloads the data into the database in this new sequence. With this process, reorganization needed to be an offline process. That is, access to the database was stopped prior to the reorganization, and started again only after the reorganization had completed and an image copy (preferably) had been taken.

Integrated HALDB online reorganization (OLR) allows you to reorganize your HALDB databases while they are concurrently accessed for read or update processing by application programs. It can be used for PHDAM and PHIDAM database organizations, in an IMS TM/DB or DBCTL environment. The databases can also be reorganized when being accessed in a data sharing environment, as well as being available to Extended Recovery Facility (XRF) and Remote Site Recovery (RSR) environments.

The OLR process does not support changes to the data definitions for HALDBs. It provides only reclustering and space distribution advantages, although DBDS space allocation changes are allowed.

A reorganization is initiated by operator command, using either the classic type-1 or the new type-2 commands. The progress of the reorganization can also be checked through operator command.

The pace of an online reorganization is determined by the RATE specified on the OLR operator commands, system resources and system utilization. The amount of updates to the IMS log and system contention can also affect the speed at which an OLR runs. The rate can be changed when adjustments are desired depending on system contention (that is, slow down the OLR) or lack of system contention (i.e., speed up the OLR). The reorganization process can also be temporarily paused if required, and resumed at some later time.

OLR uses two sets of database data sets during the reorganization process. One set of database records, known as the unit of reorganization (UOR), is read from the input data sets (initially the A-J and X, if PHIDAM, data sets) and then written to the output data sets (initially the M-V and Y, if PHIDAM, data sets), as shown in Figure 3-1 on page 31. When all the segments in the current UOR have been processed, the current location (cursor) is moved forward, and the set of records in the next UOR is read. A subsequent reorganization will read the M-V (and Y, if PHIDAM) data sets as the input, and write the A-J (and X if PHIDAM) data sets. IMS, through DBRC, automatically and transparently takes care of which data sets are the active copy of the database. Both sets of data sets are used during OLR. At the end of the reorganization, the old data sets may be discarded (to save DASD space).

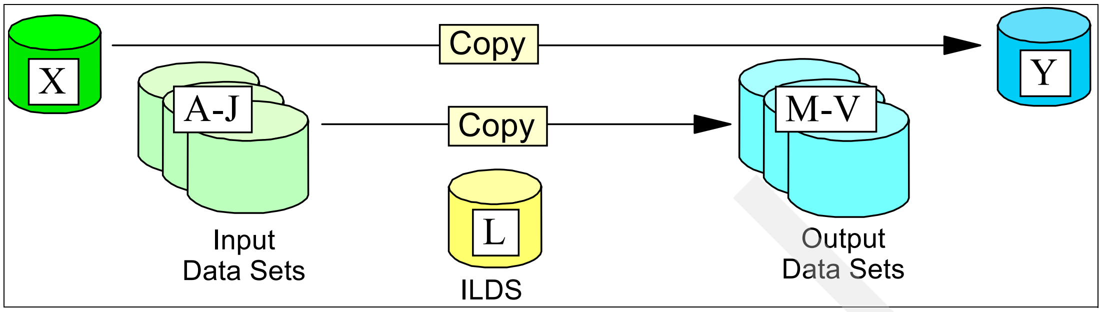


Figure 3-1 Data sets in use during an online reorganization

Concurrent IMS updates are allowed while records are moved and reorganized by OLR. IMS determines whether to expect a given database record to be in either the input or the output data sets based on whether the record is either beyond or prior to the cursor, as shown in Figure 3-2. This is completely transparent to all application programs accessing the database.

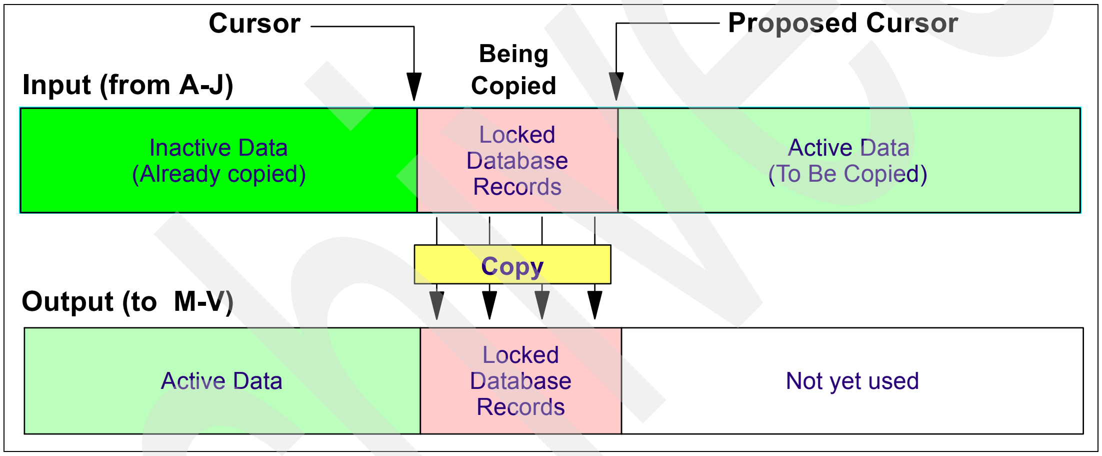


Figure 3-2 Use of the cursor when reorganizing from the input to the output data sets

For possible recovery of either an input or an output data set, all database changes are logged. The indirect list data set (ILDS) list entries (ILEs) are updated with new target RBAs, but these changes are not logged. Databases can therefore be recovered using the standard recovery utilities or the IMS Database Recovery Facility (DRF) tool. OLR also provides full DBRC support. Note that a standard batch database reorganization does not log any changes.

Databases are reorganized at the partition level, with each partition independently reorganized. This provides flexibility; partitions can be reorganized in parallel, and it is not required to reorganize the whole database at once. The online reorganization runs in the DLISAS address space, with a separate process (ITASK) used for each partition being reorganized.

OLR can be used to reorganize the PHDAM data component, the PHIDAM data component and primary index, including HALDBs which have secondary indexes and logical relationships. Database with secondary indexes can be reorganized, but the secondary index (PSINDEX) cannot be reorganized. In a data sharing environment, online reorganization may be run on any data sharing system, with concurrent access and update allowed to these databases from other systems.

Before the OLR starts, there is a single active set of data sets for the HALDB partition - either the A-J (and X) set or the M-V (and Y) set. The data sets in the other set contain no useful information. At the end of the OLR initialization process, the OLR status of the partition is recorded in the RECON with a cursor-active status. Once the OLR initialization process has completed, the original active set of data sets is treated as the input set and the other (paired) set becomes the output set. Both sets of data sets are active, since they both contain active data as you can see in Figure 3-2 on page 31.

After all database records have been copied, the input data sets become the inactive set. Unless an initial load or an offline reorganization is done, successive OLRs for the partition then alternate between the two sets of data sets.

At the completion of OLR, the input data sets and their contents become obsolete. Image copy and recovery records are no longer applicable for inactive data. The records of prior image copies and log data of inactive data sets remain in the RECON, and can be used for time stamp recovery if required.

3.2 Details of OLR processing

In this section, we describe the detailed setup and processing of HALDBs for OLR.

3.2.1 Data sets used by OLR

OLR extends the group of data sets required to support HALDB. Prior to IMS Version 9, HALDB databases were stored in one or more database data sets. These data sets, usually referred to as the A-J (and X) data sets, were used to store the segments and indexes, and the primary index data, as shown in Figure 3-3.

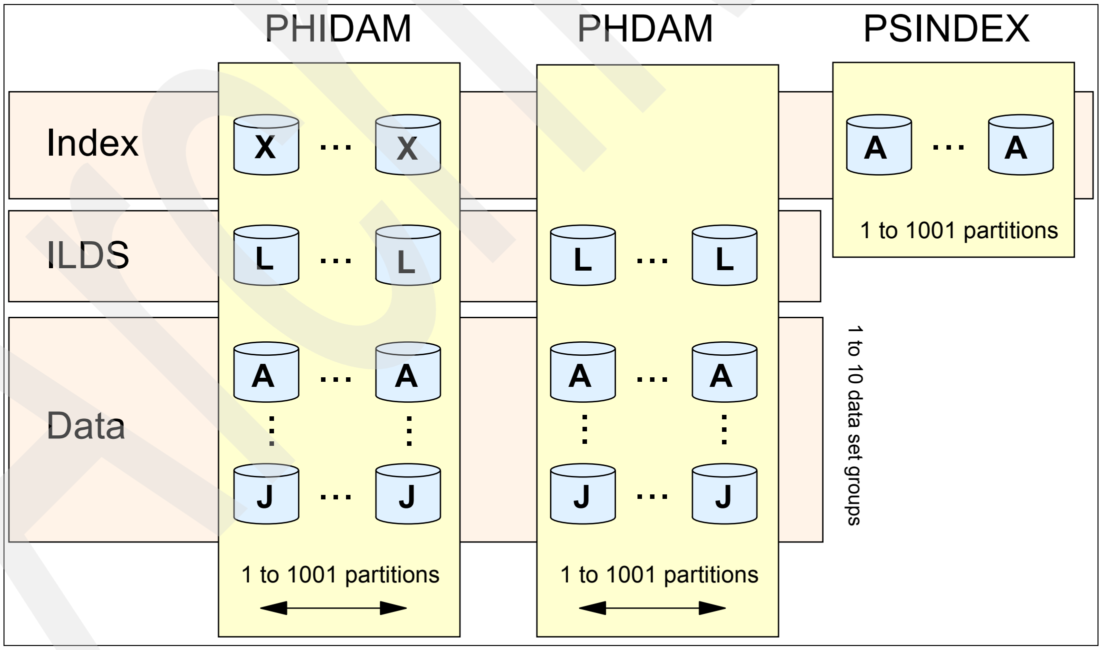


Figure 3-3 Single set of data sets in a HALDB that is not OLR capable

OLR requires the use of two sets of database data sets for each HALDB that is capable of being reorganized by OLR, as shown in Figure 3-4 on page 33. The data is read from the current set of data sets (initially the A-J and X data sets), and written to the other set of data sets (the M-V and Y data sets). The A-J, X and M-V, Y data sets swap roles at the end of each reorganization.

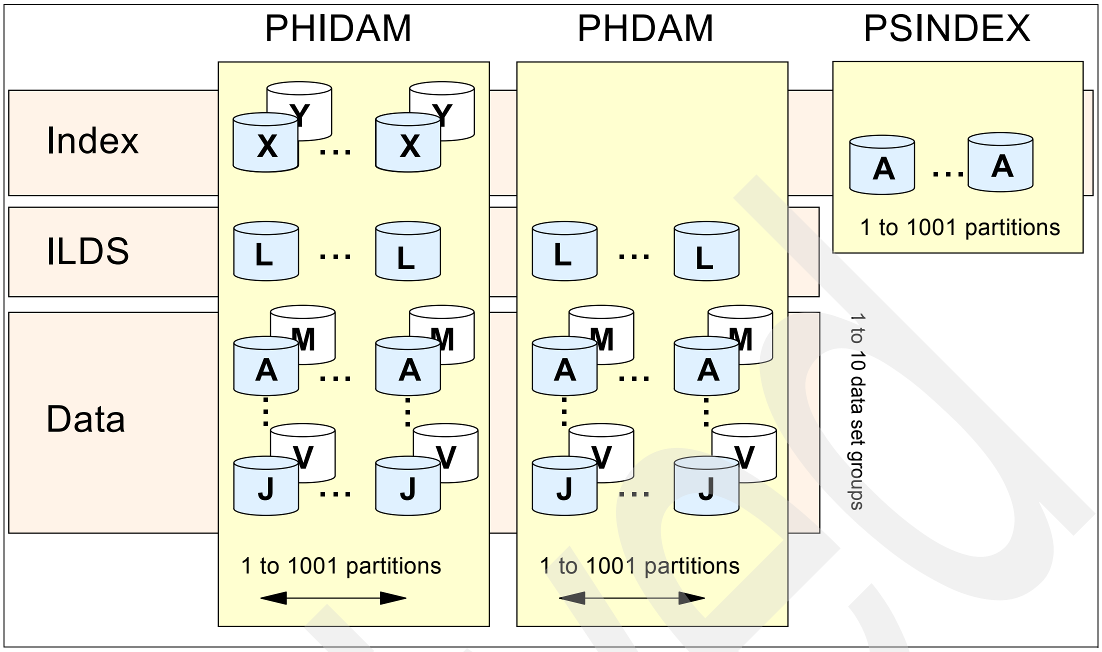


Figure 3-4 Two sets of data sets in HALDB that is OLR capable

Before or after reorganization, the active data sets are either the A-J (and X) data sets, or the M-V (and Y) data sets. These are the data sets being accessed by applications, and the selection of data sets to use is handled transparently by IMS. The inactive data sets are the data sets which are not being accessed by applications.

During a reorganization, all data sets (A-J, X and M-V, Y) are active data sets. The input data set contains the unreorganized data, including both active and inactive data. The output data set contains reorganized data which are active.

The cursor is the dividing line between active data and inactive data. A diagram of the cursor is shown in Figure 3-2 on page 31. The cursor is only used while reorganization is in progress or suspended.

3.2.2 Reorganization by copying segments

The process of database reorganization is simply the relocation of segments in the database, with the goal of reducing the I/O cost when later accessing the segments.

Reorganization is performed by copying segments from their current location to their optimal location in the database. With OLR, segments are read from one set of HALDB data sets (for example A-J, and X), and then written (inserted) back to the other set of HALDB data sets (for example M-V, and Y). The ILDS entries for the secondary index and logical relationship targets are also updated during this process.

The reorganization process is similar to that of a long-running BMP. Segments are read from the database, processed by the application (in this case, OLR), and then written back to the database. To protect the integrity of the information in the database, locking protocols are used to prevent simultaneous access to the data.

Similarly, when segments are copied from the old to the new data sets during an online reorganization, all updates are captured to the IMS logs. Therefore, IMS databases can be recovered across the time of an OLR.

3.2.3 Current position during a reorganization - the cursor

The cursor is used to track the progress of the reorganization through the database, by pointing to the last committed reorganized record. For PHDAM databases, the cursor is a root anchor point (RAP) relative byte address (RBA). All records chained from it and previous RAPs have been copied to the output data sets. For PHIDAM the cursor is a root key. All records with this key and lower keys have been copied to the output data set. The cursor is stored in the block after the first bit map in the input data set.

Figure 3-5 shows the usage of the cursor during a reorganization.

3.2.4 Application access during OLR

When an application is reading data from a HALDB which is being reorganized, the selection of data set to be read is based on the cursor value.

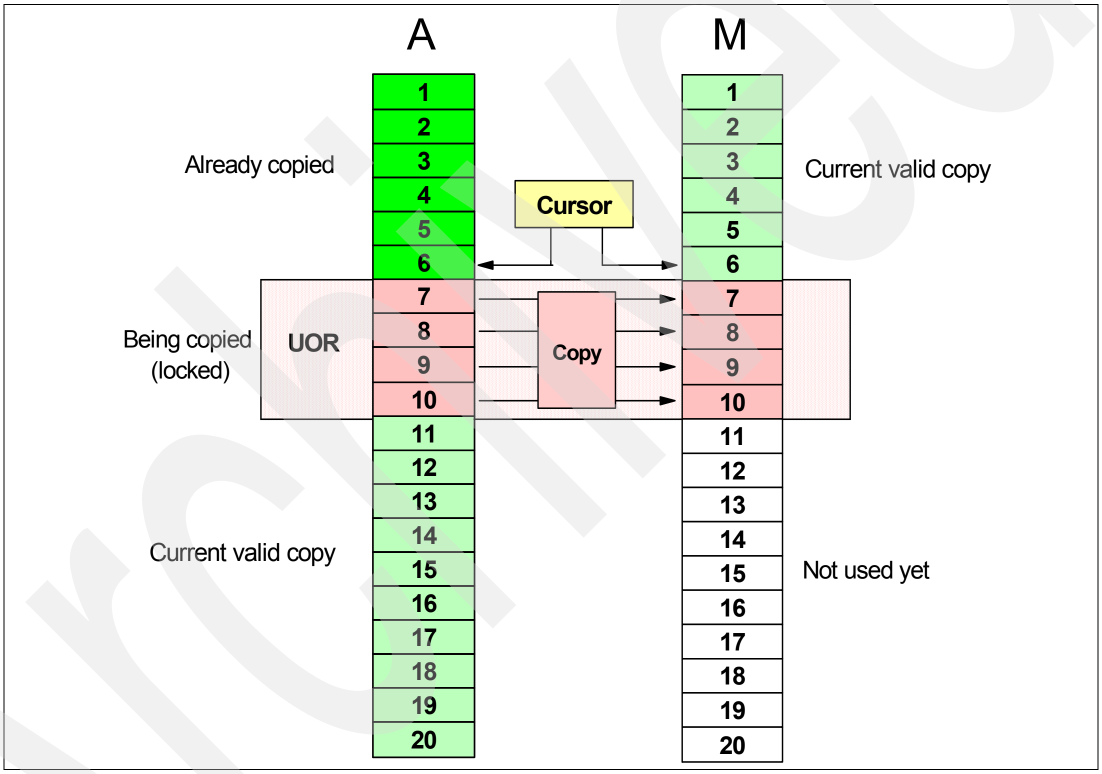


Figure 3-5 Location of access to a database during online reorganization

When reading the database shown in Figure 3-5, where the cursor is on record 6, segments from record 1 through 6 are accessed from the M data set. Segments from record 11 will be accessed from the A data set. Any program attempting to access records 7 through 10 will wait for the online reorganization to finish processing the current unit of reorganization, due to the lock held by online reorganization. When OLR has completed processing that UOR, the locks are released and the program accesses the reorganized information from the M data set. The example is not correct for a PHIDAM with TB pointing. In that case record 6 would be locked too.

3.2.5 Completion of reorganization

When OLR completes, the previously active data sets (A-J and X in our example) become the inactive set, and may be deleted. The other set of data sets (M-V and Y data in our example) become the active set, and the cursor-active status is reset to inactive. Next time this partition is reorganized, online reorganization will reorganize from the currently active M-V and Y data sets to the A-J and X data sets.

3.3 Setting up for OLR

The basic process for OLR involves two parts:

- Setup

  This means allowing OLR for the selected databases, and optionally allocating the second set of data sets.

- Start

  Issuing the commands to invoke OLR.

3.3.1 DBRC setup

Online reorganization must be enabled through DBRC for each of the HALDB databases you want to be able to process with online reorganization. This is achieved by using the **OLRCAP** (online reorganization capable) parameter of either the **INIT.DB** or **CHANGE.DB** command in DBRC. The **OLRCAP** attribute allows online reorganization for all partitions of this database. OLRCAP is the default for new HALDBs registered in IMS Version 9. You can also choose to register a HALDB database as not OLR capable through the use of the **OLRNOCAP** parameter.

If you register a HALDB as capable of being accessed by online reorganization, this database can no longer be accessed from, or even visible to, either an IMS Version 7 system, or from an IMS Version 8 system which has not had the OLR coexistence SPE installed.

The *OLRNOCAP* attribute allows a HALDB to be accessible from IMS Versions 7, 8 and 9. It also prevents online reorganization for this HALDB.

DBRC will check the authorization status for a database before it will allow the *OLRCAP* flag to be set. The command is disallowed if any of the database's partitions are authorized unless MINVERS of 9.1 is set in the RECON. A database cannot have the OLRNOCAP value set if online reorganization is active or if the M-V data sets are active.

To enable OLR for all HALDB databases, the command **CHANGE.DB ALL OLRCAP** can be used. The command skips any HALDB databases that are authorized, and issues warning message DSP0195I for any skipped databases and their partitions. The command **CHANGE.DB ALL OLRNOCAP** is not allowed.

3.3.2 Output data set creation

To simplify the OLR process for a HALDB partition, each of the output data sets can be created automatically by IMS. For each data set group defined in the DBD and for the primary index of a PHIDAM database, the output data set is created if it doesn't already exist as a cataloged data set. The indirect list data set (ILDS) is not created automatically because there is not a corresponding output version of it.

In order to reserve approximately the same amount of space that was reserved for the input data set regardless of the DASD types involved, the output data set's space is requested as a number of OSAM blocks or VSAM records.

If an input data set is SMS-managed, then the corresponding output data set is SMS-managed as well, and the storage class used to create the output data set is the same as that of the input data set. If the input data set has extents on only one DASD volume, the output data set is also created on a single volume. If the input data set has extents on more than one DASD volume, the output data set is created to allow the same number of volumes. If the storage class of the input data set has the guaranteed space attribute, the primary space allocation will be taken on each of the volumes when the output data set is created.

For a non-SMS-managed input data set, IMS automatically creates the output data set only if the input data set is a on a single volume. If you need a multi-volume non-SMS-managed output data set, you must create the data set yourself before starting the OLR.

If an input data set is not SMS-managed and is a single-volume OSAM data set, the output data set is created as though UNIT=SYSALLDA has been specified on a DD statement. This causes the data set to be created on a storage volume or, if no storage volume is available, on a public volume.

If an input data set is not SMS-managed and is a single-volume VSAM data set, the output data set is created on the same volume as the corresponding input data set. This limits the usefulness of automatic data set creation for a non-SMS-managed VSAM data set, because there may not be enough space. Not having enough space is one of the reasons why the reorganization might not start. Note that the situation of not having enough space is not unique to the non-SMS-managed VSAM case; it applies to any type of data set. However, this problem is certainly more likely to occur with non-SMS-managed VSAM because there is only one volume on which IMS tries to create the output data set.

Optionally, you can create and catalog the output data sets prior to the start of online reorganization. You may use preallocation if you want to allocate the data sets on specific volume, change the space allocation, the number of blocks/CIs, the primary and secondary allocations, or, for a PHIDAM primary index, the free space percentage. Note that the PHIDAM primary index CI size cannot be changed during an OLR.

3.4 Starting online reorganization

An online reorganization is started through operator commands. The command can be either a type-1 or type-2 command. The format of type-1 command is:

```
/INIT OLREORG NAME(partname1)
```

The format of the type-2 command is:

```
INIT OLREORG NAME(partname1,partname2,...)
```

Both of these commands have two keywords. They allow you to set the rate of execution and, optionally, to not to delete the input data sets at the completion of the reorganization (delete option is the default).

There are the following steps in the start process for online reorganization of HALDBs:

- Initiation
  - Validation
  - DBRC actions
  - Output data set initialization
- Start processing

3.4.1 Initiation

OLR initiation process consists of the following parts:

- Validation

  Before IMS will begin the online reorganization of a HALDB, the status of the HALDB master and partition is checked. OLR will not start while the partition or master is stopped, locked, /DBRed, /DBDed, or opened with ACCESS=RD or RO. Reorganization will not start if IMS is being shutdown.

- DBRC actions

  Once a partition has completed the OLR validation process, IMS updates the RECON to indicate the partition is to be processed by OLR. The IMS system performing this online reorganization is recorded in the RECON partition record for this database. The output data sets are registered in the RECON if not already registered, and an ALLOC record is created for the output data set (A or M), and a REORG record is created with a start time equal to the ALLOC time.

  For each partition being reorganized, the RECON is updated to indicated that a cursor is active. That is, a reorganization is in progress and must be completed. The IMSID for the system owning the reorganization of this partition is also recorded in the RECONs. Lastly, if running in a data sharing environment, each of the sharing subsystems are notified that a “cursor is active” for the partition.

- Output data set initialization

  After OLR has been started, and after any nonexistent output data sets are created automatically and any preexisting ones are validated by IMS, the output data sets are initialized. Block 1 contains the reorganization number, which is one greater than the previous reorganization number. Block 2 contains the cursor to be used for the reorganization as follows:

  - For PHDAM databases, the cursor is the RBA of last PHDAM RAP reorganized.
  - For PHIDAM databases, the cursor is the key of the last PHIDAM root segment reorganized.

3.5 Reorganization process

The online reorganization process consists of the following tasks:

1. Determine the size of unit of reorganization (UOR)

   The first part of the reorganization process is to determine the size of the unit of reorganization (UOR). The algorithm used is an internal function, and is designed to minimize the impact of OLR processing on other applications accessing this database. During a reorganization, the size of the UOR may be dynamically adjusted to prevent long elapsed times. This is performed internally by IMS, with statistics on the quantity of data processed in each UOR captured to the x’2950’ log record, described further in 3.9.1, “Logging” on page 43.

2. Get database record locks for the UOR

   Database records in the UOR are locked by OLR. For a PHDAM database, locks are taken for each of the RAPs in the UOR. For a PHIDAM database, locks are taken for all the records in the UOR.

   If a lock request cannot be granted, online reorganization does not wait. Instead, it ends the UOR at the previous PHDAM RAP or PHIDAM key. There is one exception. If the lock is the first one in the UOR, online reorganization waits to get the lock. Obviously, it would do no good to end a UOR without having processed any records.

   If online reorganization is the victim in a deadlock, dynamic backout of the UOR is done and the reorganization continues at this UOR. The user does not have to restart the reorganization. This is similar to what happens when an online IMS transaction is the victim in a deadlock.

3. Read and copy each segment in the UOR

   Segments are read from the input data sets, and are inserted into the output data set using the space management algorithm for load mode. When space is not available for a segment in the most desirable block, space may be found for it in a block that is already in the buffer pool. This tends to produce writes in physical sequential order, but there may be exceptions.

   In a data sharing environment, block locks are acquired to prevent the sharing IMSs from updating the blocks in the UOR while OLR is updating them.

   All segments moved during the reorganization process are logged for recovery purposes, and their entry in the ILE is also updated if necessary.

   This process is repeated until the end of the UOR is reached (that is, the proposed cursor value is reached).

4. Commit UOR

   All changes made to the UOR are committed. IMS then updates the cursor and notifies any data sharing systems of the new cursor value as part of this notify. IMS purges the buffers through the standard processes, writes the x'3730' log records, and finally release all the locks.

5. Repeat process until end of partition is reached

   Keep doing all the above until the end of the partition is reached.

3.6 Normal termination and cleanup

When all data has been reorganized, OLR notifies the sharing systems that the reorganization has completed. The sharing subsystems close and deallocate the input data sets before responding to the notify. IMS closes and deallocates the input data sets, closes the input ALLOC record with the STOPTIME, and cleans up the RECON records:

- Set the A or M data set indicator
- Set the cursor to inactive
- Reset the reorganization ownership to 'none'
- Update the REORG record (stoptime = DEALLOC time).

When OLR receives notification responses from the data sharing subsystems, it deletes the data sets unless the NODEL option was specified on the OLR initiation command. OLR waits for up to 5 minutes for notification responses. If no response is received, OLR tries to delete the input data sets, but if it can't do so, due to another system holding them, then OLR just ends normally. The user can optionally delete the data sets at a later time if required.

3.7 Pausing an online reorganization

Once an online reorganization has commenced and the cursor-active status has been recorded, the partition has current data stored across both the A-J (and X) set of data sets as well as the M-V (and Y) set. From this point, the reorganization must eventually be run through to completion. If it becomes impossible to complete an online reorganization for any reason, an offline reorganization must be run in order to reset the online reorganization settings. Note that offline reorganization reload always loads the A-J data sets. Also, the partition cannot be accessed unless both sets of data sets are available if an OLR has not completed.

The online reorganization can be suspended and resumed (terminated and re-initiated) as often as desired through operator commands, as described in 3.8.3, “Terminate an online reorganization” on page 40 and 3.8.1, “Start an online reorganization” on page 39 respectively.

3.8 OLR commands

IMS Version 9 has added a number of new commands for OLR. The commands used with OLR are:

- INITIATE and /INITIATE
- UPDATE and /UPDATE
- TERMINATE and /TERMINATE
- QUERY and /DISPLAY

Both type-1 and type-2 commands have been added, and may be used interchangeably.

Type-1 commands for OLR

The type-1 commands for OLR are **/INITIATE, /UPDATE, /TERMINATE** and **/DISPLAY**. They can be entered via a z/OS console, a master terminal, an LTERM, a CMD or ICMD call, or through OM. Type-1 commands are passed to IMS AOI exit, and are logged to the secondary master terminal.

Type-2 commands for OLR

The type-2 commands for OLR are **INITIATE**, **UPDATE**, **TERMINATE**, and **QUERY**. These commands can be entered via TSO SPOC, the IMS Control Center, a REXX exec, or a user program. The responses are in XML format.

3.8.1 Start an online reorganization

The **INITIATE** command is used to start an online reorganization, and can also be used to reinitiate a previously terminated reorganization.

Type-1 command:

```
/INITIATE OLREORG NAME(partname1)
SET(RATE(100|nn))
OPTION(DEL|NODEL)
```

Type-2 command:

```
INITIATE OLREORG NAME(partname1,partname2,...)
SET(RATE(100|nn))
OPTION(DEL|NODEL)
```

The parameters are:

NAME

Partition name(s).

Only one name may be specified when a type-1 command is used.

RATE

Rate of reorganization.

The **RATE** parameter allows you to alter the pacing of the reorganization. Specifying **RATE(100)** allows OLR to runs at maximum speed. If you specify a rate of less than 100, online reorganization waits after each commit so that average speed of reorganization is nn% of the unconstrained speed. **RATE(100)** is the default.

Examples: For **RATE(50)**; after each commit, reorganization waits for the same period of time as it took to process the previous UOR. If the last UOR was reorganized in 1 second, OLR would wait for the same elapsed time (1 second), run for 1 second, wait 1 second, and so on. For **RATE(25)**; after each commit, OLR waits for 3 times as long as the last interval took. So if a UOR processed in, say, 1 second, OLR would wait 3 seconds, run 1 second, wait 3 seconds, and so on.

OPTION

Whether or not to delete the inactive (original input) data sets upon completion of the reorganization. **OPTION(DEL)** is the default.

3.8.2 Modify a reorganization in progress

Both type-1 and type-2 commands can be used to modify a reorganization in progress.

Type-1 command:

```
/UPDATE OLREORG NAME(partname)
SET(RATE(100|nn))
OPTION(DEL|NODEL)
```

Type-2 command:

```
UPDATE OLREORG NAME(*|partname1,partname2,...)
SET(RATE(100|nn))
OPTION(DEL|NODEL)
```

The parameters for both the type-1 and type-2 commands are the same as for the **INITIATE** command shown in 3.8.1, “Start an online reorganization” on page 39.

3.8.3 Terminate an online reorganization

Both type-1 and type-2 commands can be used to suspend an online reorganization. The reorganization can be restarted at any IMS in the data sharing group some time in the future with the **INITIATE** command.

Type-1 command:

```
/TERMINATE OLREORG NAME(partname)
OPTION(FORCE|ABORT)
```

Type-2 command:

```
TERMINATE OLREORG NAME(*|partname1,partname2,...)
OPTION(FORCE|ABORT)
```

The parameters are:

NAME

Partition name(s).

Only one name may be specified when a type-1 command is used.

OPTION

When to terminate the OLR.

When the **OPTION** parameter is omitted, OLR terminates at the end of the current UOR. When **OPTION(FORCE)** is specified, OLR ends at the next PHIDAM root or PHDAM RAP. When **OPTION(ABORT)** is specified, OLR terminates immediately with a user abend, and the output of the current UOR is backed out.

After processing a **TERMINATE** command, the cursor is still active in the RECONs. Both sets of database data sets are used until the OLR is subsequently completed. The reorganization can be resumed by any IMS in the data sharing group by using the **INITIATE OLREORG NAME(partname)...** command.

The **DEL|NODEL** and **RATE** options, specified at the time the OLR was initiated, are not retained after the **TERMINATE** command, although they may be specified again on the **INIT OLREORG** command used for resumption of the OLR.

3.8.4 Display reorganizations in progress

IMS Version 9 provides a number of commands to query reorganizations in progress.

Type-1 command:

```
/DISPLAY DB name1 name2  ... ALL
```

The response shows 'OLR' in the CONDITIONS field if the partition has a reorganization in progress locally.

Type-2 command:

```
QUERY OLREORG NAME(*|partname1,partname2,...)
STATUS(RUNNING|OWNED|NOTOWNED|UNKNOWN)
SHOW(ALL|RATE|BYTES|STATUS)
```

The parameters are:

NAME

Partition name(s), optional.

The default is “NAME(*)” which returns all partitions with reorganizations in progress.

STATUS

Optional, show partitions with reorganization status.

RUNNING

Show all online reorganizations running on IMSs to which this command is routed.

OWNED

Show those reorganizations owned by other IMSs, as well as those “RUNNING”.

NOTOWNED

Show those reorganizations that are suspended, and thus not owned by any IMS.

UNKNOWN

Show those with unknown status (not authorized on this subsystem).

SHOW

Specifies the output fields to be displayed.

RATE

Returns the RATE value for each reorganization.

BYTES

Returns the number of bytes written to the output data sets.

STATUS

Returns status of RUNNING, OWNED, NOTOWNED, or UNKNOWN for each reorganization.

ALL

Returns output from RATE, BYTES, and STATUS options

There is no equivalent type-1 command which shows all the information available with the type-2 **QUERY** command shown in Example 3-1. The type-1 command **/DIS DB OLR** can be used to display all partitions that have reorganizations in progress locally. The OLR keyword cannot be combined with other conditions.

Example 3-1 QUERY OLREORG command example

```
QRY OLREORG NAME(PVHDJ5A) SHOW(BYTES)
Response:
Partition MbrName    CC BYTES MOVED
PVHDJ5A   IMS2        0      1256256
```

Example 3-2 shows an example of type-1 command **/INIT OLREORG**.

Example 3-2 Type-1 /INIT OLREORG command example

```
/INIT OLREORG NAME(PDHDOKA) SET(RATE(50))
Response:
DFS058I 12:08:07 INITIATE COMMAND IN PROGRESS
DFS2970I - OLR STARTED FOR NAME=PDHDOKA
DFS0725I INIT      OLREORG COMMAND FOR DB DBHDOKA  COMPLETE.
```

Example 3-3 shows an example of **/DIS DB OLR** command.

Example 3-3 Display online reorganization status

```
/DIS DB OLR
Response:
DATABASE  PART      RATE         BYTES  STATUS
DBHDOK01  PDHDOKA     50         86776  RUNNING
*03099/120858*
```

Example 3-4 shows an example of **/DIS DB** command.

Example 3-4 Display database status command

```
/DIS DB PDHDOJA DBHDOK01
Response:
DATABASE  TYPE  TOTAL UNUSED  TOTAL UNUSED ACC  CONDITIONS
DBHDOJ01  PHIDAM                           UP
PDHDOJA   PART                             UP   ALLOCS, OLR
DBHDOK01  PHDAM                            UP
PDHDOKA   PART                             UP   ALLOCS, OLR
PDHDOKB   PART                             UP   ALLOCS
PDHDOKC   PART                             UP   ALLOCS
```

3.8.5 Database commands

There have been changes to the following commands in an online reorganization environment:

- /DBD DB
- /DBR DB
- /STOP DB
- /START DB

If these commands are issued for partitions that are being reorganized, IMS rejects the command, and message DFS0488I RC=58 is issued.

Similarly, if the command is issued for a master database, the command is not processed if any partitions are being reorganized in the IMS processing the command. Message DFS0488I RC=58 identifies any partitions for which the commands have not been processed.

3.9 Logging, operations, I/O errors and restart

In this section, we discuss logging, system operations, I/O errors, and restart after system and reorganization failures in an online reorganization environment.

3.9.1 Logging

All updates to IMS databases being processed by online reorganization are logged, as though the OLR was an application program.

Scheduling and termination

IMS logs an X'08' record when scheduling an online reorganization. This is comparable to the scheduling of any application program (PSB schedule). The PSB name is x'F0' concatenated with the partition name. For example, the PSB name for partition ABCD is 0ABCD.

IMS logs an X'07' log record at the termination of an online reorganization.

Database change log records

IMS logs an X'50' record for inserts and updates to the output data sets. This is the same as the logging performed for an application program change to the database.

The DCB number has X'80' bit on for M-V (and Y) data sets, and the M-V data sets have the low order bit on in the RBA fields.

Commit processing

At the completion of processing for each UOR, IMS logs an x'3730' record.

OLR log records - x'29xx'

The online reorganization x’29xx’ log records are as follows:

Log Record Type	Description
x’2900’	OLReorg command received

Log Record Type	Description
x’2910’	Ownership established
x’2920’	UPDATE OLR command processed
x’2930’	Output data set information
x’2940’	Cursor active
x’2950’	Cursor movement - includes statistics for UOR
x’2970’	Cursor inactive
x’2990’	Ownership relinquished

Statistics log record

Statistics for OLR are available in the x'2950' log records, and are recorded for each UOR. The information stored includes:

- Segments moved in this UOR
- Bytes moved in this UOR
- Total segments moved before this UOR
- Total bytes moved before this UOR
- UOR size in either RAPs (PHDAM) or roots (PHIDAM) depending on the partition
- Locks held by this UOR
- Start time of this UOR
- Execution time (elapsed time) of this UOR
- Time interval waited before this UOR (due to the RATE parameter).

3.9.2 IMS normal termination and restart

If OLR is running when IMS is shutdown with the `/CHECKPOINT FREEZE` or `/CHECKPOINT DUMPQ` command, IMS terminates OLR at the next commit point. The cursor remains active, and ownership is not relinquished.

If the `/CHECKPOINT PURGE` command is used to terminate IMS while an online reorganization is in progress, IMS waits for OLR to complete before terminating.

When IMS is normally restarted (`/NRE`) after termination with OLR active, IMS authorizes, allocates, and opens all input and output data sets, and resumes OLR automatically.

3.9.3 I/O errors

Should IMS encounter a write error during an online reorganization, IMS processes it in the same way as other database write errors. That is, IMS will write to the I/O toleration buffer, create an EEQE, notify any sharing subsystems, and continue processing the reorganization.

If there is a failure in an online reorganization, including a database read error, the processing of the UOR is ended, and dynamic backout is invoked to backout all the in-flight work for this UOR. IMS then relinquishes ownership of the online reorganization, but leaves the cursor still active indicating that both sets of the data sets are in use.

3.9.4 Resuming online reorganization

A suspended or failed reorganization may be restarted from the cursor position in the partition. The `INIT` or `/INIT` command is used, with any delete and rate options needing to be specified again on the resumption, as they are NOT retained from the previous execution. This is done with the **`OPTION(DEL|NODEL) SET(RATE(nn))`** options on the **`INIT`** or **`/INIT`** command.

OLR may be restarted on any IMS system in a data sharing environment.

3.9.5 IMS or z/OS failure

If an IMS system fails with OLR active, OLR is treated much like an in-flight application program. Any locks held for the OLR in a data sharing environment become retained locks, with conflicting lock requests from other systems for these resources being rejected. These lock rejects are likely to cause applications to abend. Requests for shared locks, such as reads of the data in the UOR, are not rejected. This means that updates within the UOR are not allowed. Of course, the records in the partition which are not in the in-flight UOR are available. The cursor remains active and is used to find which data set holds the requested database records. Ownership of the OLR remains with the failed IMS system, since OLR cannot be restarted on another IMS.

Emergency restart backs out the in-flight UOR and releases the locks. Releasing the locks allows other data sharing systems to have access to the database without experiencing lock rejects. Emergency restart also resumes the OLR.

3.9.6 FDBR with OLR

Fast Database Recovery (FDBR) tracks an active IMS system in a data sharing environment. If the IMS system fails, FDBR quickly backs out its in-flight work and releases its locks. This applies to OLR as well as application programs. Releasing the locks allows other data sharing systems to have access to the database without experiencing lock rejects.

FDBR does not release the ownership of the OLR by the failed IMS system. Ownership is recorded in the RECONs. FDBR does not access the RECONs. Since ownership is not released, OLR cannot be resumed on another IMS system in the IMSplex. When you restart the IMS system, the OLR is not automatically resumed, because at the end of /ERE restart, IMS will notify DBRC that the partition is no longer authorized. IBM is looking for the way to change this so that the OLR will be resumed after a successful /ERE.

3.10 Utilities

The following utilities have been enhanced for the OLR environment:

- Image copy
- Change accumulation
- Database recovery
- Batch backout
- HD Unload and HD Reload.

3.10.1 Image copy

The Image Copy, Image Copy 2, and Online Image Copy utilities may be used to create image copies of HALDB data sets. These utilities have been enhanced to copy the active HALDB data set. It does not matter whether the control statements specify an A-J data set DDNAME or an M-V data set DDNAME, as the utility uses the information in the RECON to determine which data set is active and copies it. Dynamic allocation should be used for image copies. The utilities will dynamically allocate the active data set. If a DD statement is supplied for an inactive data set, the utility will deallocate the inactive data set and allocate the active data set.

If you use GENJCL.IC, DBRC creates a control statement for the active data set. The skeletal JCL supplied by IMS will not create DD statements for database data sets - it uses dynamic allocation for these data sets.

There is a case where the use of static allocation will cause a failure. If the DD statement specifies an inactive data set which no longer exists, the job will fail due to a “data set not found” condition.

IMS does not allow an image copy to be processed for a partition while OLR is active for that partition (that is, a cursor is active). This remains true even if the OLR is temporarily suspended by the use of the **TERMINATE OLREORG** command.

A recovery of the output data sets from an online reorganization can be done without an image copy. The recovery is done by using only logs or logs and change accumulation data sets. Since the logs may contain many other records, it may be desirable to take image copies of the partition data sets soon after the completion of online reorganization. This will limit the number of logs required for a subsequent recovery. The image copies are likely to be concurrent image copies since this does not require the deallocation of the partition. A recovery using a concurrent image copy requires the input of the logs being written at the time of the concurrent image copy. This makes it desirable to switch logs between the completion of the OLR and the start of the concurrent image copy.

3.10.2 Change accumulation

The change accumulation utility (DFSUCUM0) has been enhanced to accumulate changes for both the A-J and M-V data sets. The control statements need only specify the A-J or the M-V DDNAMEs, as DBRC includes the partner data sets in the same CA group (for example, both the A and the M data sets).

The time of the start of OLR should be treated as purge time for output data sets, as this is equivalent to the image copy or reorganization time.

GENJCL.CA treats the start of OLR as the purge time for output data sets, and generates the DB0 statement with the correct purge time. This causes the utility to accumulate changes only since the beginning of OLR.

3.10.3 Database recovery

The IMS database recovery utility (DFSURDB0) recovers either the A-J or the M-V data sets, and the control statement *must* specify the correct DDNAME. DD statements *must* be included in the JCL.

Full recovery to the middle of an OLR is allowed, but time stamp recovery to the middle of OLR is not allowed. IMS Database Recovery Facility (DRF) tool may be used for this type of time stamp recovery.

Recovery of the output data sets after OLR starts but before an image copy is complete does not use image copy input. In this situation, the DFSUDUMP DD statement must specify DUMMY.

Timestamp recoveries cannot be done to a time in the middle of an OLR. This is true even if the time is one at which the OLR was suspended with a **TERMINATE** command. If a user attempts a time stamp recovery to a time between the start and stop times of an OLR, the utility detects this and will not execute. GENJCL.RECOV also detects this situation and will not generate recovery JCL.

If a database data set needs to be recovered due to a failure during an OLR, a full recovery is allowed. Of course, the partition must be /DBRed for the full recovery.

GENJCL.RECOV

The GENJCL.RECOV command can be used to generate the JCL to recover a database data set being used for OLR. If DDNAME is not specified, DBRC generates recovery JCL for active data sets. If the time is during OLR, both the A-J and M-V data sets are recovered.

If DDNAME is specified, GENJCL.RECOV fails if an inactive data set is specified.

The **USEOLRRG** keyword has been added for GENJCL.RECOV. When USEOLRRG is specified, the generated JCL will contain the logs created during the OLR, but no image copy.

IMS Database Recovery Facility (DRF)

IMS Database Recovery Facility (DRF) can recover partitions of OLR capable databases. It recovers the active data sets for the databases and partitions specified in any of the following commands:

```
/REC ADD DB dbname or partname
/REC ADD DBDS dbname or partname ddname
/REC START ...
```

DRF time stamp recovery is supported to a time during an OLR, and will recover the A-J and M-V data sets. DRF uses RECON information to decide which data sets, A-J or M-V, to recover. It recovers the data sets which were active at the time specified for the recovery. If the cursor was active at the time to which the recovery is being done, then both the input and output data sets are selected (since both sets are active), otherwise only the active set of data sets are recovered. This is true even when a data set DDNAME is specified with the command:

```
/RECOVER ADD  DBDS dbname ddname
```

If the ddname specified in the command was inactive during an OLR at the time to which we are recovering, its partner data set is recovered.

Full recovery using DRF

If OLR is complete and no post-reorganization image copy is available, DRF recovers the output data sets using the logs and/or the change accumulation data sets since the reorganization start time to recover the database.

If OLR is complete and a post-reorganization image copy is available, OLR recovers output data sets using the image copy plus any change accumulation data sets and the logs for the recovery.

If OLR is not complete, DRF recovers the input data sets from the most recent image copy, the change accumulation data sets and the logs. DRF also recovers the output data sets using the logs since the reorganization start time. Change accumulation data sets are not used, even if available.

Point-in-time recovery (PITR)

DRF can be used to recover a database to a point in time. DRF uses the same process as for a full recovery as detailed above, except that change accumulation data sets are not used.

3.10.4 Batch backout

The batch backout of an OLR in progress is generally not required. Any OLR abends invoke dynamic backout, and an IMS emergency restart also invokes dynamic backout.

Batch backout is required when IMS system fails and is then cold started. The backout must be run in a DLI (not DBB) region, as this is necessary in order to build an OLR PSB. IMS recognizes OLR by its PSB name, which is a numeric zero(0) followed by the partition name.

3.10.5 HD Unload and HD Reload

The HD Unload (DFSURGU0) and HD Reload (DFSURGL0) utilities may be used to complete a reorganization. They would only be needed when OLR cannot complete successfully, that is, OLR terminated with its cursor active.

The partition must first be /DBRed from all subsystems. The HD Unload utility reads both sets of data sets, and uses the cursor to determine from which to read. Dynamic allocation is used to allocate all the required data sets. The HD Reload utility always writes to A-J, X data sets, and dynamic allocation allocates only A-J, X data sets. The reload utility cleans up RECONs, setting the cursor to not active, and notifying DBRC that the A-J, X data sets are the active data sets.

3.11 DBRC changes for OLR

A number of changes have been made to DBRC for OLR. In this section, we discuss the extra fields that have been added to the RECON records and DBRC commands, as well as the coexistence of OLR with IMS Versions 7 and 8.

3.11.1 RECON records

The RECON records have been modified to include additional fields for OLR.

Master database record

The master database record has a flag indicating that this database is online reorganization capable. This flag can only be set for PHDAM and PHIDAM databases.

Partition database record

The partition database record has a number of new fields:

- Cursor active flag
- SSID of IMS owning OLR, if any
- Indicator for active data sets (A-J, M-V, or both sets)

  When both sets are active, this also indicates which are input and which are output.
- Flag for online reorganization capable

DBDS records for A-J and M-V data sets

The database data set records in the RECON have been extended, with the M-V records having many of the same attributes as the corresponding A-J data sets:

- GENMAX, RECOVPD, REUSE|NOREUSE values are the same
- They are members of the same CAGRP and DBDS groups.

Image copy cleanup processing treats corresponding data sets as a whole, and hence an Image copy of M may result in deletion of image copy record for A, and an image copy of M turns off the *image copy needed* flag for A.

REORG record

Recoveries cannot use inputs from both before and after a reorganization. REORG records are written to allow DBRC to create correct recovery JCL and verify recovery JCL. OLR makes the recovery JCL generation process more complex. Additional REORG records are needed and additional information is needed in the REORG records.

REORG records for both input and output data sets are created when OLR begins. If a partition has two data set groups, an OLR would create REORG records for the A, B, M, and N data sets. This information is used to prevent attempted recoveries of the input data sets to a time after the OLR. It is also used to allow recoveries of output data sets without image copies.

For offline reorganizations, the REORG record contains the time of the completion of reload or prefix update. This is the RUNTIME. For online reorganizations, both the start and stop times are recorded in the REORG record. The start time is the RUNTIME.

There is a flag in the REORG record for the database data set to indicate whether the online reorganization start time may be used for a recovery without an image copy. The flag is set to yes for output data sets and set to no for input data sets.

Change accumulation group

Partner data sets (A-J and M-V) must belong to the same CA group. If data sets from either is defined or added to a change accumulation group, the partner is also defined or added by DBRC.

DBDS group

Partner data sets must belong to the same DBDS group. If data sets from either is defined or added to a database data set group, the partner is also defined or added by DBRC.

3.11.2 DBRC commands

The following commands have been enhanced for OLR:

- INIT.DB
- INIT.PART
- CHANGE.DB
- CHANGE.DBDS
- NOTIFY.REORG
- NOTIFY.IC
- NOTIFY.UIC
- GENJCL.RECOV
- GENJCL.CA

INIT.DB

The INIT.DB command has had new parameters for master database:

```
OLRCAP | OLRNOCAP
```

These indicate whether master database is OLR capable or not.

INIT.PART

There are no new parameters on the INIT.PART command for OLR. DBRC automatically creates the M-V or Y DBDS records as necessary when the first OLR for the partition is performed. These records are deleted by a `CHANGE.DB... OLRNOCAP` command.

CHANGE.DB

With CHANGE.DB command, there is one new parameter for a master database:

`OLRCAP | OLRNOCAP`

This indicates whether the master database is OLR capable.

All partitions must be unauthorized before this command is executed unless MINVERS of 9.1 is set in the RECON. The A-J data sets must be the active set before OLRNOCAP can be specified.

There are some new parameters for HALDB database partitions. As these commands can compromise database integrity, these parameters should not be used for normal operations.

`OLRRGOFF | OLRRGON`

Indicates whether OLR is in progress.

`OLRIMSID(name) | NOOWNER`

Identifies owner of an OLR in progress.

`OLRDBDS(A | M)`

Identifies the active set of DBDSs for this partition.

CHANGE.DBDS command

The parameters to the CHANGE.DBDS command now apply either to both data sets, or can be applied separately.

The parameters of the CHANGE.DBDS commands that apply to both data sets are:

- ICJCL
- OICJCL
- RECOVJCL
- RECVJCL
- GENMAX
- RECOVPD
- REUSE
- NOREUSE

The parameters of the CHANGE.DBDS commands that apply to just one data set are:

- ADDEQE
- DELEQE
- ICON
- ICOFF
- RECOV
- NORECOV

NOTIFY.REORG

There are several new parameters for the NOTIFY.REORG command, but they should only be required with OLR in exceptional circumstances.

The new parameters are:

OFFLINE | ONLINE

Indicates whether the reorganization is offline or OLR. ONLINE requires that both A-J and M-V DBDS records exist.

STOPTIME(timestamp)

Specifies the stop time of an OLR. Required with ONLINE and only valid with ONLINE.

OLDSTOPT(timestamp)

Specifies previous stop time. Used when DRF PITR recovers partition to middle of a completed OLR.

RECOV | NORECOV

Indicates whether the data sets may be used as input for recovery. Only valid with ONLINE.

NOTIFY.IC and NOTIFY.UIC

These commands fail if the specified RUNTIME of the image copy falls within the time frame of an OLR.

GENJCL.RECOV

There is one new parameter for GENJCL.RECOV:

USEOLRRG

Indicates that active data sets are input to recovery (an image copy is not used as input).

GENJCL.IC and GENJCL.OIC

The GENJCL.IC and GENJCL.UIC commands do not generate DD statements for HALDB data sets, as dynamic allocation of the active data sets is done by the image copy utility.

GENJCL.CA

The GENJCL.CA command generates control statements for partner (A-J and M-V) data sets.

3.11.3 OLR coexistence with IMS Version 7 and IMS Version 8

There are some decisions that need to be made before a HALDB is made OLR capable in an environment where the database can be accessed by IMS Version 7 or IMS Version 8.

If HALDB is made online reorganization capable, then an IMS Version 7 system cannot access the database. Similarly, unless the required maintenance has been applied to an IMS Version 8 system, it will not be able to access a database that has been made online reorganization capable.

The IMS Version 8 OLR coexistence SPE consists of the following maintenance: PQ78493, PQ78758, PQ78916, and PQ78917. When these have all been successfully applied, an IMS Version 8 system can access databases that have been made online reorganization capable, and can participate in data sharing while OLR is active in an IMS Version 9 system. However, the IMS Version 8 systems cannot invoke OLR.

When the database is HALDB OLR capable, the database utilities (GENJCL processing, Database Image Copy, Database Image Copy 2, Database Recovery, Database Recovery and Change Accumulation) must be run using the IMS Version 9 utilities.

3.12 Data sharing, FDBR, RSR, and XRF

This section discusses the support for OLR in IMS Version 9 data sharing, Fast Database Recovery (FDBR), Remote Site Recovery (RSR), and Extended Recovery Facility (XRF) environments.

3.12.1 Data sharing support

IMS Version 9 supports the use of OLR in all data sharing environments, including batch data sharing and online data sharing.

The OLR owner system notifies sharers of:

- Cursor active (OLR started)
- Cursor updated (UOR completed)
- Cursor inactive
- OLR completed

The sharing systems process these notifies, and use the cursor to identify which data sets to use for a request. Normal call processing for applications programs will wait for locks if there is a conflict in accessing segments in a UOR.

3.12.2 FDBR support

Fast Database Recovery is supported in a system which uses OLR. When FDBR detects an IMS failure in a system with OLR active, it backs out in-flight UORs, releases data sharing locks, closes and unauthorizes the partition. When you restart the IMS system, IMS does not resume the online reorganization because the partitions are not authorized after the FDBR terminates.

3.12.3 XRF support

The IMS Extended Recovery Facility (XRF) supports IMS systems running OLR. The alternate system reads x'29xx' log records, in order to keep track of the status of OLR. At takeover the in-flight UOR is dynamically backed out, and OLR is automatically resumed.

3.12.4 RSR support

IMS Remote Site Recovery (RSR) supports IMS systems running OLR.

For DBRC RECON tracking, the log router routes some x'29xx' log records to the DBRC tracker. These are:

x'2910'	OLReorg ownership established
x'2940'	Cursor active
x'2970'	Cursor inactive
x'2990'	OLReorg ownership relinquished

For recovery level tracking, the x'29xx' and x'50' log records are written to the SLDS.

For database level tracking if the partition is online at the tracking site during OLR, when the x'2930' (output data set information) log record is received, the output data sets are used if they exist. Otherwise, they are created. When x'50' (database update) log records are received, the output data sets are updated. When the x'2970' log record (cursor inactive and OLR complete) is received, the input data sets are deleted if the **INITIATE** command included **OPTION(DEL)**.

For database level tracking if the partition is offline at the tracking site during OLR, log records are archived to the SLDS and the OLR status is updated in the RECONs.

If the partition was offline when the OLR was done, online forward recovery (OFR) is invoked when the partition is started. If OLR is in progress when the partition is started, OFR recovers A-J and M-V data sets. If OLR has completed when the partition is started and no image copy is available, the output data set is recovered from the OLR start time. If the OLR has completed when the partition is started and an image copy is available, the output data set is recovered using the image copy and subsequent log records.

RSR does not track ILDS or PHIDAM index updates. These data sets must be rebuilt using DFSPREC0 after takeover.

If there is an unplanned takeover initiated at the tracking site (**/RTAKEOVER UNPLAN** command), the tracking IMS completes its current work and shuts down. Any databases marked as recovery-level-tracking (RLT) must be recovered (or changed to database level tracking (DLT)) so that online forward recovery (OFR) recovers them.

The emergency restart (**/ERESTART**) at the RSR site does not automatically restart any OLR in progress at the time of the failure. This is different from a normal /ERE. The resumption must be done with an **INITIATE** command after DFSPREC0 has been used to rebuild the ILDS and primary index for PHIDAM.

3.13 OLR performance

The reorganization of HALDB databases with OLR can now be achieved without the need for any database outage. Online database reorganization is now no longer on the critical path, in contrast to the high priority critical task that usually categorizes an offline reorganization. OLR can be initiated at any time it is deemed necessary without the need to quiesce any system activity. This aspect of OLR is part of the design to provide the ultimate data availability solution. While reorganization of data is key to the performance health of a database, being able to reorganize online with full access to the data being reorganized brings with it some additional performance considerations.

There are a number of options which can influence the performance of OLR. These include:

- OSAM sequential buffering
- Logging
- Lock contention
- Database buffer pools
- Buffer contention
- DASD contention
- Pacing
- Address space priority
- Spreading the work across a sysplex

3.13.1 OSAM sequential buffering

OSAM sequential buffering can be used by OLR to reduce the elapsed time for reading and writing the records in a unit of reorganization. The *SBONLINE* parameter needs to be specified in the DFSVSMxx member of your IMS PROCLIB. The use of OSAM sequential buffering is recommended when OLR is being used for OSAM databases. It is also recommended that the DFSVSMxx member does not include the MAXSB keyword unless storage constraints exist.

3.13.2 Logging

The amount of data logged by OLR may impact online performance. All data is logged when segments are moved during an online reorganization. OLR has also introduced a few additional log records, which are described in “Logging” on page 43. Planning for additional log volumes is critical to the success of OLR. As an estimate, assume 10 cylinders of OLDS data will be created for every 1 cylinder of data moved during the online reorganization process.

Because the volume of log data can be overwhelming when considering the change accumulation process, it is recommended that OLR be run on one or more IMS systems which are dedicated to OLR processing. This will avoid excessive log processing for change accumulation for non-HALDB databases and partitions not using OLR.

When using multiple LPARs dedicated to OLR it is recommended that the LPAR have a minimum of 2 CPs (shared or dedicated) to allow more parallelism between IMS and IRLM.

Where OLDS logging becomes a performance bottleneck, it is recommended to use a 24K log block size, increase the number of log buffers to the maximum of 999, and use fast DASD in the most cache friendly configuration possible. The impact of OLR to the logging rate on the IMS system can be adjusted through the OLR rate. Reducing the OLR rate will have the side effect of lowering the OLDS logging rate.

When planning for running multiple concurrent OLRs consideration must be given to the impact of logging on the rest of the IMS system. It is recommended that no more than 4 OLRs (using rate=100) should execute concurrently in a single IMS system in which there is online activity to prevent excessive waits for logging due to high log latch contention.

3.13.3 Lock contention

OLR could create lock contention, but this is likely to be minimal. OLR limits the size of its UORs in an attempt to hold locks for no more than a second. That is, its algorithm for UOR size includes an adjustment to limit the times that locks are held. The data base record locks on PHDAM RAPs and PHIDAM roots are exclusive locks. Applications requesting locks on these resources will wait until they are released at the end of a UOR. Block locks are obtained on output data set OSAM blocks and VSAM CIs. Data sharing users could be affected by these locks. They prevent updates of the same block or CI. They do not prevent reads of other database records in the blocks and CIs.

Locking of database records by OLR hardly ever causes a deadlock. This is because OLR almost always uses conditional lock requests when already holding a lock. This means that if the requested lock is not available, OLR does not wait for it. Instead, it ends the UOR at the current location.

OLR requests block locks unconditionally. These lock requests could cause a deadlock in a data sharing environment. When OLR creates a deadlock, it is the victim. The other participants in the deadlock are not affected. UOR is dynamically backed out and automatically restarted.

3.13.4 Buffer pool definitions

The M-V (and Y) data sets are assigned to the same buffer pools as the corresponding A-J (and X) data sets. Only the A-J, L, and X identifiers may be specified in the DBD statements of DFSVSMxx members and DFSVSAMP data sets for HALDB data sets. The M-V (and Y) identifiers are not allowed on these DBD statements.

3.13.5 Buffer contention

OLR makes intensive use of buffer pools for the data sets it is reading and writing. The input and output data sets always use the same pool or subpool. Users should ensure that they have adequate buffers in these pools and subpools. OLR uses a VSAM string (PLH). IMS dependent regions and threads also use strings. A string is needed for each concurrent I/O in the pool. The number of strings for a pool defaults to the MAXREGN specification of system definition or the PST execution parameter. It may be overridden by specifying the STRINGNM parameter on the POOLID statement used to define the VSAM pool. Users should ensure that they have enough strings to support the maximum number of concurrent I/Os for a pool. If there are inadequate strings, I/Os must wait.

3.13.6 DASD contention

OLR is a highly I/O intensive process. DASD contention could occur. It is beneficial to use fast DASD with large cache where possible. It is also beneficial to use FICON® channels where high path utilization can contribute to I/O delays.

3.13.7 Pacing

The affect of OLR on the performance of systems and applications may be limited by using the RATE parameter on the INIT or UPD command to pace the OLR process. OLR does not hold any locks when the waits for this pacing are done, nor is the locked data held for a longer duration.

The RATE of the OLR controls the interval between UORs. While it is recommended to reorganize a database during periods of low activity for the database, when a database must be reorganized immediately, OLR can be initiated and the impact to the system and application tasks can be adjusted by increasing or decreasing the OLR rate until an acceptable balance is achieved.

3.13.8 TCBs used by OLR

OLR runs in the DL/I address space (DLISAS). This address space has a “parent” TCB with ten “child” TCBs, used for dynamic allocation. This was added in IMS Version 8. The parent TCB is used for HSM processing, and the child TCBs are used for database allocation, open, close and end of volume processing, as well as OLR. Each executing OLR will use one of these 10 TCBs, determined by the local DMB number using modulo 10 arithmetic. For example, if 3 partitions are being reorganized, 1 to 3 TCBs will be used.

3.13.9 Executing OLR in a sysplex

OLR may be run on any IMS in a data sharing sysplex. Some installations may decide to run online reorganizations on a dedicated system. The buffer pool definitions can then be tuned specifically for OLR, which would avoid buffer contention in this IMS. It has the added advantage of using a separate set of logs, reducing any potential logging contention.

Logs produced by a system dedicated to OLR will only have log records from databases using OLR. If a CA Group does not contain any of these databases, it will never need logs produced by this IMS system for change accumulations of other databases or for recoveries of other databases.

3.13.10 CPU use with OLR

OLR CPU requirements vary based on the OLR rate and partition size. A single OLR running at full speed (rate=100) can cause the LPAR (using 2 cps) CPU utilization to increase by as much as 18% on a zSeries processor for the duration of the OLR. The CPU utilization can be reduced by reducing the OLR rate. Reducing the OLR rate will also have the effects of reducing the logging rate and elongating the OLR elapsed time.

4

Fast Path enhancements in IMS Version 9

In this chapter we introduce IMS Fast Path enhancements in IMS Version 9. This chapter contains information relating to the following items:

- Multi-area structures for SVSO
- Area open and close enhancements
- Optional EMHQ structure for shared queues
- Serviceability and usability enhancements

4.1 Fast Path multi-area structures for SVSO

Previous releases of IMS required that each shared VSO area has its own structure. For availability reasons, duplexed structures were recommended. For users with many shared VSO areas, this could be a problem since z/OS limits the number of structures that may be defined. For z/OS 1.4 no more than 512 structures may be defined in a CFRM policy. Duplexing of structures exacerbates the problem.

IMS Version 9 now allows multiple areas to share a structure see Figure 4-1. This has two advantages. First, it may be used to avoid the restriction on the number of structures. Second, fewer structures simplify the management of structures.

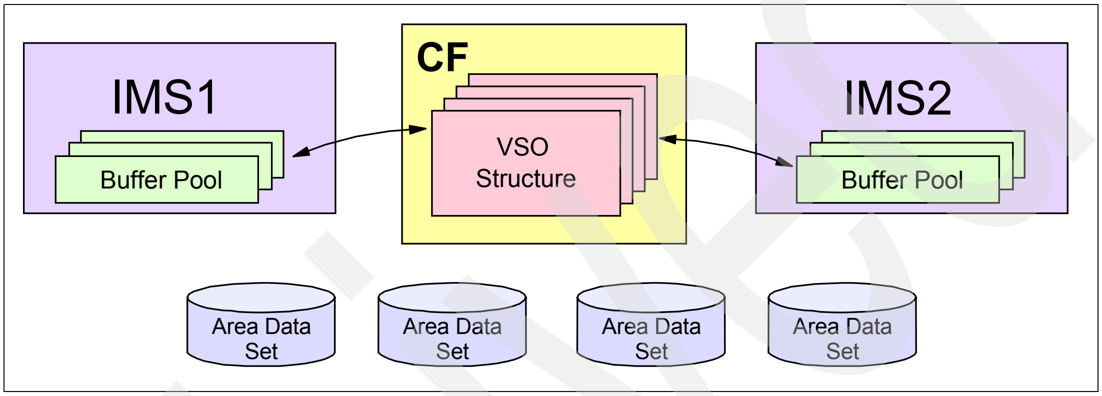


Figure 4-1 Multi-area structures for SVSO

4.1.1 Single area structures

With single area structures, each area has its own structure or pair of structures. Areas with the same CI size and the same buffer look aside option may share a private buffer pool. This is the pool that resides in the IMS control region. Single area structures remain an option in IMS Version 9.

Single area structures are defined to DBRC. The VSO keyword is for the use of VSO. CFSTR1 is used to define the primary structure used by the area. CFSTR2 is optional. It defines the duplex structure for the area. The LKASID or NOLKASID keyword is used to specify if look aside buffering is used for the area.

A private buffer pool may be defined for an area. It is specified in the DFSVSMxx member with a DEDB statement. A private buffer pool is devoted to an area or the areas of one DEDB. The definition includes the primary buffer allocation, the secondary buffer allocation, and the maximum number of buffers allowed for the pool as follows:

```
DEDB=(poolname,cisize,pri,sec,max,lkasid,DBorAreaName)
```

A private buffer pool may be defined without specifying an area name or DEDB name. This pool is used by areas which are not assigned to pool which specifies an area name or DEDB name. Such areas whose CI sizes and look-aside attributes match those specified for the pool use the pool as follows:

```
DEDB=(poolname,cisize,pri,sec,max,lkasid)
```

A default pool is built for the area if the appropriate DEDB statement is not present. The pool is shared by areas with the same CI size and look-aside attributes.

4.1.2 Multiple area structures

IMS Version 9 supports multiple area structures. These structures may be used for multiple shared VSO areas. All areas using a structure must have the same CI size and must share the same buffer pool. Accordingly all areas in the buffer pool use the same structure.

The use of a multiple area structure by an area is specified to DBRC. The MAS keyword on the INIT.DBDS or CHANGE.DBDS command of DBRC is used to specify that an area will use a multiple area structure. CFSTR1 keyword indicate that SVSO is used for the area and it specifies the structure name this area shares. There is no CFSTR2 keyword when MAS is specified. This means that IMS managed duplexing is not available for multiple area structures, and system managed duplexing is recommended instead.

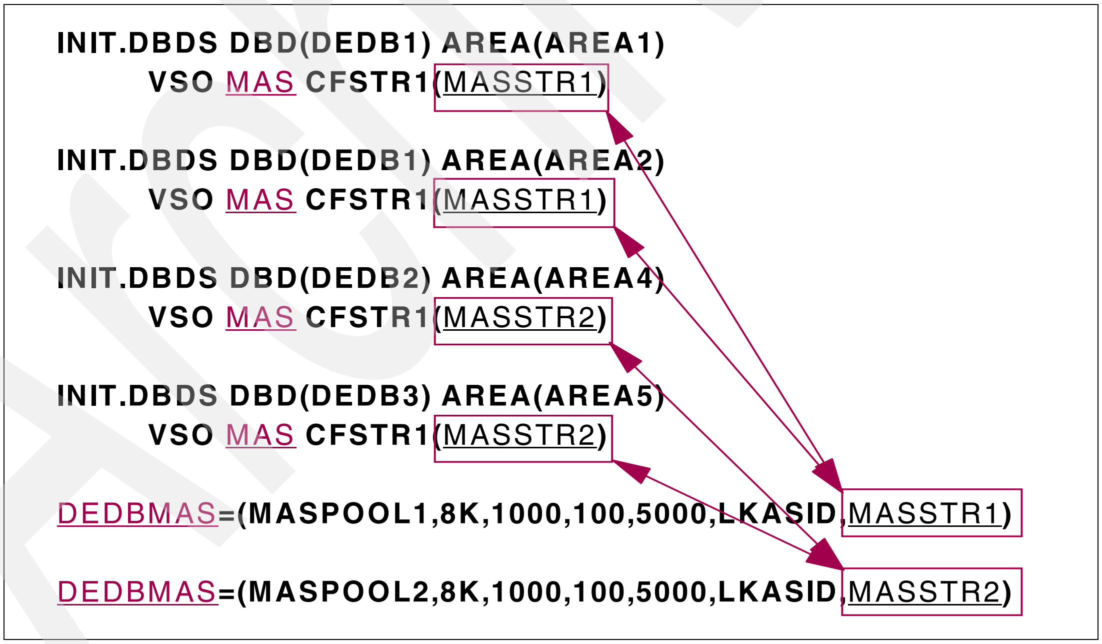
All areas using a structure must also use the same shared private buffer pool. The definition of the pool differs from that used for single area structures. The statement is the DEDBMAS statement. The DEDBMAS statement includes the name of the structure. It does not include the name of the areas or DEDBs using the pool. The default pool is built for the structure's areas if the appropriate DEDBMAS statement is not present. DEDBMAS is coded as follows in DFSVSMxx:

```
DEDBMAS=(poolname,cisize,pri,sec,max,lkasid,StructureName)
```

Example definitions

Figure 4-2 shows an example of definitions in DBRC for multiple area structures, and the correlation between the DBRC definitions and buffer pool definitions in DFSVSMxx as follows:

- AREA1 and AREA2 of database DEDB1 use structure MASSTR1.
- AREA4 of database DEDB2 and AREA5 of database DEDB3 use structure MASSTR2.



```
INIT.DBDS DBD(DEDB1) AREA(AREA1)
      VSO MAS CFSTR1(MASSTR1)

INIT.DBDS DBD(DEDB1) AREA(AREA2)
      VSO MAS CFSTR1(MASSTR1)

INIT.DBDS DBD(DEDB2) AREA(AREA4)
      VSO MAS CFSTR1(MASSTR2)

INIT.DBDS DBD(DEDB3) AREA(AREA5)
      VSO MAS CFSTR1(MASSTR2)

DEDBMAS=(MASPOOL1,8K,1000,100,5000,LKASID,MASSTR1)

DEDBMAS=(MASPOOL2,8K,1000,100,5000,LKASID,MASSTR2)
```

Figure 4-2 Multiple area definitions in DBRC and in DFSVSMxx

Example 4-1 on page 60 shows the CFRM policy definitions for three SVSO areas in our test environment. IMS duplexing is used for the structures and each area has its own individual pair of structures defined.

Example 4-1   CFRM policy for single area structures

```
/*----------------------------------------------------------------*/
/* IMS DATA SHARING - VSO - B-PLEX - SINGLE AREA STRUCTURES - IMS DUPLEXING */
*----------------------------------------------------------------*/

   STRUCTURE NAME(IMOB_AREAWH01A)  SIZE(4096)
       PREFLIST(CF06,CF03)

   STRUCTURE NAME(IMOB_AREAWH01B)  SIZE(4096)
       PREFLIST(CF03,CF06)

   STRUCTURE NAME(IMOB_AREAWH02A)  SIZE(4096)
       PREFLIST(CF06,CF03)

   STRUCTURE NAME(IMOB_AREAWH02B)  SIZE(4096)
       PREFLIST(CF03,CF06)

   STRUCTURE NAME(IMOB_AREAWH03A)  SIZE(4096)
       PREFLIST(CF06,CF03)

   STRUCTURE NAME(IMOB_AREAWH03B)  SIZE(4096)
       PREFLIST(CF03,CF06)
```

In our environment, we want to reduce the number of defined structures and with IMS Version 9, we can define all the areas to use one shared structure. As the DEDB SVSO structures now also support system managed duplexing, we can also define the DUPLEX(ENABLED) for the new structure. As mentioned earlier, IMS managed duplexing is not available for multiple area structures. Because this structure will be shared by all of the three areas used in the previous example, we increase the size accordingly. Example 4-2 shows the CFRM policy definitions for the shared multiple area structure which replace the single area structure definitions shown in Example 4-1.

Example 4-2   CFRM policy for the multiple area structure

```
STRUCTURE NAME(IMOB_AREAWHOM1)  SIZE(12288)
       DUPLEX(ENABLED)
       PREFLIST(CF03,CF06)
```

We also have two other DEDB SVSO databases in our test system, which we want to change to use the shared structure. Example 4-3 shows the single area structure definitions for these databases.

Example 4-3   CFRM policy for single area structures

```
/*----------------------------------------------------------------*/
/* IMS DATA SHARING - VSO - B-PLEX - SINGLE AREA STRUCTURES - IMS DUPLEXING */
*----------------------------------------------------------------*/

   STRUCTURE NAME(IMOB_AREADI01A)  SIZE(4096)
       PREFLIST(CF06,CF03)

   STRUCTURE NAME(IMOB_AREADI01B)  SIZE(4096)
       PREFLIST(CF03,CF06)

   STRUCTURE NAME(IMOB_AREADI02A)  SIZE(4096)
       PREFLIST(CF06,CF03)

   STRUCTURE NAME(IMOB_AREADI02B)  SIZE(4096)
       PREFLIST(CF03,CF06)
   STRUCTURE NAME(IM0B_AREAIT01A)  SIZE(8192)
       PREFLIST(CF06,CF03)

   STRUCTURE NAME(IM0B_AREAIT01B)  SIZE(8192)
       PREFLIST(CF03,CF06)
```

These areas have a different CI size compared to areas in the previous example, so we cannot change them to use the same shared structure. To convert them to system managed duplexing, we define DUPLEX(ENABLED) for this new structure. This structure will be shared by three areas in two different DEDB SVSO databases. We give the structure a generic name of IM0B_AREASMAS1. Example 4-4 shows the CFRM policy definitions for the shared multiple area structure which replace the single area structure definitions shown in Example 4-3 on page 60.

Example 4-4 CFRM policy for the multiple area structure

```
STRUCTURE NAME(IM0B_AREASMAS1)  SIZE(16384)
       DUPLEX(ENABLED)
       PREFLIST(CF03,CF06)
```

In IMS.PROCLIB, we define the shared private buffer pool for the newly defined structures by adding the following lines to the appropriate DFSVSMxx member:

```
DEDBMAS=(WAREMAS1,500,100,2000,LKASID,IM0B_AREAWHOM1)
DEDBMAS=(GENEMAS1,1000,100,5000,LKASID,IM0B_AREASMAS1)
```

So, now we have removed the old structures from the CFRM policy, defined two new structures, and defined the shared private buffer pools for these new structures. To finalize this change, we still have to do the following:

- Close and deallocate the areas from IMS(s)
- Run the IXCMIAPU utility for updating the CFRM
- Issue the command for activating the new policy
- Issue the DBRC commands in IMS as shown in Example 4-5
- Start the areas

Example 4-5 DBRC commands for changing SVSO areas to use multiple area structure

```
CHANGE.DBDS DBD(WAREDB) AREA(AREAWH01) -
            VSO MAS CFSTR1(IM0B_AREAWHOM1)
CHANGE.DBDS DBD(WAREDB) AREA(AREAWH02) -
            VSO MAS CFSTR1(IM0B_AREAWHOM1)
CHANGE.DBDS DBD(WAREDB) AREA(AREAWH03) -
            VSO MAS CFSTR1(IM0B_AREAWHOM1)
CHANGE.DBDS DBD(DISTDB) AREA(AREADI01) -
            VSO MAS CFSTR1(IM0B_ARESMAS1)
CHANGE.DBDS DBD(DISTDB) AREA(AREADI02) -
            VSO MAS CFSTR1(IM0B_AREASMAS1)
CHANGE.DBDS DBD(ITEMDB) AREA(AREAIT01) -
            VSO MAS CFSTR1(IM0B_AREASMAS1)
```

As a result of this exercise, we managed to reduce the number of structure definitions for these three DEDB SVSO databases with six areas from the original 12 structures to two by combining the areas having the same CI size to use a common structure, and by converting from using the IMS managed structure duplexing to using system managed structure duplexing.

4.1.3 Multiple area structures additional considerations

As you saw in the previous examples, all areas sharing a structure must have the same CI size, but the sharing areas may be from any DEDB. This is because all areas sharing a structure also share a buffer pool. There is no restriction on the number of areas that share a structure.

Duplexing of structures is recommended for shared VSO. If a non-duplexed structure is lost, the structure cannot be rebuilt. SVSO structures are *store-in* structures. They typically hold data which has not been written to DASD and which is no longer in buffer pools. If a non-duplexed structure is lost, its area or areas must be recovered. This makes the duplexing of structures critical for most installations.

The use of look aside buffering is determined by the specification on the DEDBMAS statement. LKASID or NOLKASID may be specified in the DBRC RECONs for areas using MAS, but it is ignored.

IMS does not determine if a multiple area structure is large enough to hold the areas that are assigned to it. This is different from single area structures with the PRELOAD option. The **/DIS FPV** command response is expanded in IMS Version 9 to include the count of CIs in the root addressable part as well as the number of CIs in the structure. This makes it easier to determine any requirement to increase the size of the structure for areas with the PRELOAD option.

4.1.4 System managed duplexing versus IMS managed duplexing

System managed duplexing has hardware and software requirements. There are no special requirements for multiple area structures. They have the same requirements as other structures using system managed duplexing.

The software requirement for system managed duplexing is z/OS 1.2 or higher; IMS Version 9 requires z/OS 1.4 or higher. The hardware requirements for system managed duplexing with IMS Version 9 are the same as with previous IMS releases:

- zSeries or G5/G6 servers
- Two or more Coupling Facilities
  - zSeries or 9672 G5/G6 with appropriate CFLEVEL
- CF-to-CF links

For more details on system managed duplexing refer to the technical white paper that can be found on the Web at the following address:

http://www.ibm.com/server/eserver/zSeries/library/techpapers/gm130103.html

IMS can manage duplexing for the DEDB VSO structures unless they are using the multiple area structure support introduced in IMS Version 9. The VSO structure uses the store-in model. As records gets updated, the data is asynchronously written to the structure, but not to DASD. At system checkpoints, updated data is then hardened to DASD. The issue then is that if a structure or Coupling Facility should fail, the data on DASD cannot be assumed to be valid and a database recovery process is needed. This requires restoring the last image copy and applying the logs, and could take a substantial amount of time. To assist with this, IMS has implemented support for IMS-managed software duplexing of DEDB VSO structures on an area by area basis since IMS Version 6. If one structure should become unavailable, the IMS managed duplexed structure is still available for reads and updates in simplex mode. To reinstate duplex mode, an IMS **/VUNLOAD** command is issued followed by a **/STA AREA** command. IMS does not stop data sharing during this process. The **/VUNLOAD** command copies the data to DASD. The **/STA AREA** command populates the two structures. All data is available during this time.

Compare this time-consuming IMS recovery procedure with system managed duplexing, where the duplexed structure may be automatically reestablished when the backup Coupling Facility comes online. There are no IMS commands required with system managed duplexing. Also, the VSO structure recovery will be consistent with other IMS database and system structures which currently support rebuild. Therefore there is no need for multiple operational recovery procedures. System managed duplexing provides the benefit of much easier system management characteristics compared with IMS managed software VSO duplexing. In addition, IMS managed duplexing is not available for multiple area structures.

4.1.5 Failures without successful duplexing

Normally, duplexing avoids failures associated with structures. This discusses what happens when duplexing is not used or when duplexing does not prevent the failure. The following types of failures can occur without duplexing:

- **Structure failure**

  If a structure fails, the data in the structure is not available. All of its areas are stopped and marked recovery needed.

- **Loss of connectivity to a structure**

  If there is a loss of connectivity to a structure, all of the areas in the structure are stopped on each system losing connectivity. Committed updates which are not yet written to the structure are written to DASD. Structure remains in a failed persistent status.

- **Loss of connectivity from all IMS systems**

  If connectivity is lost from all IMS systems, all of the areas in the structure are stopped. The structure remains in a failed persistent status.

- **A read or write error**

  A read or write error for a structure is treated as a loss of connectivity. If the duplexed structure is available, sharing continues. If not, the error is treated as a loss of connectivity.

Messages

This shows the IMS messages that are written with multiple area shared VSO structures.

- Connection when IMS connects to structure for first area using it in this IMS:

  ```
  DFS2822I AREA area CONNECT TO STR str SUCCESSFUL
  ```

- Association when other areas start using structure with existing connection:

  ```
  DFS2851I AREA area ASSOCIATE TO STR str SUCCESSFUL
  ```

- Disassociation when area is taken out of a multi-area structure:

  ```
  DFS2851I AREA area DISASSOCIATE FROM STR str SUCCESSFUL
  ```

- Disconnection when last area for an IMS is taken out of a multi-area structure:

  ```
  DFS2853I AREA area DISCONNECT TO STR str SUCCESSFUL
  ```

/DIS command

Example 4-6 on page 64 shows an example of the response to a **/DIS FPV** command. The ENTRIES column shows the number of CIs in the structure. The CHANGED column shows the number of changed CIs that have been written to the structure. They will be written to DASD by cast out processing which will reset the count. The AREA CI# column is new in IMS Version 9. It shows the number of CIs in the direct part of the area.

Example 4-6 /DIS FPVirtual

```
AREANAME STRUCTURE    ENTRIES  CHANGED  AREA CI#  POOLNAME  OPTIONS

DD01AR0  DD01AR0STR1  0000045  0000025  00000045  DD01      PREO, PREL
DB21AR8  MAS1KSTR     0000044  0000044  00000150  MAS1K     PREO
DB21AR5  MAS1KSTR     0000099  0000044  00000150  MAS1K     PREO, PREL
DB21AR6  MAS1KSTR     0000150  0000044  00000150  MAS1K     PREO, PREL
```

In the example, area DB21AR5 has 150 CIs in the direct part, but only 44 CIs from the area are in the structure. The OPTIONS column indicates that it is a preloaded area. This means that the structure is not large enough to hold the direct part of the areas using it. The size of the structure should be increased to hold all of the CIs for the AREAs using it.

4.2 Area open and close enhancements

IMS Version 9 opens and closes areas in parallel. Parallel processes are done with multiple TCBs. The parallel open is an asynchronous process that happens with online activity, so it does not delay IMS restart completion. IMS now restarts and reopens the areas after the reconnection to IRLM, and reopens the areas after /ERE completes. There are several options associated with these opens. The new options can improve the performance of restarts and simplify operations.

The FPOPN=D option has been added to IMS Version 8 by the APAR PQ74606. It disables the preopen process for DEDB areas. It applies to cold starts, warm starts, and emergency restarts. The open of areas with PREOPEN specified in the RECONs is like the open of other areas. It occurs either when they are first referenced by applications or as a result of a /START AREA command.

AREAs with the VSO PRELOAD option are affected by this process. They are not loaded until after they have been opened due to the first reference or as a result of a /START AREA command.

New parameters for IMS restart in IMS Version 9

The FPOPN=P option causes PREOPEN processing to be done outside of restart processing. This allows new work to begin more quickly. It applies to both normal and emergency restarts.

The FPOPN=R option causes /ERE processing to open all areas which were open at the time of the IMS abnormal termination. /ERE will not open areas that were closed. If an area with PREOPEN specified to DBRC was closed at the time of the IMS termination, /ERE will not open it.

The FPOPN=A option combines the P and R options. /ERE processing opens all areas that either were open at the time of the IMS abnormal termination or have PREOPEN specified to DBRC.

IMS (re)start option for opens of areas:

- FPOPN=N

  N (no), preopen executes in line as part of restart before online activity begins. This is like previous releases.

- FPOPN=P

P (preopen), initiate preopen at end of restart and executes asynchronously with online activity.

- FPOPN=R

  R (restore), for /ERE, reopen all areas that were open at termination. For `/NRE` command, treated as P option. Executes asynchronously with online activity.

- FPOPN=A

  A (all), for /ERE, combines P and R; for `/NRE` command, treated as P option

- FPOPN=D

  D (disable preopen) disables preopen

IMS restart processing

In previous releases, areas with PREOPEN specified to DBRC were opened as part of the restart process. The restart process did not complete until these opens completed. Online processing could not begin until restart completed. Figure 4-3 illustrates the processing time lines for IMS versions before IMS Version 9.

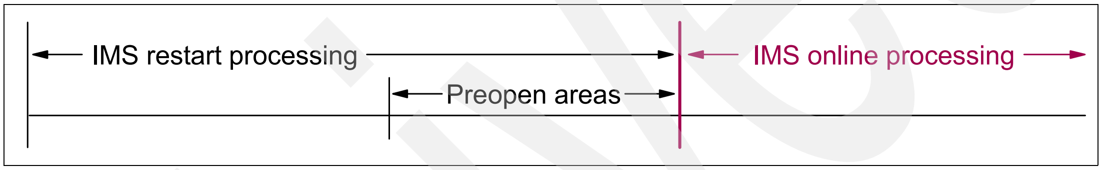


Figure 4-3 Pre-IMS Version 9

In IMS Version 9 with the FPOPN=N option, preopen processing remains part of restart processing, but it is shortened. These opens are done in parallel using 10 TCBs. This shortens the elapsed time of restart processing. Figure 4-4 shows the timing for an IMS Version 9 system that has FPOPN=N specified.

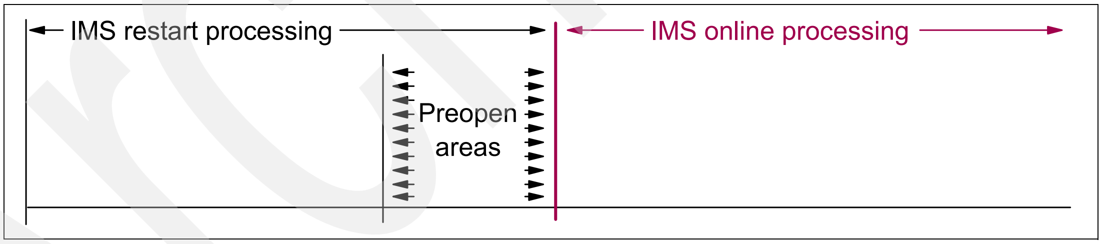


Figure 4-4 IMS Version 9 with FPOPN=N

In IMS Version 9 with FPOPN=P, R, or A, the open process is not part of restart. This allows online processing to begin sooner. Open processing is done in parallel. For normal starts, only areas with PREOPEN specified are opened. For emergency restarts, the areas opened depend on the option chosen. Figure 4-5 illustrates these options.

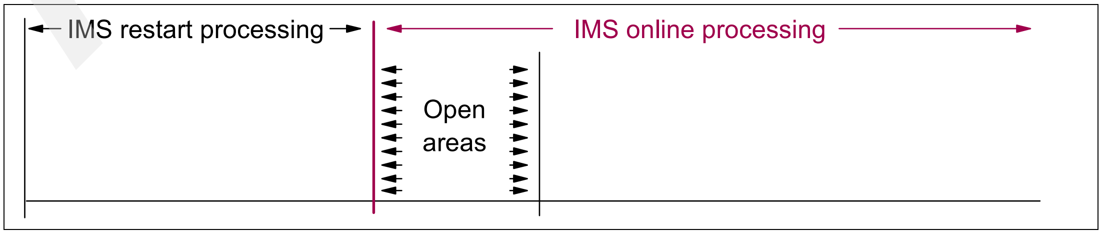


Figure 4-5 MS Version 9 with FPOPN=P, R, or A

IRLM reconnect options

When IRLM fails, IMS does not fail. Instead, it quiesces its work and waits to reconnect to IRLM. It also stops all of the DEDB areas. Previous releases of IMS did not restart these areas when IMS reconnected to IRLM. Operators had to issue the /START AREA commands. IMS Version 9 gives users the option to cause IMS to automatically restart the areas when the reconnect is done.

The new IRLM reconnect options are controlled by the FPRLM start up parameter. All actions that are taken, are executed asynchronously with online activity. You can specify FPRLM parameter as follows:

- FPRLM=N

  The N (no) option causes IMS to act as it did with previous releases. It does not start areas when the reconnection to IRLM is made.

- FPRLM=P

  The P (preopen) option applies to areas with the PREOPEN parameter. They are started and opened when IMS reconnects to IRLM regardless of their status prior to the IRLM abnormal termination.

- FPRLM=S

  The S (start) option causes IMS to start all of the areas which were stopped during the IRLM disconnect. The starts are done when IMS reconnects to the IRLM.

- FPRLM=R

  The R (restore) option starts and opens only those areas which were stopped as part of the IRLM disconnect process. It does not start areas which were already stopped, including those with PREOPEN specified to DBRC. Message DFS3210I is issued after all of these areas have been started.

New messages

There are new messages associated with the parallel open, start, close, and stop processes in IMS Version 9. Message DFS3715I is issued when a process starts, and it has the following format:

```
DFS3715I DEDB AREA xxxxxxxx PROCESS STARTED, RSN=rr
```

DFS3719I is issued when a process completes, and it has the following format:

```
DFS3719I AREA xxxxxxxx PROCESS COMPLETED, RSN=rr
```

The text of the message identifies the process, which can be one of the following:

PREOPEN The process used to open the areas with the PREOPEN parameter.

REOPEN The process used to open the areas that were open at termination or IRLM disconnect time.

RESTART The process used to start areas which were started at the time of the IRLM disconnect.

SHUTDOWN The process used to close areas in parallel when IMS is shut down.

RLM REVR The IRLM reverify process. It is used to close and stop areas in parallel for an IRLM disconnect.

Operations considerations

We suggest the following choices:

- FPOPN=R and FPRLM=R

With these values defined you can gain the following benefits:

- Emergency restart restores system to state at time of IMS failure
- Restart completes earlier, new work begins earlier
- Reconnect to IRLM restores system to state at time of IRLM failure

Automated operations may be adjusted. Users who currently have automation to reopen and restart areas after a restart or reconnect can eliminate this automation.

4.3 Optional EMHQ structure for shared queues

In previous IMS releases there was a need to define EMHQ structures even though the environment did not use shared expedited message handler (EMH) messages when Fast Path was included in the system. IMS Version 9 does not require that the EMHQ structures be defined when not using shared EMH. These actions are required if you do not require the EMHQ structure:

- **IMS**

  The EMHQ= parameter in the IMS DFSSQxxx PROCLIB member specifies the primary structure for the EMH queues. It is deleted when no structure is required. This disables shared EMH processing in IMS.

- **CQS**

  Each CQS member has a local PROCLIB member, CQSSLxxx. There is a global PROCLIB member, CQSSGxxx, which is shared by all of the members. The STRUCTURE statements for the EMH structure may be deleted when no structure is required. When STRUCTURE statements for shared EMH structure in CQSSLxxx and CQSSGxxx PROCLIB members are not present, CQS does not allocate EMHQ resources any longer in the following restart.

- **CFRM policy**

  You may also delete the STRUCTURE statements for the EMHQ structure and its logger structure from the CFRM policy. This is not required, because it is permissible to have definitions for structures which will not be built. Nevertheless, you will want to clean up the policy by deleting these definitions, so the recommendation is to delete STRUCTURE statement for EMHQ and its CQS log from policy.

- **LOGR policy**

  You may delete the DEFINE LOGSTREAM statement for the EMHQ log from the LOGR policy. This is not required because it is permissible to have definitions for log streams which will not be created. Nevertheless, you will want to clean up the policy by deleting this definition, so the recommendation also here is to delete DEFINE LOGSTREAM statement for EMHQ log.

- **Automation**

  If there is any automation which issues commands for the deleted structures, it should be modified.

- **IMS restart**

  To change from or to the usage of EMHQ, you must use a cold start of IMS. The cold start may be either a `/NRE CHECKPOINT 0` or a `/ERE COLDCOMM`. EMHQ usage cannot be changed across a warm or emergency restart, so you cannot use `/ERE COLDSYS`, an `/NRE` warm start command, or any other form of `/ERE` command.

4.4 Fast Path serviceability and usability enhancements

There are several small enhancements for serviceability, usability, and documentation including the following:

- Expanded log records
- Fast Path log records in ILOGREC
- Expanded /DIS commands
- Data space elimination
- Trace table
- Utility enhancements
- Fewer Fast Path snap dumps

Log records

In IMS Version 9, the following enhancements have been made to Fast Path log records:

- When a dependent region abends, the Fast Path dependent region blocks, EPST and EMHB, are written in the x’6706’ log records. However, the x’6706’ log records are not written for U0777 and U3303 abends.
- The Fast Path dependent region buffer values, NBA and OBA, are included in the Fast Path sync point log record (x'5937'). These are the normal and overflow buffer allocations for the dependent region. The number of buffers actually used is also in the log record. It was included in previous releases. The number of SDEP segments inserted are included in the log record. The IMS Performance Analyzer product reports these new fields.

Fast Path log records in ILOGREC macro

Fast Path records are now included in ILOGREC macro. You can use the following specifications to map the Fast Path log records:

- ILOGREC RECID=ALL

  Includes all Fast Path log records
- ILOGREC RECID=59nn

  Creates specific Fast Path log record

Old Fast Path log record macros, such as DBFLGSYN and DBFLSRT, are still available.

Commands

The response to the **/DIS AREA** command has been expanded to include the identification of the IMSID where IMS Fast Path utility is active. The utilities include:

- High Speed Sequential Processing utility (HSSP)
- DBFUHDR0, High Speed Reorganization utility (HSRE)
- DBFUMDL0, SDEP Delete utility (MDL0)
- DBFUMSC0, SDEP Scan utility (MSC0)
- DBFUMRI0, DEDB Create utility (MRI0)
- DBFUMMH0, DEDB Compare utility (MMH0)
- VSO Preload is active (PRLD)
- An IBM DBTOOL Online utility (TOOL)

Example 4-7 shows a response to /DIS AREA command issued in IMS3 system, and you can see that there is an HSSP utility running in IMS2.

Example 4-7 /DIS AREA

```
/DIS AREA DB21AR0
```

```
DFS000I  AREANAME  EQECT   TOTAL UNUSED  TOTAL UNUSED   DBNAME   EEQECT  CONDITIONS      IMS3
DFS000I   DDNAME   REMAIN SEQ DEPENDENT DIR ADDRESSABLE                                   IMS3
DFS000I  DB21AR0   N/A     29    28      74    0-UW     DEDBJN21 0       VIR, PREO, PREL IMS3
DFS000I   UTILITY:  HSSP  PVTTOT#   45 PVTAVL#    30     IMSID: IMS2                      IMS3
DFS000I   DB21AR01    10     N/A    N/A    N/A    N/A     N/A                             IMS3
```

When the /START DB command is issued at an RSR tracking site, online forward recovery (OFR) is activated without requiring /START AREA commands for each area. These /START AREA commands were required with previous IMS releases.

VSO data space

IMS Version 9 does not create any VSO data spaces unless VSO is used. The first data space is created when the first open of a VSO area is done. Some installations limit the number of data spaces that may be created. The change will potentially benefit such installations when they do not use VSO.

Fast Path trace table

In IMS Version 9 there is a new trace table for Fast Path. Fast Path trace in previous releases is actually a dependent region trace, and it does not use the IMS internal trace function. New trace has similar format to DL/I trace table. These traces are documented in *IMS Version 9: Diagnosis Guide and Reference,* LY37-3203. Fast Path trace is turned on by the command shown in Figure 4-6. The IPCS dump formatter is changed to reflect this new trace entry as well.

```
                                     NOLOG            MEDIUM
/TRACE SET ON TABLE FPTT OPTION              VOLUME
                                     LOG              LOW
                                                      HIGH
```

Figure 4-6 Fast Path trace command

High speed reorg utility

In previous releases high speed reorg utility would stop after five failures due to insufficient buffers for IOVF. This is the default for IMS Version 9, but the new STOPAFTERFAIL#= specification may be used to change the number of failures required for the stopping of the reorg. Example 4-8 shows an example how to specify this new control statement.

Example 4-8 Example control statements

```
TYPE HSR
STOPAFTERFAIL#=10
AREA DB21AR1
GO
```

If STOPAFTERFAIL# is not specified, the default of 5 is taken. If STOPAFTERFAIL#=0 is specified, the utility will continue to run until it reaches the end of the DEDB area. The maximum value for STOPAFTERFAIL# is 999. Example 4-9 shows the messages when the STOPAFTERFAIL#=1 is specified and reached.

Example 4-9 Stopafterfail

```
TYPE REORG                                                                   TEM00100
 STOPAFTERFAIL#=1
 AREA DB21AR3
 GO                                                                           TEM00150
 AREA DB21AR3 HAS       5 UOW'S.
    EACH UOW HAS   15 CI'S;    5 AP CI'S,   10 DOVF CI'S.
    EACH CI IS   2K, ONE BUFFER SET REQUIRES      30K OF STORAGE
    INDEPENDENT OVERFLOW HAS       1 OVERFLOW UNITS, A TOTAL OF      74 DATA CI'
 STATS FOR REORG OF AREA DB21AR3 :
    # OF UOWS REQUESTED TO REORG=       5; LOW UOW       0, HIGH UOW       4
    # OF UOWS ACTUALLY REORG'D=       2; LOW UOW       0, HIGH UOW       4
       # OF UOWS SKIPPED=       2, ALL ANCHOR POINT CI'S WERE EMPTY
       # OF UOWS FAILED=       1, MAX OF 999 FOLLOWS:
         UOW       1 FAILED; REASON: 02, INSUFFICIENT BUFFERS FOR IOVF I/O
    # OF IOVF CI'S FREED BY REORG:       0 -       0 (REUSED) =       0
    # OF IOVF CI'S ALLOCATED:       0 (NEW) +       0 (REUSED) =       0
    # OF FREE IOVF CI'S AFTER REORG=      74, PERCENT FREE= 100
    PRIVATE BUFFER SET INITIAL ALLOCATION=   3, EXTENSION COUNT=   0
       RAP INPUT BUFFER SETS=   1, OUTPUT/IOVF BUFFER SETS=   2
       UOW       0 USED    0 BUFFERS FOR IOVF I/O, THE HIGHEST USAGE
    PERFORMANCE STATS: # OF ASYNCHRONOUS READ AHEAD I/O'S=       0,
                       # OF WAITS FOR UOW LOCKS=       0,
           # OF WAITS FOR PRIVATE BUFFERS=   0
DFS2662A UTILITY NOT EXECUTED AS REQUESTED
```

SDEP utilities

For enhancing the diagnostics, some messages written by SDEP utility processing have been modified to add the area name as follows:

```
DFS2634I AREA=xxxxxxxx HWM CI OWNED, SDEP HARDENED LOCALLY
```

This message is written when SDEP SCAN/DELETE runs and a sharer needs to harden update buffers. When SDEP SCAN/DELETE has QUITCI in SYSIN, but the sharer does not own HWM (high water mark) CI and RBA of HWM CI is greater than the one of current CI, this message is written on the sharer.

```
DFS2635I AREA=xxxxxxxx HWM CI HARDENED REMOTELY
```

This message is written when the SCAN or DELETE utility runs on non-HWM-Owned partner(s).

```
DFS2636I AREA=xxxxxxxx PARTNER QCI FREE LOCKS
```

This message is written when I/O request of Fast Path utility detects CI locks on pre allocated CIs.

SDEP CI format

IMS Version 9 does not support the IMS Version 5 DEDB CI format. Migration to the format introduced in IMS Version 6 must be done before the migration to IMS Version 9. It is highly unlikely that this will affect any installations. If an SDEP CI has been written to an area by IMS Version 6, 7, or 8, the CI format has been migrated to the *new* format.

Fast Path abend reduction

Prior to IMS Version 9, some Fast Path modules issue user abends even though the situation is not critical, for example, when a /DIS command finds a problem in its processing and causes termination of the control region.

For reducing abends, U3275, U2484, U0732 and U3999 abends are eliminated where appropriate. Instead of these user abends, a few messages are changed and a few are added. The changed messages are:

```
DSP0175I UNAUTHORIZATION PROCESSING WAS UNSUCCESSFUL
DSP0230I UNAUTHORIZATION PROCESSING WAS UNSUCCESSFUL
```

The solution or new messages are:

```
DFS0015I mmmm nn FAILED TO GET CONTROL BLOCK cccc
DFS0019I DBRC UNAUTH REQUEST FAILED FOR DBD=dbdname, AREA=areaname RC=xx
DFS0020I NO DDIR WAS FOUND, /DIS STATUS DB COMMAND REJECTED
DFS0020I NO DMCB WAS FOUND NO FP DB IS AVAILABLE, /DIS STATUS DB COMMAND REJECTED
DFS0020I DMCB HAS BAD POINTER TO DDIR, /DIS STATUS DB COMMAND REJECTED
DFS0020I DMAC HAS BAD POINTER TO DMCB, /DIS STATUS DB COMMAND REJECTED
```

Refer to *IMS Version 9: Messages and Codes Volume 2,* GC18-7828 for the explanation of these messages.

Fast Path abend reduction is retrofitted to IMS Version 8 by APAR PQ73630 and to IMS Version 7 by APAR PQ73631.

5

DBRC enhancements

This chapter describes these topics:

- The new application programming interface (API) for DBRC
- The functional enhancements to DBRC
- RECON migration and coexistence.

5.1 DBRC application programming interface

Many IMS customers have created their own methods for accessing the information stored in the RECON. In the past, many organizations have parsed the output of **LIST.RECON** commands, or written their own programs that read the RECON data set directly. The problem with this approach is the requirement to modify these programs for each new release of IMS and sometimes even for the same release after the maintenance has been applied.

There is now a supported method for retrieving data from the RECON. IMS Version 9 provides an application programming interface (API) for DBRC, which allows users to write programs which read data from the RECON. These programs can access any of the records that are in the RECON. Programs using this API are now the supported and preferred method for retrieving data from the RECON. A sample program using the API is provided, in the IMS.ADFSSMP library, called DSPAPSMP.

With the DBRC API, it will not be necessary to modify the programs which read the RECON through the API. Hence users do not need to understand any of the internal formats of the information in the RECON, such as the segmentation of logical records into multiple physical records, or the format of the records.

The API is provided using the assembler language macro interface. The functions provided include:

- Start the API environment, which allocates and opens the RECONs.
- Query the RECONs, to retrieve information from the RECON into an area of memory which is allocated by the API
- Release buffer storage, after the program has finished using the information previously retrieved.
- Stop the API environment, which will deallocate the RECONs.

The DBRC API environment supports access to RECONs in IMS Version 9 or later format. Programs can access information from RECONs being used for IMS Version 7 and Version 8 systems only after the RECONs being used for these systems have been upgraded to IMS Version 9 format.

The assembler macros are in the IMS.ADFSMAC and IMS.SDFSMAC libraries, and invoke modules in SDFSRESL. The API allocates RECONs in the same manner as the utilities, with enqueues and reserves. This prevents programs for accessing information that has only been partially written by DBRC, although it also adds an extra potential lockout if the application program reads large amounts of information from the RECON.

Programs using the DBRC API execute in an application program address space, using AMODE 31/RMODE ANY. Cross memory mode is not supported. IMSplex is supported, with the PLEX name supplied in the JCL or provided by the SCI registration exit routine.

The DBRC API is documented in *IMS Version 9: Database Recovery Control (DBRC) Guide and Reference,* SC18-7818.

5.1.1 Application overview

The functions provided by the DBRC API macros are:

- DSECT
- STARTDBRC
- QUERY,TYPE=xxxxxxxx,...
- RELBUF

- STOPDBRC

DSPAPI FUNC=DSECT

This macro includes into your program the required DSECTs and equates used by the API. It is required by any program using the API. The DSECTs included into your program by this macro are described in the Appendix A, “DBRC API control blocks” on page 233.

DSPAPI FUNC=STARTDBRC

This macro provides several functions. Firstly, it returns the API TOKEN, which must be stored by the program and provided on subsequent macro calls. It is used to identify this program to DBRC. The macro then opens the RECONs. The RECONs are allocated using MDA if they have not been previously allocated by the user. Alternatively, you may allocate RECONs with DD statements or by dynamic allocation (SVC 99). Finally, the macro opens the output data set (SYSPRINT), which is used for any API messages.

The default DD name for the output data set is SYSPRINT, although this may be overridden if desired.

DSPAPI FUNC=QUERY

This macro specifies the type of information that you want to access from the RECON, through the use of the TYPE parameter. The valid values for TYPE are:

- RECON

  Information about the status of the RECON is returned to your program.
- DB

  Information about one or many databases is retrieved.
- xxxGROUP

  The API supports the retrieval of information from several different types of groups. These are the change accumulation group (CAGROUP), database data set groups (DBDSGROUP), database groups (DBGROUP), recovery group (RECOVGROUP), and global service group (GSGROUP).
- LOG

  The recovery log and system log data set (RLDS and SLDS) information is returned.
- OLDS

  The online log data set (OLDS) information is returned.
- SUBSYS

  Subsystem information for a specific subsystem or for all subsystems is retrieved.
- BACKOUT

  Backout information is returned to your program, either for one subsystem or for all subsystems.

The macro will also acquire storage to hold the information provided to your program, and returns the address of the storage to the program. The format of the information is documented by the DSPAPQxx macros, which are described in Appendix A, “DBRC API control blocks” on page 233.

DSPAPI FUNC=RELBUF

This macro releases the storage acquired by IMS to hold the information returned in previous QUERY calls. This call is required, otherwise the storage allocated by previous calls by this program will not be released.

DSPAPI FUNC=STOPDBRC

This macro closes the RECONs, deallocates them if STARTDBRC allocated them, and closes SYSPRINT. It does not release the storage holding QUERY information.

Examples of database QUERY calls

The following are examples of DSPAPI FUNC=QUERY calls:

- To retrieve the information from the first database record in the RECON, the following call would be used:

  ```
  DSPAPI FUNC=QUERY,TYPE=DB,LOC=FIRST,...
  ```

  The LOC=FIRST parameter returns first database in the RECON.
- To retrieve the information for a specific database:

  ```
  DSPAPI FUNC=QUERY,TYPE=DB,DBNAME=ACCDB,...
  ```
- To retrieve the information about the database data set, and allocation and image copy information for database ACCDB:

  ```
  DSPAPI FUNC=QUERY,TYPE=DB,DBNAME=ACCDB,
     DDN=ACCDS1,LIST=(ALLOC,IC),...
  ```
- To retrieve the information from the next database after ACCDB:

  ```
  DSPAPI FUNC=QUERY,TYPE=DB,DBNAME=ACCDB,
     LOC=NEXT,...
  ```
- To retrieve the information for a list of databases:

  ```
  DSPAPI FUNC=QUERY,TYPE=DB,DBLIST=MYLIST,...
  ```

The data requested through FUNC=QUERY macro is returned to the program through linked blocks of storage.

The format of each of the different types of blocks that can be returned through the DBRC API is described in the *IMS Version 9: Database Recovery Control (DBRC) Guide and Reference,* SC18-7818. They can be included into your program through the DSPAPI FUNC=DSECT macro. There is a header to each block that is described by the DSPAPQHD macro, and the data portions of the blocks is described by the DSPAPQxx macros:

DSPAPQAR	Fast Path Area block
DSPAPQBO	Backout block
DSPAPQCA	CA Execution block
DSPAPQCG	CA Group block
DSPAPQDB	Full Function DB block
DSPAPQDG	DBDS, Database, and Recovery Group block
DSPAPQDS	Database Data Set DBDS) block
DSPAPQEL	EEQE List
DSPAPQFD	Fast Path DEDB block
DSPAPQGG	Global Service Group block
DSPAPQHB	HALDB block

DSPAPQHD	DBRC API query output header
DSPAPQHP	HALDB Partition block
DSPAPQIC	Image Copy block
DSPAPQLA	Log Allocation block
DSPAPQLG	RLDS/SLDS block
DSPAPQLI	Log Information block
DSPAPQNF	Not Found block
DSPAPQOL	Online Log Data Set block
DSPAPQRC	RECON Status block
DSPAPQRI	DBDS/Area Recovery Information block
DSPAPQRR	DBDS Reorganization block
DSPAPQRV	DBDS Recovery block
DSPAPQSL	DB/Area Authorized Subsystem List block
DSPAPQSS	Subsystem block

The general structure of each of these blocks is shown in Figure 5-1.

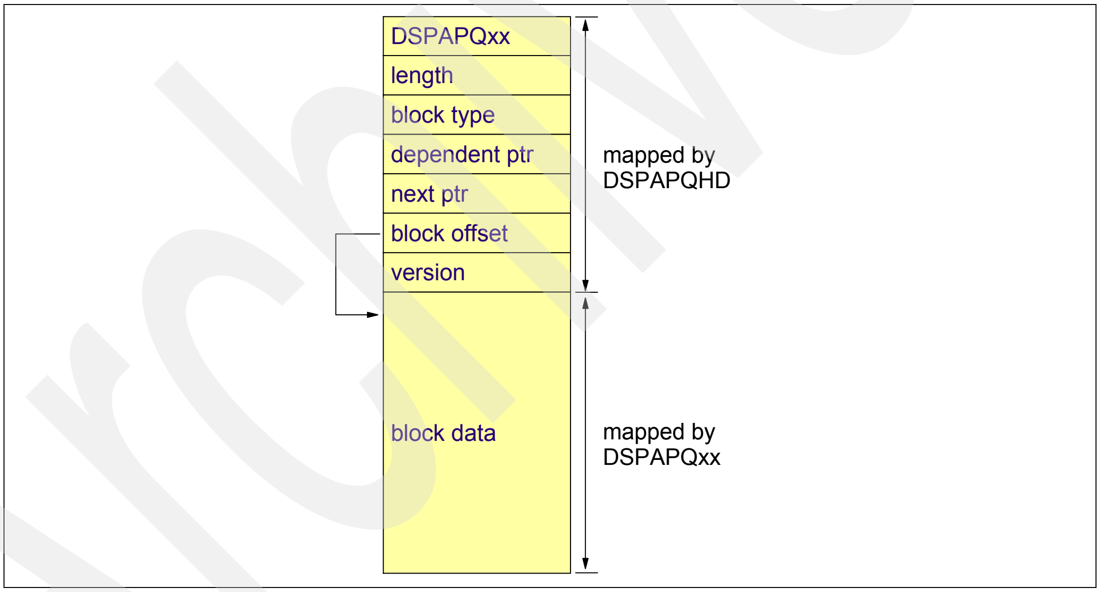


Figure 5-1 Structure of the blocks returned by the DBRC API

The information returned by the API contains two types of information, namely data and pointers. The information is chained, with pointers between the blocks, and the blocks are chained in a hierarchy.

For example, the blocks returned for a full function (non-HALDB) database are shown in Figure 5-2 on page 78. The first piece of information returned is the database block. The database block points to the block for the next database in the list. The database block also points to the first DBDS block for each database. The DBDS block points to the subsequent DBDS block for the database, as well as to recovery information for the DBDS. The recovery information block points to the allocation, image copy, recovery, and reorganization blocks. Each of these blocks (allocation, image copy, recovery, and reorganization blocks) points to the next block of the same type (allocation to the next allocation, IC to the next IC, and so on). IMS will return only the blocks of information requested, in the macro call, to your program.

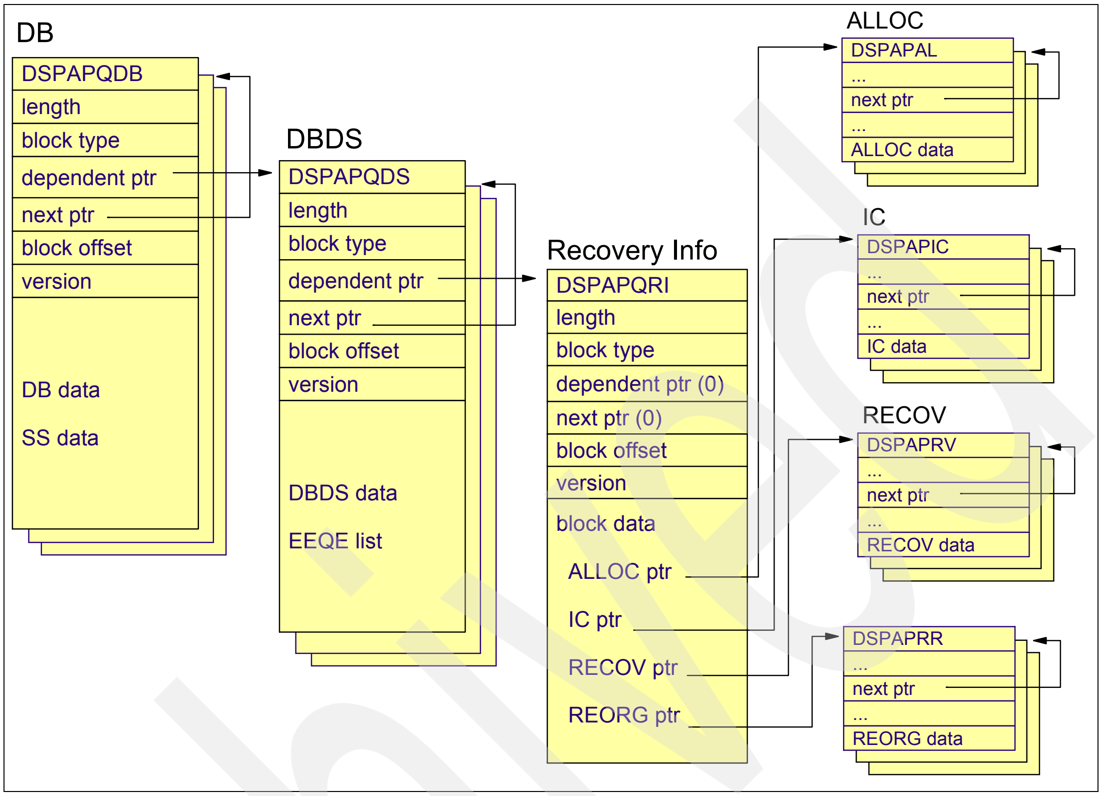


Figure 5-2 Blocks returned from a DBRC API call for a full function (non-HALDB) database

The version parameter

Programs using the DBRC API can be written to be independent of the release of IMS. The API macros have a version parameter, which is specified by the user. This parameter allows programs to remain the same when new data is added by an IMS release.

The version determines the parameters which may be specified, as well as the information returned and the format of information returned. New data is only returned to a program when a new version is specified. A version may change with IMS releases, although the version for function or type will be changed only when parameters or returned data is changed.

Version for IMS Version 9 is '1.0', which is the default.

Version numbers are not tied to a release. Each function has its own version number, and each TYPE for the QUERY function has its own version number. If a future release does not change the format of the information returned for a function or type, the version number for the function or type will not change.

Sample program overview

The structure for a typical program using the DBRC API would be as follows:

- Include the API DSECTs and supply working storage, using these macros:

  ```
  DSPAPI FUNC=DSECT
  DSPAPQxx macros
  ```

  The program uses the DSPAPI FUNC=DSECT to include the required API DSECTs and equate statements which the program will use with later DSPAPI macros. The program includes the DSPAPQxx macros to generate the DSECTs which map the blocks which it will request. The DSPAPQHD block is always used, so this macro is always needed. Other mapping macros, such as DSPAPQDB for databases, are included.

- Initialize the API, which will open the RECONs and receive the API token

  ```
  DSPAPI FUNC=STARTDBRC
  ```

- Issue one or more QUERY requests

  ```
  DSPAPI FUNC=QUERY
  ```

  QUERY functions are used to retrieve information from the RECONs. These create blocks. The blocks are processed by the program to accomplish its requirements. When the blocks from a previous QUERY call are no longer needed, their storage is released with a RELBUF request.

- Process the information returned from the QUERY requests
- Release buffer storage returned for each QUERY request

  ```
  DSPAPI FUNC=RELBUF
  ```

- Terminate the DBRC API, which will close the RECONs

  ```
  DSPAPI FUNC=STOPDBRC
  ```

DBRC API token

The DBRC API uses a token to distinguish each of the programs using the API which are running concurrently. The token is a four-byte field, returned by the DSPAPI FUNC=STARTDBRC call. The token must be supplied each time the program uses one of the DSPAPI macros to request information. When the program has finished accessing the RECON, and has processed the DSPAPI FUNC=STOPDBRC macro call, the API token is no longer valid and cannot be used again.

Storage used by programs

IMS automatically acquires storage during the processing of queries to the DBRC API. This storage is used to hold the blocks of information passed back to the application program. The address of the first block is passed to the application upon return from the macro.

DBRC never releases this storage unless specifically requested to do so. This is performed by the DSPAPI FUNC=RELBUF BUFFER=addr call. The BUFFER parameter points to first block in the chain to be released, and will release all storage connected to the block at that address.

The DSPAPI FUNC=STOPDBRC call, which is used to terminate the connection between the application and DBRC, does not release storage.

5.2 DBRC functional enhancements

There are a number of enhancements that have been added to IMS for Version 9. These include:

- Command authorization security for /RMxxxx commands
- Allow more than 32K database registrations
- GENJCL.IC enhancement for HALDB

5.2.1 Command authorization for /RMxxxx

IMS Version 8 added command authorization for DBRC commands, support for the DBRC utility (DSPURX00) and the Partition Definition Utility (PDU). It did not include authorization checking for the /RMxxxx commands.

IMS Version 9 support

DBRC command authorization security has been enhanced to include checking for the /RMxxxx commands through RACF (or an equivalent security product). This enhancement extends consistent security to cover online DBRC commands. The optional DBRC command authorization exit routine (DSPDCAX0) can also be used to verify that a user is authorized to issue a particular command.

The authorization support for /RMxxxx commands does not apply to commands issued from the z/OS console (MCS/E-MCS consoles and WTOR replies). They are always authorized for these DBRC commands. For other terminals, the user ID is used to evaluate authorization when it is available. The user ID is available when a user is signed on. If the user ID is not available, the LTERM name is used. For application programs using AOI, the user ID is the user ID assigned to the dependent region where the program is executed. There is an exception to the use of user ID. If a transaction whose TRANSACT macro specifies AOI=TRAN issues a /RM command by using AOI, the transaction code, not the user ID, is used to evaluate authorization.

DBRC command authorization invocation

DBRC command authorization is activated for batch and online commands by:

```
CHANGE.RECON CMDAUTH(SAF|EXIT|BOTH|NONE,safhlq)
```

Profiles can differ for different RECONs, which is controlled by safhlq. Commands can be authorized at:

- Command verb level

  For example, GENJCL command
- Command verb + resource type level

  For example, GENJCL.RECOV command
- Command verb + resource type + resource name level

  For example, GENJCL.RECOV DBD(ACCDB)

Example RACF definitions:

```
RDEFINE FACILITY IMOA.GENJCL.RECOV.ACCDB UACC(NONE)
```

This restricts GENJCL.RECOV DBD(ACCDB) command for the RECONs with high level qualifier of IM0A.

```
PERMIT IMOA.GENJCL.RECOV.ACCDB CLASS(FACILITY) ID(JOUKO4) ACCESS(READ)
```

This permits user JOUKO4 to issue the command.

Example command:

```
/RMGENJCL DBRC='RECOV DBD(ACCDB) DDN(ACCDB1)'.
```

This command is only allowed when submitted by user JOUKO4.

5.3 More than 32K database registrations

The maximum number of databases that could be registered to IMS in earlier releases was 32,767. When the last DMB number assigned reached 32,767, new database registrations would fail. If a database was deleted from the RECONs, its global DMB number was not reused.

IMS Version 9 allows the registration of more than 32,767 databases, although the maximum registered at any time is still 32,767. A global DMB number is assigned to each database when it is registered. The last number assigned is kept in the RECON header. This number is incremented when a database is registered.

With IMS Version 9, when the last number assigned reaches 32,767, IMS reads the database records to discover which numbers are in use. A bit map of these numbers is created. This bit map is used to reassign numbers no longer in use. The RECON header has been modified to add a field. In IMS Version 9 there is a last number reused field. IMS Version 9 uses this to process the bit map when reusing DMB numbers. The last number reused is listed as "LAST USED DMB#=" in a listing of the RECONs. It only appears when DMB numbers have reached 32767 and are being reused. The DMB#= field indicates the highest DMB number ever used in this RECON. An example of the **LIST.RECON** command output is shown in Example 5-1.

Example 5-1 Sample LIST.RECON illustrating the global DMB number for a database.

```
RECON
RECOVERY CONTROL DATA SET, IMS V9R1
DMB#=32767 LAST USED DMB#=6 INIT TOKEN=02296F1600074F
...
```

The following message is issued when the global DMB number reaches the old limit and the reassignment of global DMB numbers begins:

```
DSP1100I DBRC IS BUILDING THE DMB TABLE RECORD
```

If 32,767 databases are currently registered in the RECONs, an attempt to register another database will result in the issuing of the following message:

```
DSP1102I THE DMB LIMIT HAS BEEN REACHED
```

5.4 GENJCL.IC for HALDB

The **GENJCL.IC** command does not generate DD statements for HALDB data sets. These DD statements are never needed since dynamic allocation using DBRC information is always available for HALDB. The supplied ICJCL skeletal JCL member changed from IMS Version 8, so that dynamic allocation is now used.

The benefit, especially useful for HALDB OLR users, is that the IC utilities determine which are the active data sets. It then uses this data set. For example, if the M data set is active, the utility determines this and uses the M data set. It also dynamically allocates this data set. Including a DD statement for the A data set would cause a JCL error if the A data set did not exist. For this reason, it is recommended that DD statements for HALDB data sets never be included in image copy utility JCL.

5.5 DBRC migration and coexistence

IMS Version 9 provides the usual supported migration and coexistence paths as earlier releases. You can migrate from IMS Version 7 to IMS Version 9, and fall back is supported as long as you have the IMS Version 7 DBRC SPE installed (PQ72838).

You can also migrate from IMS Version 8 to IMS Version 9, with fallback supported through the IMS Version 8 DBRC SPE (PQ72840).

To upgrade your RECONs to IMS Version 9 format, the IMS Version 9 DBRC Utility (DSPURX00) is used with the following control card:

```
CHANGE.RECON UPGRADE
```

This utility may be executed while IMS subsystems are running. The upgrade fails if there is a subsystem record for an IMS Version 7 or IMS Version 8 subsystem without the DBRC SPE installed. Some utilities do not create subsystem records, and are therefore not protected by the check of subsystem records. If they are running without the SPE, unpredictable results may occur. Examples of such utilities include: change accumulation, log archive, DSPURX00, and the HALDB Partition Definition Utility (PDU).

The RECON upgrade process, from IMS Version 7 to IMS Version 9, reads the subsystem (SSYS) records to check for the DBRC SPE. The utility then updates every record from IMS Version 7 to IMS Version 8 format, which is done for segmented record support added in IMS Version 8. The RECON header is then upgraded from IMS Version 7 to IMS Version 9 format. The time required for the upgrade depends on the number of records in the RECON.

The upgrade of IMS Version 8 RECONs is much simpler. Only the header must be changed, since the other records do not have to be read or changed. This upgrade process should be very quick.

During the RECON upgrade process, existing HALDBs are upgraded to reflect that they are not capable of running HALDB online reorganization. There is a new field for the master database listing:

```
ONLINE REORG CAPABLE =YES|NO
```

There is also a new field for a partition listing:

```
ONLINE REORG CAPABLE =YES|NO
```

If you are running RSR, the following RSR fields will be created:

```
OLR ACTIVE HARD COUNT =n
OLR INACTIVE HARD COUNT =n
```

5.5.1 HALDB OLR coexistence

The ability to reorganize a HALDB while it is online is a new feature of IMS Version 9. However, this function is not retrofitted into earlier releases of IMS.

If a HALDB is capable of being reorganized online with OLR, then an IMS Version 7 system cannot access this database.

An IMS Version 8 system must have the OLR coexistence SPE installed before it is allowed to access any HALDB databases that are accessible with OLR. The SPE is provided through four APARs. They are PQ78493, PQ78758, PQ78916, and PQ78917. These IMS Version 8 systems can participate in data sharing while OLR is active in the IMS Version 9 system, but cannot invoke OLR.

An IMS Version 8 system without the OLR coexistence SPE cannot access a HALDB database that is defined as OLR capable in the RECON.

5.5.2 DEDB shared VSO multiple area structures

DEDBs using multiple area structures are flagged with the MAS attribute in the RECON. The MAS attribute is shown in Example 5-2.

Example 5-2 DEDB shared VSO using multiple area structures

```
DBDS
   DBD=DISTDB AREA=AREADI01 TYPE=FP
   SHARE LEVEL=3 DSID=00001 DBORG=DEDB DSORG=VSAM
   GSGNAME=**NULL** USID=0000000060
   AUTHORIZED USID=0000000060 RECEIVE USID=0000000060 HARD USID=0000000060
   RECEIVE NEEDED USID=0000000000
   CAGRP=PSAFPCA1 GENMAX=3 IC AVAIL=0 IC USED=3 DSSN=00000059
   NOREUSE RECOVPD=0 VSO PREOPEN PRELOAD
   CFSTR1=MASSTR1 CFSTR2=**NULL** LKASID MAS
...
```

If you are using DEDBs with multiple area structures, these areas are not available to be shared with IMS Version 7 or IMS Version 8 systems.

5.5.3 MINVERS format change

The minimum version of IMS that is allowed to read the RECONs is stored using the MINVERS parameter. This is a parameter to the **INIT.RECON** and **CHANGE.RECON** commands. In prior releases of IMS, the value was stored as 71 or 81. With IMS Version 9, a period has been added to the release number displayed from the RECON, for readability. An example of the command for changing the MINVERS is as follows:

```
   CHANGE.RECON MINVERS(‘9.1’)
```

The value of MINVERS is now 7.1, 8.1, and 9.1, although 71, 81, 91 continue to be accepted for compatibility. The minimum value for MINVERS with IMS Version 9 is 7.1. If a RECON with MINVERS of 61 is upgraded to IMS Version 9, the MINVERS value is changed to the minimum, 7.1. Otherwise, upgrades do not change the MINVERS value.

An IMS Version 8 system with the DBRC SPE installed will accept either format.

5.5.4 DSPSLDRC eliminated

The SLDS RECON record mapping macro (DSPSLDRC) has been removed from IMS Version 9. The RLDS (PRILOG) mapping macro (DSPLOGRC) may be used in its place, as the SLDS and RLDS RECON records have the same format.

6

Security considerations with IMS Version 9

In this chapter we describe and provide information about security options, that is, IMS Version 9 System Authorization Facility (SAF) enhancements that provide replacement functions for security maintenance utility (SMU) security and considerations for migrating from SMU security to SAF based security. We discuss these topics:

- 6.1, “Overview of security enhancements” on page 86
- 6.2, “Resource access security (RAS)” on page 86
- 6.2.2, “RAS migration examples” on page 89
- 6.3, “AOI security” on page 91
- 6.4, “MSC link security” on page 95
- 6.5, “/LOCK, /UNLOCK and /SET commands” on page 97
- 6.6, “Signon verification security” on page 99
- 6.7, “LTERM-based terminal security” on page 104
- 6.8, “Additional migration considerations” on page 110

6.1 Overview of security enhancements

IMS Version 9 provides several security enhancements. These enhancements provide System Authorization Facility (SAF) support for several security functions that were previously provided only by IMS internal modules known collectively as the security maintenance utility (SMU).

SAF is a system service that is part of the z/OS operating system. When the SAF receives a request from IMS, it routes the request to external security products such as the Resource Access Control Facility (RACF), user-supplied pre-RACF and post-RACF SAF exit routines, or both. IMS invokes the SAF by issuing the RACROUTE macro with keywords and parameter values that describe the authorization checking request. The SAF interface provides support for RACF as well as other equivalent vendor products. Although the term RACF is used throughout this chapter, the reader should interpret the term to mean 'RACF or an equivalent security product'.

Since IMS Version 9 is the last release to provide SMU support, the security enhancements in IMS Version 9 allow IMS users to eliminate the use of SMU and migrate to SAF/RACF. From a security perspective, IMS Version 9 is a migration release because it provides the same SMU security functions that are available in prior releases as well as SAF/RACF replacements for many of the SMU functions.

This enhancement is intended to satisfy the customer requirement to eliminate the need for SMU for any security function in IMS. For any SMU function not currently supported by RACF, support is added via the SAF interface in IMS Version 9. Customers don't want to have to maintain two security packages. The IMS Version 8 announcement letter stated that SMU security will no longer be supported in a future release. Additionally SMU limits the number of resources it supports, and this continues to be a frequent problem for IMS customers. There needs to be an alternative way to request that a statically-defined terminal be required to sign on.

6.2 Resource access security (RAS)

In prior releases of IMS, application group name (AGN) security was implemented with a combination of SMU and RACF or an IMS user exit (DFSISIS0) to provide resource access security. SMU was always required. This capability allowed an IMS security administrator to define groups of resources (PSB, transaction, LTERM) by an application group name (AGN) and provide a security mechanism to determine whether or not regions, for example, BMPs, could access the resources in the protected group.

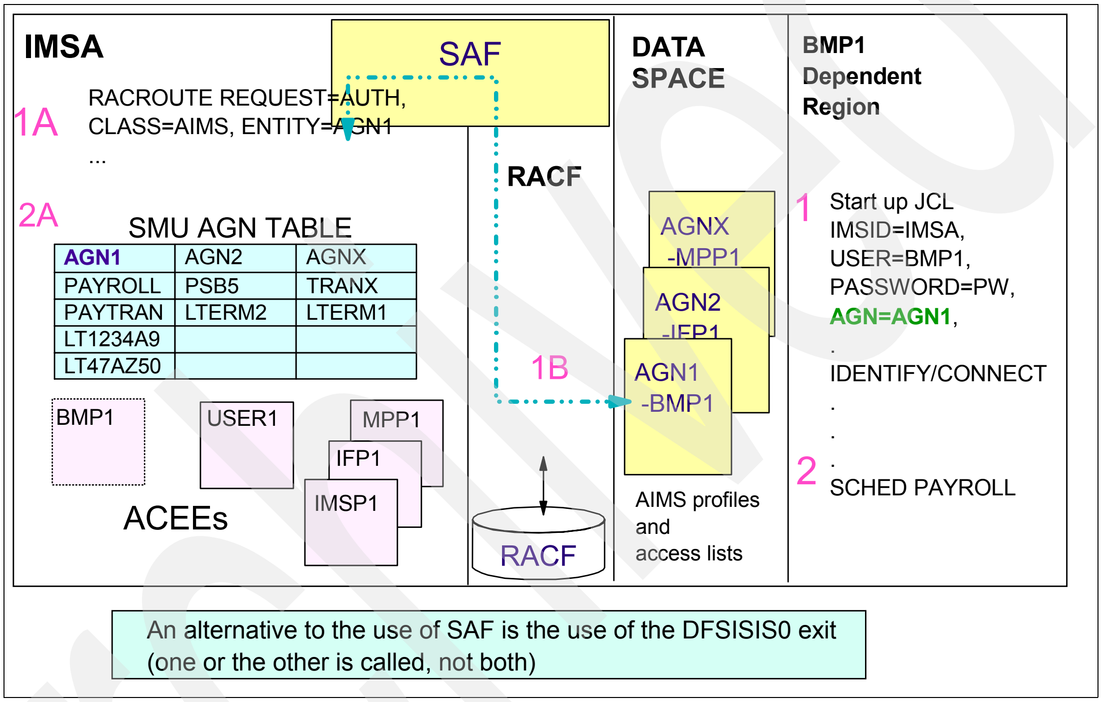
In a SMU and AGN environment, the checking process is a two-step process. The first check involves RACF. The second does not:

1. Each dependent region has a user ID associated with it. Before a dependent region can connect to IMS, IMS performs a RACF authorization check to see whether the dependent region user ID can use the application group name specified in it's AGN= startup parameter. IMS performs this check using the RACF class name (AIMS) and the name of the AGN passed to it in the EXEC statement parameter list. If the region's user ID is not PERMITted to use the application group name, RACF returns a “not authorized” return code, and IMS does not allow the dependent region to connect. If RACF returns an “authorized” return code, the connection is made.
2. The second part of the two-step process is an IMS function only. IMS checks the name of the transaction or PSB or logical terminal that is being requested by the dependent region

against the entries in the AGN Table and allows or disallows use depending on whether the name is in the entry for the application group name.

For message processing regions and Fast Path regions, application resource access security is somewhat like class scheduling in that transactions can be scheduled only in regions whose application group name allows them. For batch message processing regions, this level of control prevents unauthorized users from starting an z/OS job that can access the resources defined in the control region.

The Figure 6-1 describes the use of SMU and AGN as a review to better understand the changes in IMS Version 9. Note that systems migrating to IMS Version 9 can continue to use this process until a migration to RACF can be completed.



AGN1	AGN2	AGNX
PAYROLL	PSB5	TRANX
PAYTRAN	LTERM2	LTERM1
LT1234A9		
LT47AZ50		

An alternative to the use of SAF is the use of the DFSISIS0 exit (one or the other is called, not both)

Figure 6-1 Prior releases - SMU and AGN

In IMS Version 9, the two-step process is now a direct check of the user ID authorization against the specific resource. To accomplish this and ensure that the new RAS capability in IMS Version 9 provides equivalent protection, four new security classes for PSBs and LTERMs have been added.

The four new RACF security classes will be added to the RACF product as default classes. If your environment does not have the level of RACF that includes the new classes as defaults then they must be added to the installation-defined class descriptor table. Macro ICHERCDE is used to define a new class. The procedure for adding classes to this table is described in “Adding Installation-Defined Classes” in the *z/OS V1R5.0 Security Server RACF System Programmer's Guide,* SA22-7681. This is the same procedure that is used to change the default class names, for example, TIMS to Txxx where xxx is more meaningful to a specific environment. We provide an example of this later in this book, refer to “Additional migration considerations” on page 110.

IMS Version 9 provides direct SAF authorization checking of user to IMS resource by supporting the following new RACF security classes for PSBs and LTERMs:

IIMS Program specification block (PSB)

JIMS	Grouping class for PSB
LIMS	Logical terminal (LTERM)
MIMS	Grouping class for LTERM

The existing TIMS and GIMS classes continue to be used to protect transactions and groupings of transactions. APSB security using the existing AIMS security class will also continue to be done as in prior releases. The AIMS class will only be used for APSB security when SMU with AGN security is not selected for use.

The new support in IMS Version 9 is called resource access security (RAS). It is no longer called AGN. RACF, a new user exit, or both are used to provide protection. Either AGN security, or the new RAS can be used, but not both. If both are specified, then RAS is used. RAS is requested by using the system definition SECURITY macro with TYPE parameter or by execution parameter ISIS= definition. In both cases, the new values are provided in addition to the values that are already supported.

With IMS Version 9 you can have the following new specifications in system definition SECURITY macro:

TYPE=RASRACF	RAS security invokes RACF
TYPE=RASEXIT	RAS security invokes an IMS user exit DFSRAS00
TYPE=RAS	RAS security invokes RACF and user exit DFSRAS00
TYPE=NORAS	No security (turns off both RAS and SMU)

ISIS continues to be used for AGN and RAS security specifications. The parameters for AGN and RAS are unique, and can therefore correctly define the type of security desired. If ISIS=N is specified, there will be no authorization checking for the use of transactions, PSBs, and LTERMs by dependent regions. SMU will not be used either, because ISIS=1 or 2 must be specified (if the SECURITY macro has no specification) for SMU to be used for AGN. If ISIS is not specified, the default is the specification on the TYPE parameter of the IMS system definition SECURITY macro.

New specifications for ISIS execution parameter are the following:

ISIS=N	No security (turns off both RAS and SMU)
ISIS=R	RAS security invokes RACF
ISIS=C	RAS security invokes an IMS user exit DFSRAS00
ISIS=A	RAS security invokes RACF and user exit DFSRAS00
ISIS=,	defaults to SECURITY... TYPE= specification

The values that already existed for ISIS include:

ISIS=0	Deactivates AGN security and (RAS)
ISIS=1	Activates AGN security using SMU and RACF
ISIS=2	Activates AGN security using SMU and the resource access security exit routine (DFSISIS0)

A new user exit interface called the resource access security exit routine (DFSRAS00) provides authorization of IMS resources to IMS dependent regions. Although this exit replaces the use of the DFSISIS0 exit routine in a RAS environment, it is not quite a direct replacement.

DFSISIS0 was called to authorize a user to an AGN name. The AGN table with that AGN name contained the IMS resources that were indirectly authorized for use. DFSRAS00 user exit is optional and available in DCCTL, DB/DC, and DBCTL environments. DFSISIS0 remains available in an AGN environment for Version 9, but AGN security and the new RAS security cannot coexist in a single IMS system.

New user exit (DFSRAS00) is called after RACF (when both are used). It provides authorization of IMS resources to IMS dependent regions in a RAS environment.

RACF and/or DFSRAS00 make checks at scheduling time using region’s user ID as follows:

- Authorize user ID against transaction (MPP, JMP)
- Authorize user ID against PSB (IFP, NMD BMP, JBP, DRA,CCTL,ODBA)
- Authorize user ID against transaction and PSB (MD BMP)
- Authorize user ID against PSB and OUT=LTERM (NMD BMP, JBP)
- Authorize user ID against PSB and OUT=transaction (NMD BMP, JBP)

6.2.1 RAS and APSB security

When RAS is enabled, RAS check is made at every program schedule using the dependent region’s (MPP, JMP, BMP, IFP and JBP) user ID. The same applies to every CICS/DBCTL program schedule; the check is made using the user ID of the CICS® address space. Completely separately, CICS can perform a check of the terminal user against the PSB. RAS checking takes place at a program schedule, and requires that the PSB is defined in the IIMS RACF class. APSB security checking takes place for an “APSB Call” and requires that the PSB is defined in the AIMS RACF class. IMS never uses both checks for the same schedule. For the ODBA APSB call, whether to use APSB or RAS security is determined by the execution parameter ODBASE as follows:

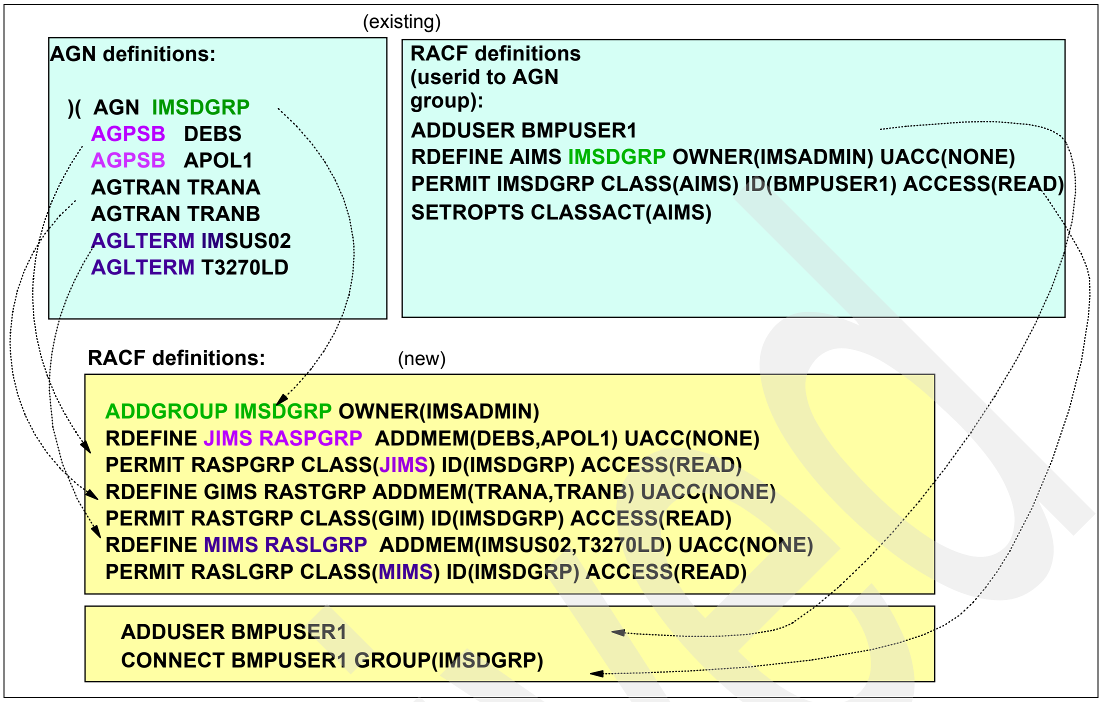
- ODBASE=Y means use APSB security
- With ODBASE=N, RAS (or AGN) security applies (if enabled)

For explicit APPC (CPI-C) APSB calls, if APSB security is performed (with the caller’s user ID), the RAS check will not be made. If APSB security is not performed, RAS check (if enabled) will be performed using the region’s user ID.

6.2.2 RAS migration examples

The next figures provide examples of converting from the use of SMU to SAF security.

Figure 6-2 on page 90 provides definitions for a BMP that is accessing PSB and LTERM resources.



(existing)

AGN definitions:

```
)(  AGN  IMSDGRP
    AGPSB   DEBS
    AGPSB   APOL1
    AGTRAN TRANA
    AGTRAN TRANB
    AGLTERM IMSUS02
    AGLTERM T3270LD
```

RACF definitions (userid to AGN group):

```
ADDUSER BMPUSER1
RDEFINE AIMS IMSDGRP OWNER(IMSADMIN) UACC(NONE)
PERMIT IMSDGRP CLASS(AIMS) ID(BMPUSER1) ACCESS(READ)
SETROPTS CLASSACT(AIMS)
```

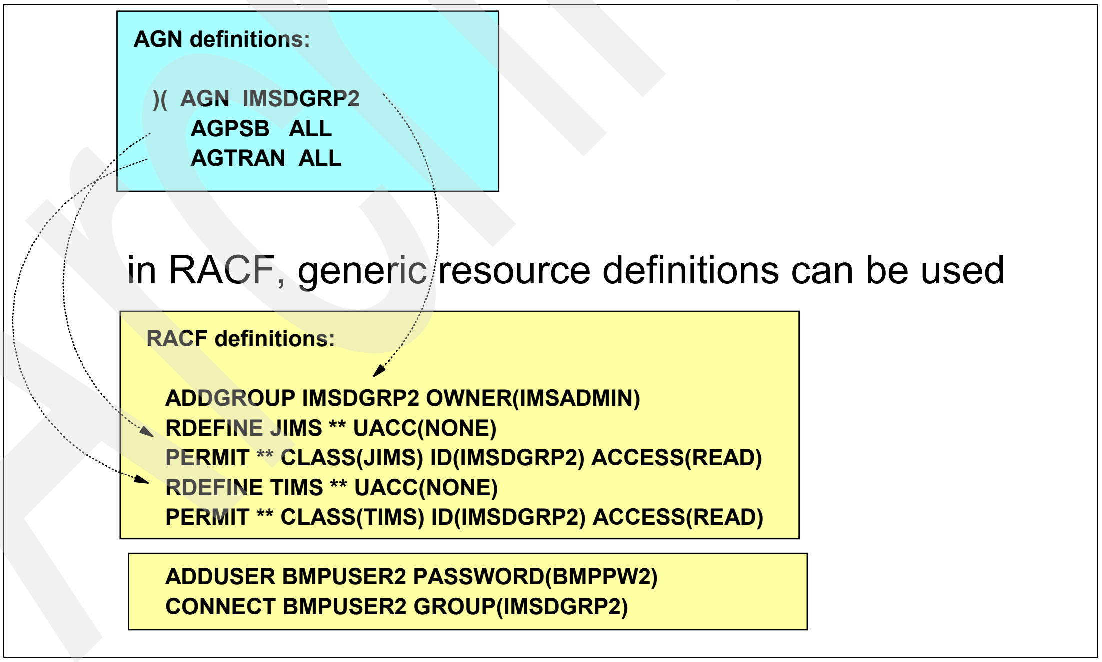
RACF definitions: (new)

```
ADDGROUP IMSDGRP OWNER(IMSADMIN)
RDEFINE JIMS RASPGRP  ADDMEM(DEBS,APOL1) UACC(NONE)
PERMIT RASPGRP CLASS(JIMS) ID(IMSDGRP) ACCESS(READ)
RDEFINE GIMS RASTGRP ADDMEM(TRANA,TRANB) UACC(NONE)
PERMIT RASTGRP CLASS(GIM) ID(IMSDGRP) ACCESS(READ)
RDEFINE MIMS RASLGRP  ADDMEM(IMSUS02,T3270LD) UACC(NONE)
PERMIT RASLGRP CLASS(MIMS) ID(IMSDGRP) ACCESS(READ)
```

```
ADDUSER BMPUSER1
CONNECT BMPUSER1 GROUP(IMSDGRP)
```

Figure 6-2 BMP accessing PSB, LTERM resources

Figure 6-3 provides definitions for an AGN name with access to all entities of a particular resource.



AGN definitions:

```
)(  AGN  IMSDGRP2
    AGPSB   ALL
    AGTRAN  ALL
```

in RACF, generic resource definitions can be used

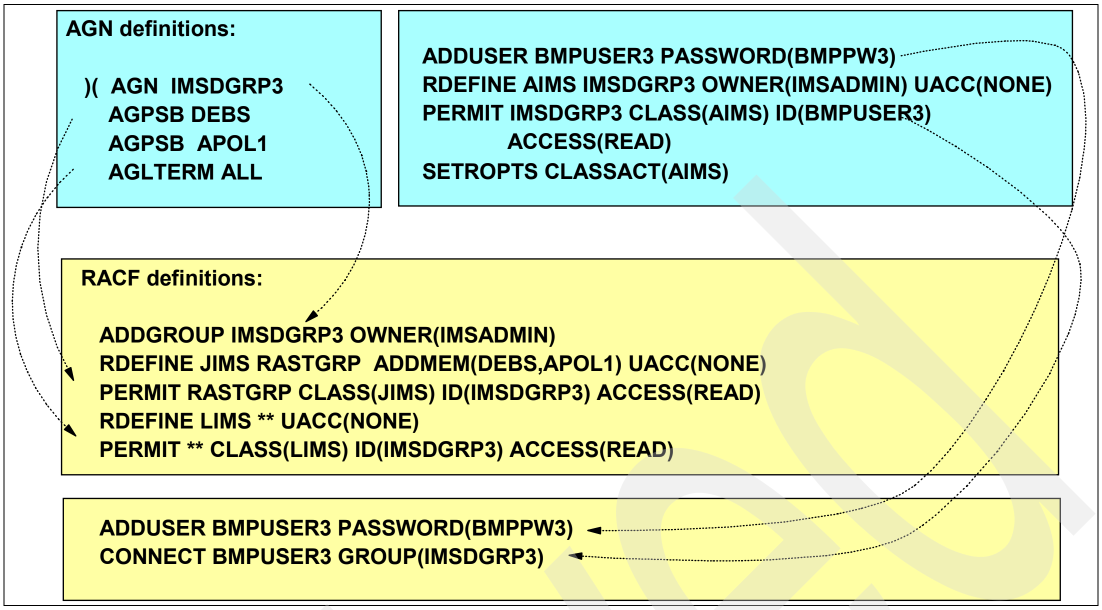
RACF definitions:

```
ADDGROUP IMSDGRP2 OWNER(IMSADMIN)
RDEFINE JIMS ** UACC(NONE)
PERMIT ** CLASS(JIMS) ID(IMSDGRP2) ACCESS(READ)
RDEFINE TIMS ** UACC(NONE)
PERMIT ** CLASS(TIMS) ID(IMSDGRP2) ACCESS(READ)
```

```
ADDUSER BMPUSER2 PASSWORD(BMPPW2)
CONNECT BMPUSER2 GROUP(IMSDGRP2)
```

Figure 6-3 AGN name with access to all entities of a particular resource

Figure 6-4 on page 91 shows a combination of resource grouping and generic resources definitions.



AGN definitions:

```
)(  AGN  IMSDGRP3
    AGPSB DEBS
    AGPSB  APOL1
    AGLTERM ALL
```

```
ADDUSER BMPUSER3 PASSWORD(BMPPW3)
RDEFINE AIMS IMSDGRP3 OWNER(IMSADMIN) UACC(NONE)
PERMIT IMSDGRP3 CLASS(AIMS) ID(BMPUSER3)
           ACCESS(READ)
SETROPTS CLASSACT(AIMS)
```

RACF definitions:

```
ADDGROUP IMSDGRP3 OWNER(IMSADMIN)
RDEFINE JIMS RASTGRP  ADDMEM(DEBS,APOL1) UACC(NONE)
PERMIT RASTGRP CLASS(JIMS) ID(IMSDGRP3) ACCESS(READ)
RDEFINE LIMS ** UACC(NONE)
PERMIT ** CLASS(LIMS) ID(IMSDGRP3) ACCESS(READ)
```

```
ADDUSER BMPUSER3 PASSWORD(BMPPW3)
CONNECT BMPUSER3 GROUP(IMSDGRP3)
```

Figure 6-4   Combination of resource grouping and generic resources definitions

6.3 AOI security

Prior to IMS Version 9, commands issued by type 1 automated operator (AO) programs (programs using CMD calls) were secured using SMU transaction command security profiles. AO programs were restricted to issuing only the commands defined in the SMU profiles. When the type 1 AOI CMD call was issued, IMS performed the security checking using the tables/matrixes loaded during restart.

On the other hand, commands issued by type 2 AO programs (programs using ICMD calls) were secured using command profiles stored in the RACF CIMS class. Type 2 programs were also optionally secured using a user exit routine (DFSCCMD0) which could perform command authorization independently or in conjunction with RACF. When security checking was required for command keywords, the exit provided a way to implement more granular levels of command security.

IMS Version 9 provides several enhancements in this area. Type1 AOI CMD calls are now secured using the SAF interface in a fashion similar to that which was already provided for the type 2 ICMD calls. Additionally, a new parameter on the TRANSACT macro provides a greater level of granularity that can make the SAF security check closer to that which was provided for SMU.

6.3.1 SAF support for type 1 AOI (CMD)

Prior to IMS Version 9, the ICMD call of type 2 AOI (DFSAOE00) already uses the SAF interface and the command authorization exit (DFSCCMD0) for security. The CMD call of type 1 AOI (DFSAOUE0), used SMU security in prior releases. In IMS Version 9, this has been enhanced to provide the option of using the SAF interface and the DFSCCMD0 user exit.

A new startup parameter in IMS Version 9, AOI1= can be specified to indicate which security product is to be used for authorization of commands for type 1 AOI, and the level of security. If no value is specified, then IMS uses the specification in the SECURITY macro as defined in the system definition process. AOI1= parameter can be specified as follows:

AOI1=A	Includes options C and R below. SAF (RACF) is called first then DFSCCMD0
AOI1=N	No authorization security checking is done
AOI1=C	DFSCCMD0 is called for command authorization
AOI1=R	RACF is called for command authorization
AOI1=S	SMU security is called for command authorization
AOI1=,	Defaults to system definition specification on SECURITY macro

Because the AOI1 specification is not included in a checkpoint record, the AOI1 value can be changed each time IMS is initialized.

During an IMS restart, the `/NRE` or `/ERE` can specify a value of either TRANCMDS or NOTRANCMDS. This only applies to TYPE 1 AOI security. If neither value is specified on a restart then IMS uses the AOI1 startup specification or what has been defined in the SECURITY macro. A specification of TRANCMDS causes SMU to be used for type 1 AOI security and overrides TRANCMD=N in system definition and AOI1 in startup. NOTRANCMDS results in no TYPE 1 AOI security, unless prevented by the specification of TRANCMD=F during system definition and a further override by AOI1= parameter.

New parameter for TRANSACT macro

The TRANSACT macro in IMS system definition has a new parameter, AOI=, which specifies whether or not a particular transaction is allowed to issue the AOI command (CMD) call. In prior releases, this information was derived during IMS restart from the SMU matrix tables. This new parameter applies to both type 1 and type 2 AOI and is modifiable by online change. AOI= can be specified as follows:

- AOI=YES
- AOI=NO
- AOI=TRAN
- AOI=CMD

When AOI=YES is specified, the authorization of the commands for the CMD calls issued by the transaction is done using the user ID of the user who entered the transaction. For some environments, if a get unique (GU) call has not yet happened, then the program name rather than the user ID is used for the authorization.

When AOI=NO is specified, no authorization is permitted. Type 1 AOI CMD calls cannot be issued.

AOI=TRAN specification is similar to that of YES, but requests that the transaction code, be used instead of the user ID of the user who entered the transaction. Use of the transaction code provides authorization checking more like that provided by the SMU transaction-command security. When a transaction is defined with AOI=TRAN, the first authorization check done for AOI for the transaction results in the access control environment element (ACEE) being built and being kept for use by future authorization checks. In this case, the type 1 AOI transactions have to be defined to RACF as a user. The transactions must also be specified on RACF PERMIT statements for each command they are allowed to issue from a type 1 AOI transaction.

AOI=CMD specification is also similar to that of YES, but requests that the command code (first three characters of the command), be used instead of the user ID for the authorization check. Use of the command code provides authorization checking more like that provided by the SMU transaction-command security. When a transaction is defined with AOI=CMD, the first authorization check done results in the security environment (ACEE) being built, and being kept for use by future authorization checks. In this case, the IMS command codes have to be defined to RACF as a user. The command codes must also be specified on RACF PERMIT statements for each type 1 AOI transaction that is allowed to issue them.

6.3.2 SAF support for type 2 AOI (ICMD)

The ICMD call for type 2 AOI (DFSAOE00) already uses the SAF interface along with the command authorization exit (DFSCCMD0) for security. The existing parameter AOIS continues to be valid with the following values:

AOIS=A RACF and DFSCCMD0 are called for ICMD command authorization. RACF is called first, then DFSCCMD0.

AOIS=C DFSCCMD0 is called for ICMD command authorization.

AOIS=N ICMD is not issued by any application program. N is the default.

AOIS=R RACF is called for ICMD command authorization.

AOIS=S Skip command authorization; all application programs can issue ICMD calls.

The new parameter, AOI=, which was described in the section about type 1 AOI, also applies to type 2 AOI security. However, if AOI=NO is specified or defaulted, all transactions are still allowed to use type 2 AOI and issue the ICMD call. It will be as though AOI=YES had been specified.

If the TRAN or CMD specification is used for type 2 AOI, IMS uses the transaction code or command code, instead of the user ID of the user who entered the transaction. This is used to check the authorization of the commands for the ICMD calls issued by the transaction. Use of the transaction code or command code provides authorization checking more like that provided by the SMU transaction-command security. If the IMS environment does not support the use of a transaction, such as a non-message-driven BMP, then this new AOI= specification will not be available, and the security checking will be like prior IMS releases (or as though YES was specified).

6.3.3 SAF support for time controlled operations (TCO)

Time controlled operations (TCO) is an IMS capability to execute time-initiated commands and transactions. TCO can generate any IMS input that an IMS operator can, except for the IMS restart commands, **/NRESTART** and **/ERESTART**. Additionally, it cannot initiate conversational transactions, full-function response mode or Fast Path input transactions.

The scripts used by TCO to execute the time-initiated commands, transactions, and message switches are stored in IMS.TCFSLIB. TCO support provides security at two different levels. One capability restricts who can load TCO scripts. The other restricts which IMS commands or transactions can be accessed.

The determination of who can load TCO scripts is done by the TCO CNT edit exit (DFSTCNT0). This exit is used instead of the message switching (input) edit routine (DFSCNTE0) when the message switch destination is DFSTCF. Neither SMU security nor RACF are used to determine who is authorized. This is not a change from previous releases.

6.3.4 Resource authorization

In prior releases, IMS commands and transactions issued by TCO had the choice of using either SMU or RACF for authorization. The use of SMU security for TCO command and transaction authorization was predicated on defining the TCO LTERM names DFSTCF and

DFSTCFI in the SMU definitions. More specifically, the SMU definitions were used to define which resources could be issued by a TCO script running under this LTERM name as shown in Example 6-1.

Example 6-1 SMU sample for TCO

```
)( TERMINAL    DFSTCFI
          COMMAND    START
          COMMAND    STOP
    TRANSACT    STATTRN
)( COMMAND    START
          TERMINAL    DFSTCFI

)( COMMAND    STOP
    TERMINAL    DFSTCFI
```

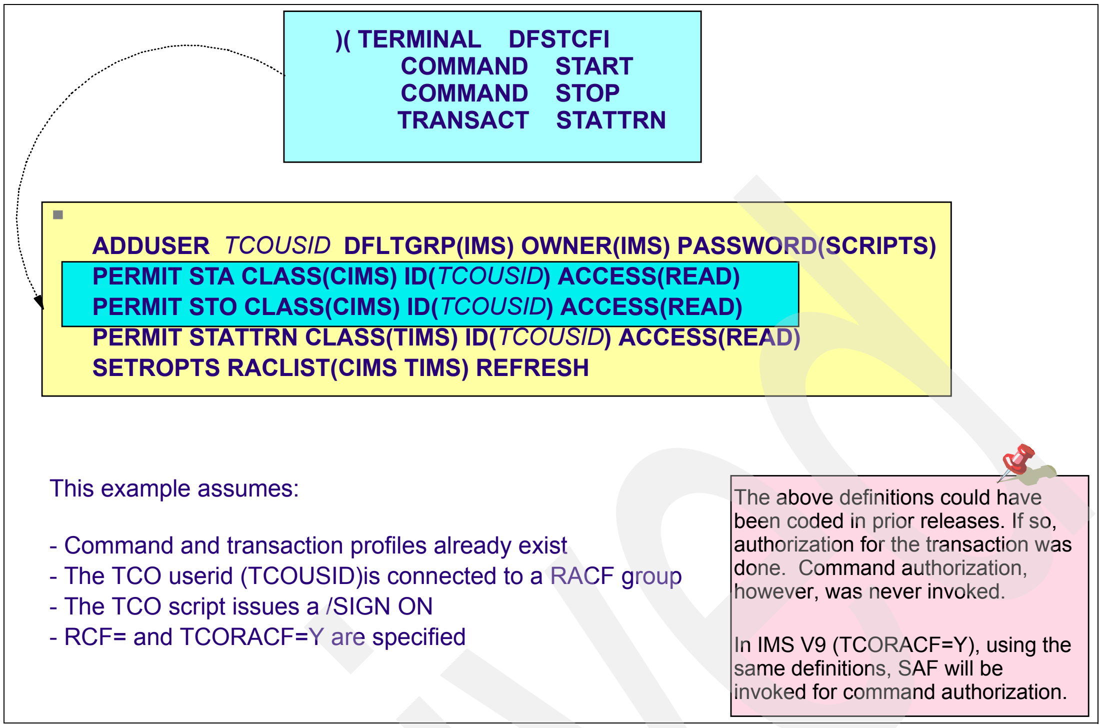
On the other hand, when the IMS startup parameter RCF was specified, SMU authorization was not used. If the script issued a /SIGN ON command, an ACEE was created and the associated user ID was available to be used in SAF calls to authorize access to transactions, but not for SAF authorization of commands.

SAF was not called to authorize access to commands because TCO was already automatically authorized to issue the same set of commands as the system console and master terminal. If the command authorization exit (DFSCCMD0) existed, it was called for command authorization.

The values of RCF = A | B | C | N | R | S | T | Y that were available in previous releases of IMS are still valid in IMS Version 9.

In IMS Version 9, a new execution parameter, TCORACF=, is used to indicate whether (Y) or not (N) the SAF interface should be called to perform an authorization check of commands from a TCO script. The value can be changed each time IMS is initialized. Therefore, specification of RCF has additional criteria. Only if both the IMS startup parameter RCF=A|S|R|B and TCORACF=Y are specified, is the SAF interface invoked to call RACF. The command authorization exit routine (DFSCCMD0) is also called, if it exists. Coding a /SIGN ON command at the beginning of the TCO script can provide the user ID which will be used for authorization. The user ID defined in the /SIGN ON is signed on to IMS and not signed off until a /SIGN OFF at the end of the script. Any commands after the /SIGN ON are checked by RACF for being authorized for use by the signed on user. The commands are also passed to the DFSCCMD0 exit. DFSCCMD0 can check the input CNTNAME to see if the input is from TCO (DFSTCFI), and if it is, allow the /SIGN command without checking the user ID. For other commands, the exit can authorize the signed on user to enter that particular command.

> **Note:** When providing a user ID/group in the /SIGN ON at the beginning of a script, the user ID/group provided must have been previously defined to RACF. If it has not, RACF returns a RC=4. As a result of the RC=4, IMS issues a FASTAUTH using the IMS control region's ACEE. If the control region's user ID/group is authorized to the command then the TCO script can also access the command. If the control region's user ID is not authorized to the command then an RC=8 is issued to deny access.



```
)( TERMINAL   DFSTCFI
   COMMAND    START
   COMMAND    STOP
   TRANSACT   STATTRN
```

```
ADDUSER  TCOUSID  DFLTGRP(IMS) OWNER(IMS) PASSWORD(SCRIPTS)
PERMIT STA CLASS(CIMS) ID(TCOUSID) ACCESS(READ)
PERMIT STO CLASS(CIMS) ID(TCOUSID) ACCESS(READ)
PERMIT STATTRN CLASS(TIMS) ID(TCOUSID) ACCESS(READ)
SETROPTS RACLIST(CIMS TIMS) REFRESH
```

This example assumes:

- Command and transaction profiles already exist
- The TCO userid (TCOUSID)is connected to a RACF group
- The TCO script issues a /SIGN ON
- RCF= and TCORACF=Y are specified

The above definitions could have been coded in prior releases. If so, authorization for the transaction was done. Command authorization, however, was never invoked.

In IMS V9 (TCORACF=Y), using the same definitions, SAF will be invoked for command authorization.

Figure 6-5 TCO command authorization sample

The Figure 6-5 assumes that RACF security profiles already exist for the START and STOP commands and for the STATTRN transaction. The TCO user ID, TCOUSID, is added to the access lists for each of the commands and for the transaction.

Although this definition could be provided in prior releases, it is not until IMS Version 9 that IMS will invoke the CIMS profile to authorize access for a TCO script.

6.4 MSC link security

Multiple systems coupling (MSC) directed routing is a function of IMS that allows for the routing of messages to other IMS systems without the target resources having to be defined as remote resources in the sending IMS. This capability relies on the application, in the sending IMS, to specify the logical link path (MSNAME) as the destination of the message. Additionally, the sending application provides the actual target LTERM or transaction name in the first few bytes of the IOAREA. When the message is received at the destination, the LTERM or transaction code is removed from the message by the link receive entry point of the DFSMSCE0 exit and the message enqueued for the proper destination.

Non-directed routing, on the other hand, relied on all target destinations in the remote IMS to be defined as remote resources on the sending system. For releases prior to IMS Version 9 where MSC link input to transaction destinations used directed routing, SAF(RACF) and the DFSCTRN0 user exit were called before and after the DFSMSCE0 user exit call. Otherwise, for non-directed routing, SMU security was called after the DFSMSCE0 call. Figure 6-6 on page 96 shows the difference between directed and non-directed routing.

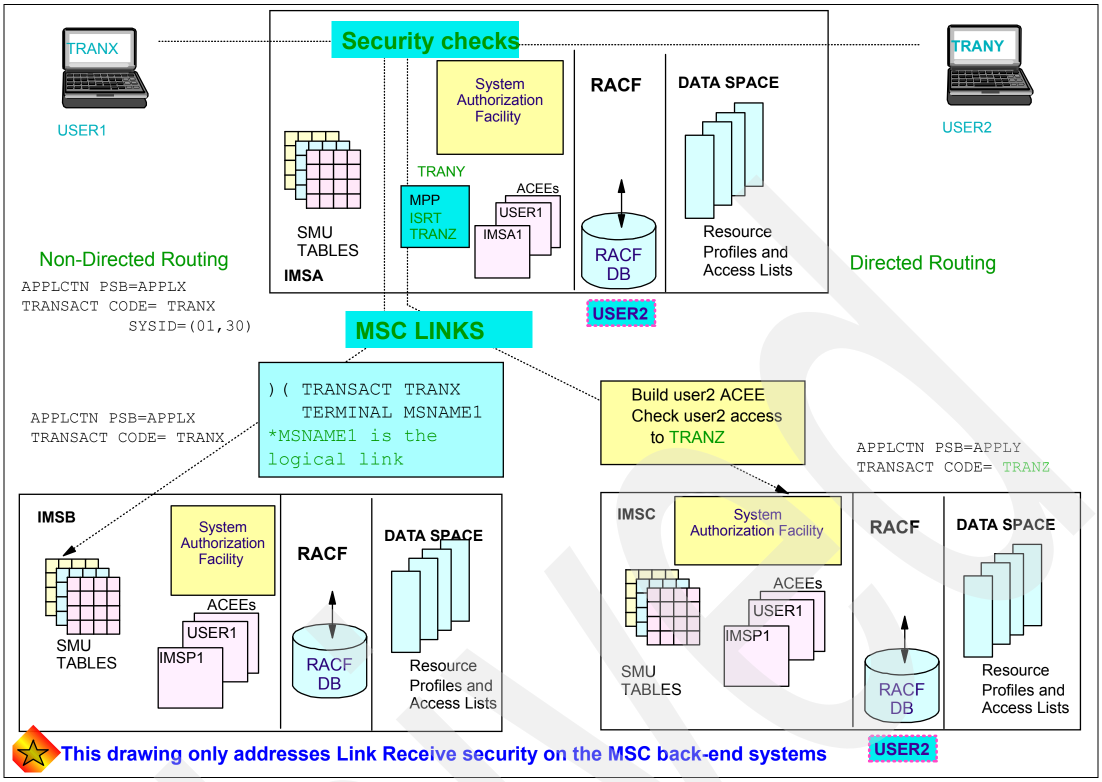


Figure 6-6 MSC link security

For IMS Version 9, a new startup parameter defines the type of security required for MSC. In the DFSDCxxx PROCLIB member, the new keyword is MSCSEC, and it can be defined as follows:

```
MSCSEC=(parm1,parm2)
```

The first parameter (parm1) can have the following values:

LRDIRECT	Link receive directed routing transaction security checking
LRNONDR	Link receive non-directed routing transaction security checking
LRALL	LRDIRECT and LRNONDR
LRNONE	No link receive security checking

The value of the second parameter (parm2) in MSCSEC specifies the level of authorization and can be one of the following:

CTL	Authorization by CTL address space security
MSN	Authorization by MSNAME
USER	Authorization by user ID of inputting terminal
EXIT	Authorization by user exit (DFSCTRN0)
CTLEXIT	Authorization by CTL address space security and by user exit (DFSCTRN0)
MSNEXIT	Authorization by MSNAME and by user exit (DFSCTRN0)
USREXIT	Authorization by user ID of inputting terminal and by user exit (DFSCTRN0)
NONE	No security authorization checking

With the link receive security available in previous releases, directed routing and non-directed routing use different user IDs. To provide similar flexibility with the new security, the DFSMSCE0 user exit can be used to dynamically set the required level of authorization by overriding the parm2 value.

6.4.1 IMS Version 9 MSC security considerations

Some considerations in the IMS Version 9 environment include the following:

- SMU security can still be invoked when non-directed routing is used and MSCSEC=LRNONDR is not specified.
- There will no longer be calls to SAF and the DFSCTRN0 user exit prior to calling the DFSMSCE0 user exit during link receive processing.
  - DFSMSCE0 can set the level of authorization checking
  - Calls to SAF and DFSCTRN0 are made after DFSMSCE0
- Additional data will be passed to the DFSMSCE0 user exit including: user ID, group name, and user ID indicator. The user ID indicator clarifies the value in the user ID field as being one of: user ID, LTERM name, PSB name, MSNAME, or other.
- RACF profiles should be kept synchronized on sending and receiving systems

The DFSMSCE0 user exit call during link receive processing has the ability to control the level of authorization checking. The user exit response in field MSLRFL3 specifies the level of authorization as one of the following:

- Authorization by MSNAME where the ACEE is dynamically created for the first authorization, then reused.
- Authorization by CTL address space security.
- Authorization by user ID of inputting terminal where the ACEE is dynamically created and deleted for each authorization.
- Authorization by user exit DFSCTRN0.
- No security authorization checking.

On entry to the DFSMSCE0 user exit for link receive, the MSLRFL3 field contains the system default value from the MSCSEC= startup parameter.

The security authorization options that can be returned by the DFSMSCE0 user exit in field MSLRFL3 are also provided as a startup option. These new options are specified as the second value in the MSCSEC= keyword in the DFSDCxxx PROCLIB member as described in the previous section.

6.5 /LOCK, /UNLOCK and /SET commands

The `/LOCK` command is used to stop the sending and receiving of messages to and from a terminal, stop the scheduling of messages containing a specific transaction code, stop the scheduling of a specific program, stop the use of a database, and stop the sending of messages to and from an LTERM, a NODE or a physical terminal. The `/UNLOCK` command releases the resource. `/SET` command establishes the destination of all messages entered into this terminal to another terminal or to a particular transaction code. If the terminal is in conversation, the `/SET` command also sets the destination of only the next message to the specified transaction.

The /LOCK, /UNLOCK and /SET commands also support specification of password security for the defined resource. In prior releases, this was accomplished with the use of SMU. If the resource was not defined with password protection in SMU, the check was not made and the password ignored.

Password protections was by the specification of the PASSWD= parameter on the SECURITY macro along with the appropriate SMU definitions. There were, however, certain restrictions:

- The **/LOCK DATABASE**, **/LOCK PROGRAM**, and **/LOCK TRANSACTION** commands are only valid when entered from a master terminal, system console, TCO script, or AO program.
- The **/LOCK TRANSACTION** command cannot be used for CPI-C driven programs.
- The **/LOCK LTERM**, **/LOCK NODE**, and **/LOCK PTERM** commands apply only to the entering physical terminal.

In prior releases, SMU provided password security. For example, if the SMU definitions were provided as seen in Example 6-2, and the following conditions were met:

- in SECURITY macro in IMS system definition, PASSWD=YES is defined
- IMS is started using **/NRE** or **/ERE COLDSYS PASSWORD** -command

The operator needed to specify the password to be able to enter the **/LOCK**, **/UNLOCK** or **/SET** command, as follows:

- /LOCK DATABASE payroll (uomecash)
- /UNLOCK DATABASE payroll (uomecash)
- /SET TRANSACTION paytran (uomecash)

Example 6-2 SMU definition sample

```
)( DATABASE  PAYROLL
   PASSWORD  UOMECASH

)( PASSWORD  UOMECASH
   DATABASE  PAYROLL
   PROGRAM   PAYPROG
   TRANSACT  PAYTRAN
```

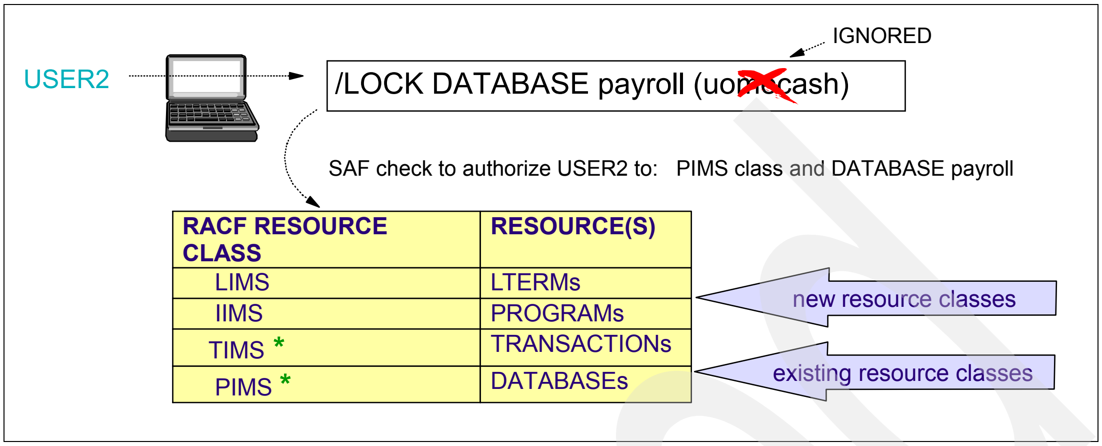
In IMS Version 9, the security of the LTERM, database, program, and transaction parameters include the use of SAF (RACF) and/or the transaction authorization exit (DFSCTRN0). The resources are defined to RACF in the classes: LIMS, PIMS, IIMS, and TIMS respectively. Because this SAF and exit call were not made in prior releases, the call cannot be made unconditionally beginning in IMS Version 9. Therefore, a new startup option LOCKSEC=Y|N (where N is the default) has been added to the DFSDCxxx IMS.PROCLIB member.

If LOCKSEC=Y is specified, the new SAF and user exit calls are made after the optional call to SMU security. If the resource is not defined to RACF, or is defined and is authorized to the user, the command is processed. If the resource is defined to RACF but is not authorized for use by the user entering the command, the command is rejected with the following new message:

```
DFS3689W USE OF <TRANSACTION|LTERM|DB|PROG> resourcename BY <LOCK|UNLOCK> REJECTED
```

If the user is authorized to issue the /LOCK, /UNLOCK and /SET commands with CONVERSATION and TRANSACTION keywords, another check is made to authorize access to resources. Refer to Figure 6-7 on page 99. Note that no SAF call is made for /LOCK and /UNLOCK NODE | PTERM. Protection of VTAM nodes and BTAM terminals relies on the use of RACF TERMINAL|GTERMINL support. The password after the parameter "payroll" can optionally be checked if the authorization check of the parameter is successful. If the RACF profile for the IMS resource (such as "payroll") is defined with the parameter

"APPLDATA('REVERIFY')", then IMS will check that the password is the same as the user's signon password.


IGNORED
USER2
/LOCK DATABASE payroll (uomicash)
SAF check to authorize USER2 to: PIMS class and DATABASE payroll
new resource classes
existing resource classes

RACF RESOURCE CLASS	RESOURCE(S)
LIMS	LTERMs
IIMS	PROGRAMs
TIMS *	TRANSACTIONs
PIMS *	DATABASEs

Figure 6-7 /LOCK command authorization checking

6.6 Signon verification security

In previous releases, statically defined terminals were only required to signon if the terminal was defined in SMU as being required to do so. This was done with the)(SIGN control statement. SMU, however, was limited to supporting:

- A maximum of 65535 LTERM definitions
- A maximum of 32767 different patterns for sets of commands and transactions (a pattern is a unique set of commands and/or transactions that can be issued by terminals)
- A maximum of 32767 terminals to require sign-on

Example 6-3 shows the SMU method for static terminal signon verification, that defines which terminals have to sign on using the **/SIGN ON** command.

Example 6-3 SMU input statements for requesting signon security

```
)(SIGN
     STERM TERM1
     STERM TERM2               OR          STERM ALL
     STERM TERM3
```

In IMS Version 9, a new initialization parameter, SIGNON=, is provided in the DFSDCxxx PROCLIB member to request that static terminals be required to signon. The format and the options for SIGNON are:

- SIGNON=ALL
- SIGNON=SPECIFIC

Specification of ALL requests that all static terminals be required to signon. This is equivalent to the SMU definition of)(SIGN STERM ALL. Note that like the SMU support, SIGNON=ALL will not set sign on required in LU6.1, 3284/3286, SLU1 (when printer-only device), and master terminal devices.

The use of SPECIFIC defines that individual static terminals may be required to signon. These terminals will be specified in either system definition with the OPTIONS=SIGNON parameter on the TYPE and/or TERMINAL macros, or via SMU definitions with the)(SIGN statements.

Signon verification security can also be defined for individual terminals via new options on the IMS system definition TYPE and TERMINAL macro statements. The new parameters on the OPTIONS= keyword are SIGNON and NOSIGNON, with NOSIGNON as the default on the TYPE macro. The specification on the TERMINAL macro, if specified, overrides the value from the TYPE macro and no specification on the TERMINAL macro defaults to TYPE macro.

Signon verification for BTAM terminals can be requested on the IMS system definition TERMINAL macro statement using the OPTIONS= SIGNON | NOSIGNON. This specification only applies to TERMINAL macro, and is not available on LINEGRP or LINE macros.

Whatever signon option is requested in system definition for a static terminal, can also be requested in SMU. The SMU settings are applied during IMS restart. The setting in the terminal control block specifying signon required is not reset by not being specified in SMU. The signon required specification can be made in system definition and SMU.

6.6.1 Sign on for static terminals

By default, IMS does not require static terminals to sign on. In IMS Version 8 and prior releases, three specifications are coded to:

- Determine whether or not sign on security is activated for one or more static terminals. The sign on security option is specified on SECLVL statement on the SECURITY macro.
- Determine whether or not SMU is invoked to enforce sign on for the static terminals that are required to sign on. The value specified on the TERMNL statement on the SECURITY macro specifies whether:
  - SMU sign on security is activated for static terminals
  - LTERM-based terminal security is activated for commands and transactions entered from static terminals
  - Both sign on and LTERM-based security are activated
- Identify the specific static terminals for which SMU is invoked to enforce sign on. The static terminals that are required to sign on are identified in SMU statements which supply the input for the SMU generation process. The static terminals that are required to sign on are identified in)(SIGN and STERM statements in SMU generation input.

6.6.2 Security level (SECLVL)

The value supplied for the sign on security level on the SECLVL= keyword on the SECURITY macro determines whether or not sign on security is activated for one or more static terminals. It is important to note that the value supplied for SECLVL has no effect on SMU security. It only indicates whether or not sign on security is activated in the IMS system. SECLVL does not designate which security facility (for example SMU, RACF, a user-coded exit routine, or a combination of these) is to be invoked to enforce sign on. Table 6-1shows the possible SECLVL values and their meanings.

Table 6-1 SECLVL values

SECLVL Value	Meaning
SECLVL=(...,NOSIGN)	Sign on security is not activated, but may be activated on the /NRE or /ERE command issued to restart IMS. NOSIGN is the default.

SECLVL Value	Meaning
SECLVL=(...,SIGNON)	Sign on security is activated, but may be deactivated on the /NRE or /ERE command issued to restart IMS.
SECLVL=(...,FORCSIGN)	Sign on security is activated and FORCED. This means sign on security cannot be deactivated on the /NRE or /ERE command issued to restart IMS.

The value specified (or defaulted to) for the sign on security level on the SECLVL keyword on the SECURITY macro may be overridden by the value specified for the SGN= keyword in IMS execution parameters in the DFSPBxxx member of IMS.PROCLIB. SGN=Y and SGN=F activate sign on security for static terminals and are equivalent to SECLVL=(...,SIGNON) and SECLVL=(...,FORCSIGN), respectively. The USER keyword on the /NRE and /ERE commands also activates sign on security. Specification of the USER keyword on the /NRE and /ERE overrides both SECLVL= (...,NOSIGN) on the SECURITY macro and SGN=N in IMS execution parameters.

Note: Forced sign on security may be deactivated on the SGN= keyword in IMS execution parameters in the DFSPBxxx member or IMS.PROCLIB.

Since the specifications for SECLVL and SGN merely activate or deactivate sign on security for static terminals and do not affect SMU security, replacement functions are not needed. The specifications for SECLVL and SGN have the same meanings in IMS Version 9 as in prior releases.

6.6.3 SMU terminal (TERMNL) security

The value supplied for the TERMNL= keyword on the SECURITY macro determines whether or not SMU modules are invoked to enforce LTERM-based terminal security in the IMS system. Table 6-2 shows the possible TERMNL values and their meanings.

Table 6-2 TERMNL values

TERMNL Value	
TERMNL=NO	SMU modules are not invoked to enforce LTERM-based terminal security. LTERM-based terminal security may be activated on the /NRE or /ERE command issued to restart IMS. TERMNL=NO is the default.
TERMNL=YES	SMU modules are invoked to enforce LTERM-based terminal security. LTERM-based terminal security may be deactivated on the /NRE or /ERE command issued to restart IMS.
TERMNL=FORCE	SMU modules are invoked to enforce LTERM-based terminal security. LTERM-based terminal security cannot be deactivated on the /NRE or /ERE command issued to restart IMS.

IMS Version 9 SAF/RACF replacement for SMU does not provide a replacement for SMU LTERM-based terminal security. SMU LTERM-based terminal security is used to restrict the entry of IMS commands and/or transactions to one or more authorized LTERMs. SAF/RACF functions provide user ID-based command and transaction authorization checking. Unlike SAF/RACF, SMU provides terminal-based command and transaction authorization checking. If your company has a requirement to provide LTERM terminal-based security checking for commands and transactions entered from static terminals, a user-coded exit routine may be used in conjunction with SAF/RACF user ID-based command and transaction authorization security to accomplish this.

The migration considerations for LTERM-based terminal security are described in the topic “LTERM-based terminal security” on page 104.

6.6.4 SMU)(SIGN and STERM statements

The static terminals that are required to sign on can be specified in SMU using the)(SIGN and STERM control statements. The terminals required to perform sign on are identified in SMU input statements that are processed during the execution of the security maintenance utility, or SMU generation. SMU security definitions are created by executing the SECURITY procedure, a member of IMS.PROCLIB. One of the outputs of a SMU generation is the signon offset list (DFSISSOx) module. The signon offset list may be referred to as the signon matrix or the signon table. Example 6-4 shows a sample input statements of SMU generation to require several static terminals (NODE1, NODE2, and NODE3) to sign on, and a sample statement to require all static terminals to sign on.

Example 6-4 SMU generation sign on input statements

```
)( SIGN
   STERM NODE1 *Sign on is required for NODE1*
   STERM NODE2 *Sign on is required for NODE2*
   STERM NODE3 *Sign on is required for NODE3*
-OR-
)( SIGN
   STERM ALL   *Sign on is required for all static terminals*
```

When SMU is used to enforce sign on, DFSISSOx is loaded from the active IMS.MATRIX data set into IMS control region storage during restart. Any input from a static terminal causes IMS to check DFSISSOx to determine if the terminal is required to sign on. When a static terminal is required to sign on, the terminal will only accept the /SIGN and /RCLSDST commands.

6.6.5 IMS Version 9 sign on security for static terminals

In prior IMS versions, either SAF/RACF or a user-coded exit routine is needed to perform sign on security for static terminals. Although SMU modules can ensure the /SIGN ON command is entered from a static terminal, there is no logic in SMU modules that verify the user ID and password supplied on the command. SAF/RACF, the sign on exit routine (DFSSGNX0), the sign on/off security exit routine (DFSCSGN0), or a combination of these is required to validate the user ID and password supplied on the /SIGN ON command.

As in prior versions, the inclusion of one or both of the user-coded exit routines in the IMS system and the value specified for the RCF= execution parameter in the DFSPBxxx member of IMS.PROCLIB determine whether user exits and/or SAF/RACF is used to validate the user ID and password. The RCF= values that cause IMS to invoke RACF for sign on security checking for static terminals are: RCF=T, RCF=S, RCF=A, RCF=B, and RCF=R. The RCF= values have not changed and continue to work the same way in IMS Version 9 as in prior IMS releases. What has changed in IMS Version 9 is that new non-SMU methods are available for identifying the static terminals that are required to sign on.

The sign on security enhancements for static terminals in IMS Version 9 are:

- A new execution parameter, SIGNON=ALL | SPECIFIC, in the DFSDCxxx member of IMS.PROCLIB, is used to specify whether all or a subset of the static terminals are required to sign on.
- Two new keywords, SIGNON and NOSIGNON, on the OPTIONS statement on the TYPE and TERMINAL macros define sign on security requirements for static terminals

The value specified for the new SIGNON= keyword in DFSDCxxx and the SIGNON and NOSIGNON keywords on the OPTIONS statements for the TYPE and TERMINAL macros provide replacements for the SMU generation process where)(SIGN and STERM statements are used to define static terminal sign security requirements in SMU environments. An additional benefit of the new enhancement is that the signon offset list (DFSISSOx) matrix table does not have to be maintained or loaded during IMS restart.

6.6.6 New keywords on the OPTIONS statement

Sign on verification security can be defined for VTAM terminals via new options on the IMS system definition TYPE and TERMINAL macro statements. To implement sign on verification for statically defined VTAM terminals, set the OPTIONS= keyword to OPTIONS=SIGNON in the TYPE or TERMINAL system definition macro. No specification for OPTIONS= on the TERMINAL macro defaults to the value specified for OPTIONS= on the TYPE macro. A specification of OPTIONS=SIGNON or OPTIONS=NOSIGNON on a TERMINAL macro overrides the value set on the TYPE macro for that specific static terminal. The default for OPTIONS= on both the TYPE and TERMINAL macros is NOSIGNON.

Sign on verification for BTAM terminals can only be requested on IMS system definition TERMINAL macro statements using OPTIONS= SIGNON or OPTIONS=NOSIGNON. To implement sign on verification for statically defined BTAM terminals, set the OPTIONS= keyword to SIGNON in the system definition TERMINAL macro. The default for OPTIONS= in the TERMINAL macro is NOSIGNON.

SIGNON=ALL and SIGNON=SPECIFIC

Instead of specifying OPTIONS=SIGNON or OPTIONS=NOSIGNON in multiple TYPE and TERMINAL macro statements, all static terminals may be required to sign on by specifying a single parameter, SIGNON=ALL, in the DFSDCxxx member of IMS.PROCLIB. There are some exceptions when SIGNON=ALL is specified. The following devices are not required to sign on when SIGNON=ALL is used unless the OPTION statement on the TYPE or TERMINAL macro is set to OPTIONS=SIGNON:

- LU6.1 devices
- 3284/3286 devices
- SLU1 printer-only devices
- Master terminals

Alternatively, a specification of SIGNON=SPECIFIC in DFSDCxxx indicates that individual static terminals might be required to sign on. SIGNON=SPECIFIC is the default value for the SIGNON= keyword in the DFSDCxxx member. The static terminals which are required to sign on are identified in either:

- System definition TYPE or TERMINAL macro statements with the OPTIONS=SIGNON parameter
- By SMU generation input)(SIGN and STERM statements

6.6.7 Static terminal sign on security migration considerations

The use of SMU to identify and enforce sign on for static terminals may be eliminated in IMS Version 9. Migration to SAF/RACF user ID-based sign on security may be accomplished by performing all of the following tasks.

- Specify SIGNON=ALL or SIGNON=SPECIFIC in the DFSDCxxx member of IMS.PROCLIB.

- If SIGNON=SPECIFIC is used, identify the static terminals that are required to sign on by specifying OPTIONS=SIGNON in IMS system definition TYPE or TERMINAL macro statements.
- Specify values for RCF= and SGN= in the DFSPBxxx member of IMS.PROCLIB that cause IMS to invoke SAF/RACF for sign on security checking without loading the signon offset list (DFSISSOx) matrix table during IMS restart. These specifications are:
  - RCF=B or RCF=R
  - SGN=D, SGN=E, SGN=W, or SGN=X

The RCF and SGN values above may be used on IMS Version 7 and subsequent releases.

Performing the above tasks causes IMS to invoke SAF/RACF to enforce sign on for static and ETO terminals. RACF verifies that the user ID and password supplied on the /SIGN ON command are valid. Optionally, the sign on exit (DFSSGNX0) and or the sign on/off security exit (DFSCSGN0) may be used in lieu of SAF/RACF or in conjunction with RACF.

To further restrict specific user IDs or groups to signing on from specific IMS terminals, define terminal profiles in the RACF TERMINAL and GTERMNL resource classes to protect the terminals and authorize the user IDs/groups that are authorized to use the protected terminals to sign on.

6.7 LTERM-based terminal security

SMU provides LTERM-based terminal security for IMS commands and/or transactions entered from static terminals. LTERM-based terminal security restricts the entry of certain IMS commands and/or transaction codes to specific LTERMs. LTERM-based terminal security is activated by specifying one of the following:

- TERMNL=YES or TERMNL=FORCE on the SECURITY macro in IMS system definition macros. A description of these SECURITY macro specifications is provided in Table 6-2 on page 101, TERMNL values.
- TERMINAL on the /NRE or /ERE command issued to restart IMS.

The IMS commands and/or transactions that are restricted to entry from specific (or authorized) LTERMs are identified in input statements to the SMU generation process. The output from the SMU generation is a set of matrix tables which contain the LTERM restrictions. The matrix tables created for enforcement of LTERM-based terminal security are the:

- Terminal offset list
- Transaction offset list
- Communications terminal matrix

Figure 6-8 on page 105 illustrates how an LTERM-based entry restriction for a command is defined to the SMU generation process. It shows that the /LOCK command and TRANA may only be entered from the static terminal (in this case NODE5) which has LTERM5 assigned to it.

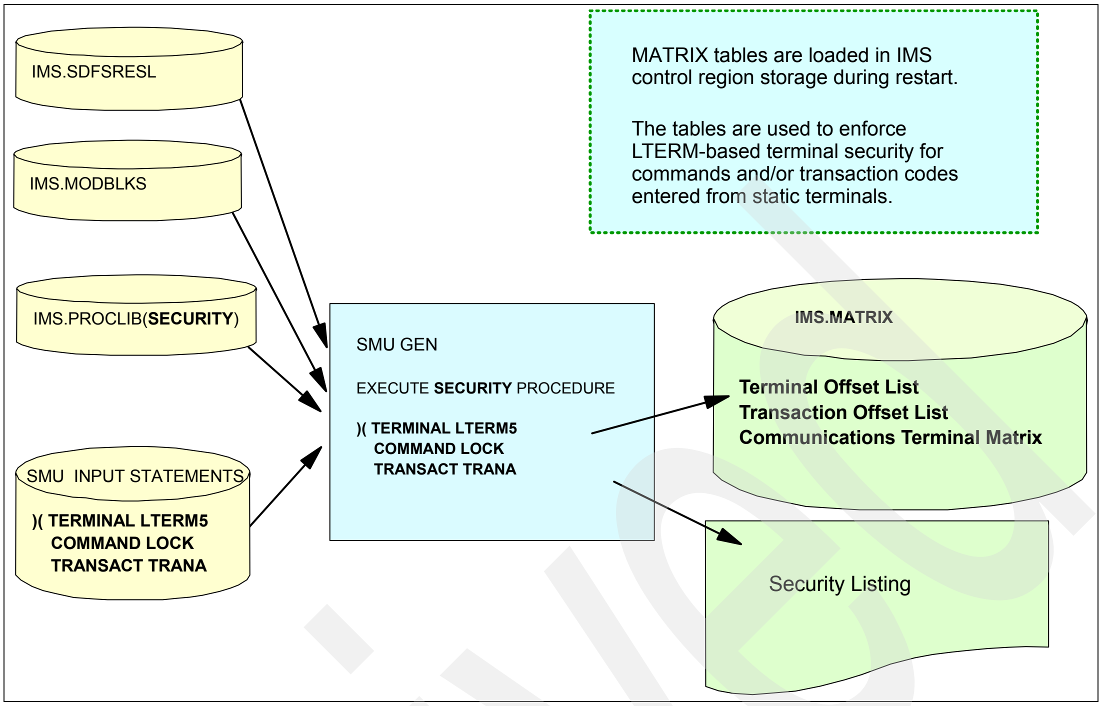

IMS.SDFSRESL
IMS.MODBLKS
IMS.PROCLIB(SECURITY)
SMU INPUT STATEMENTS
)(TERMINAL LTERM5
COMMAND LOCK
TRANSACT TRANA
SMU GEN
EXECUTE SECURITY PROCEDURE
)(TERMINAL LTERM5
COMMAND LOCK
TRANSACT TRANA
MATRIX tables are loaded in IMS control region storage during restart.
The tables are used to enforce LTERM-based terminal security for commands and/or transaction codes entered from static terminals.
IMS.MATRIX
Terminal Offset List
Transaction Offset List
Communications Terminal Matrix
Security Listing

Figure 6-8 Generating LTERM-based security

If LTERM-based terminal security is activated, IMS loads the matrix tables from the active IMS.MATRIX data set into control region storage during restart. When an IMS command or transaction code is entered from a static terminal, IMS invokes SMU modules to determine if LTERM-based entry restrictions exist for the command or transaction code. SMU modules use the matrix tables loaded during restart to enforce the LTERM-based terminal entry restrictions for commands and transactions entered from static terminals.

6.7.1 Migration considerations for LTERM-based terminal security

IMS Version 9 does not provide a SAF/RACF replacement for LTERM-based terminal security. As with prior IMS versions, SAF/RACF user ID-based command authorization and transaction authorization security checking is provided. IMS users should migrate LTERM-based terminal security to SAF/RACF user ID-based command and transaction authorization security checking. This may be accomplished by performing the following tasks:

- Define command profiles to protect IMS commands in the CIMS, DIMS, or equivalent installation-defined RACF resource classes.
- Define transaction profiles to protect IMS transactions in the TIMS, GIMS, or equivalent installation-define RACF resource classes.
- Authorize the RACF user IDs, or preferably the RACF groups, to the appropriate profiles.
- Activate the RACF resource classes (CIMS, DIMS, TIMS, GIMS, or equivalent installation-defined classes) if they are not already activated.
- If the current specification on the SECURITY macro in IMS system definition macros is set to TERMNL=FORCE, change the value to specify TERMNL=NO and regenerate IMS.
- If the current specification on the SECURITY macro in IMS system definition macros is set to TERMNL=YES, add the NOTERMINAL keyword to the /NRE or /ERE command issued to restart IMS.

- Specify a value for the RCF= keyword in IMS execution parameter that causes IMS to invoke SAF/RACF for command and/or transaction authorization security checking. Any values other than RCF=N and RCF=C may be used.

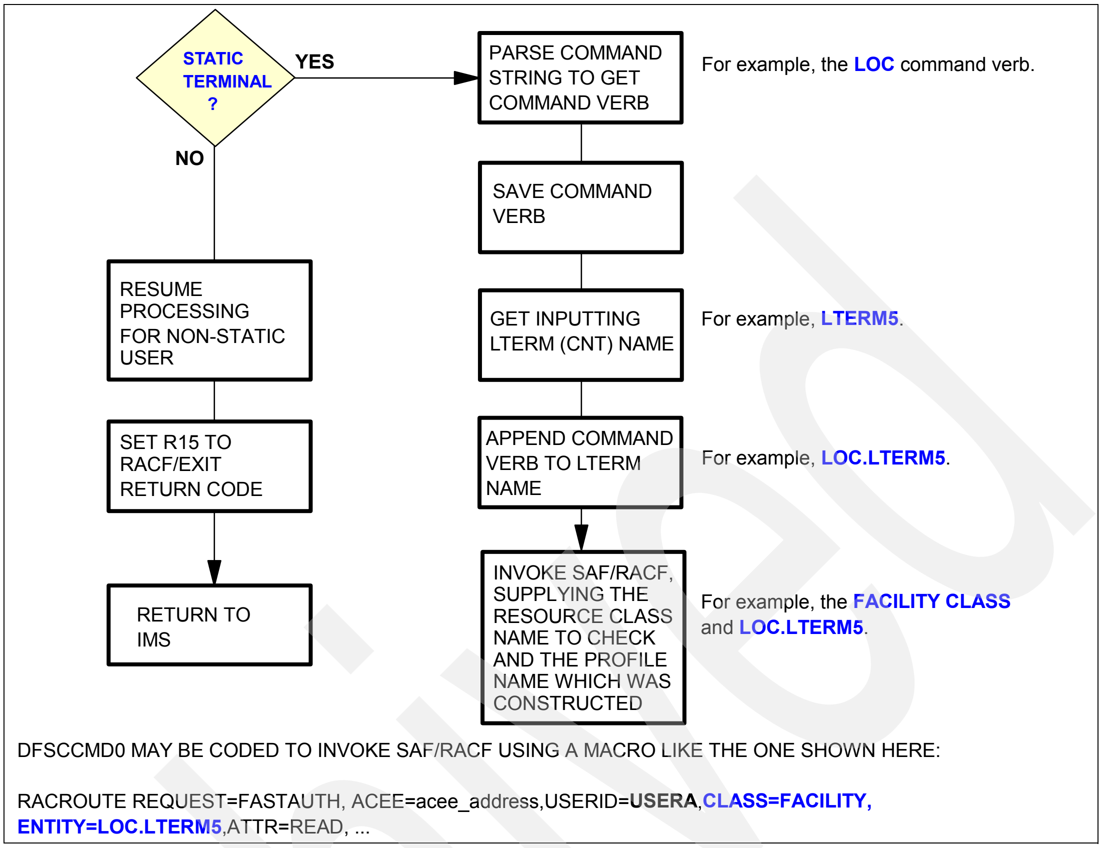
If LTERM-based terminal security is still required for certain commands and/or transaction codes entered from static terminals, there may be a number of ways to restrict terminal input to specific LTERMs.

6.7.2 IMS commands

One way to restrict IMS commands to specific LTERMs when the commands are entered from static terminals is to use SAF/RACF in conjunction with the command authorization exit routine (DFSCCMD0). Normal RACF command authorization checking, based on user ID/group, may be used. When both RACF and the exit are used to provide command security, DFSCCMD0 is invoked after SAF/RACF. The parameter list passed to the exit contains the SAF and RACF reason and return codes.

Figure 6-9 on page 107 shows an example of the DFSCCMD0 exit logic flow. The sample DFSCCMD0 exit is coded to determine if command input is entered from a STATIC or ETO terminal. The sample exit branches to the label 'STATIC' when the command input is from a static terminal. If DFSCCMD0 determines that RACF authorized the user ID to the command (or that the command was not protected) and the command is from a static terminal, the exit could be coded to re invoke SAF/RACF to check a RACF resource class to determine if the command can be entered from the inputting LTERM.

For example, a FACILITY class profile or a user-defined installation resource class could be used to determine if the command and static LTERM combination are valid. The exit could obtain the name of the static LTERM used to enter the command. Next, DFSCCMD0 could be coded to construct a RACF profile name from a combination of the command verb and the inputting LTERM name. Finally, the exit could invoke SAF/RACF to determine if the user ID/group is authorized to the command-LTERM profile.


STATIC TERMINAL ?
YES
NO
PARSE COMMAND STRING TO GET COMMAND VERB
For example, the LOC command verb.
SAVE COMMAND VERB
RESUME PROCESSING FOR NON-STATIC USER
GET INPUTTING LTERM (CNT) NAME
For example, LTERM5.
SET R15 TO RACF/EXIT RETURN CODE
APPEND COMMAND VERB TO LTERM NAME
For example, LOC.LTERM5.
RETURN TO IMS
INVOKE SAF/RACF, SUPPLYING THE RESOURCE CLASS NAME TO CHECK AND THE PROFILE NAME WHICH WAS CONSTRUCTED
For example, the FACILITY CLASS and LOC.LTERM5.

DFSCCMD0 MAY BE CODED TO INVOKE SAF/RACF USING A MACRO LIKE THE ONE SHOWN HERE:

```
RACROUTE REQUEST=FASTAUTH, ACEE=acee_address,USERID=USERA,CLASS=FACILITY,
ENTITY=LOC.LTERM5,ATTR=READ, ...
```

Figure 6-9 Sample DFSCCMD0 flow

Security administrators can define RACF profiles in the FACILITY resource class (or an installation-defined resource class) to protect the command-LTERM combination. Example 6-5 shows sample RACF commands to create profiles in the FACILITY class for the /LOCK command entered from static LTERM5. In the example, USERA must be authorized to command-LTERM combination to successfully enter the /LOCK command from LTERM5.

Example 6-5 Sample class profile

```
RDEFINE FACILITY LOC.LTERM5 UACC(NONE)
PERMIT LOC.LTERM5 CLASS(FACILITY) ID(USERA) ACCESS(READ)
```

Finally, prior to returning control to IMS, DFSCCMD0 can set register 15 to a return code that indicates the result of the SAF/RACF check:

- RACF RC=0

  The user ID (for example, USERA) is authorized to the command-LTERM combination
- RACF RC=4

  The command-LTERM combination is not defined to RACF. By default, IMS allows the command from the LTERM.
- RACF RC=8

  The user ID (for example, USERA) is not authorized to the command-LTERM combination

There are several other ways to accomplish the same thing using SAF/RACF and DFSCCMD0.

An advantage of this type of implementation is that it does not have to be limited to commands entered from static terminals. The scheme could work for restricting the entry of certain commands entered from ETO terminals as well.

6.7.3 IMS transactions

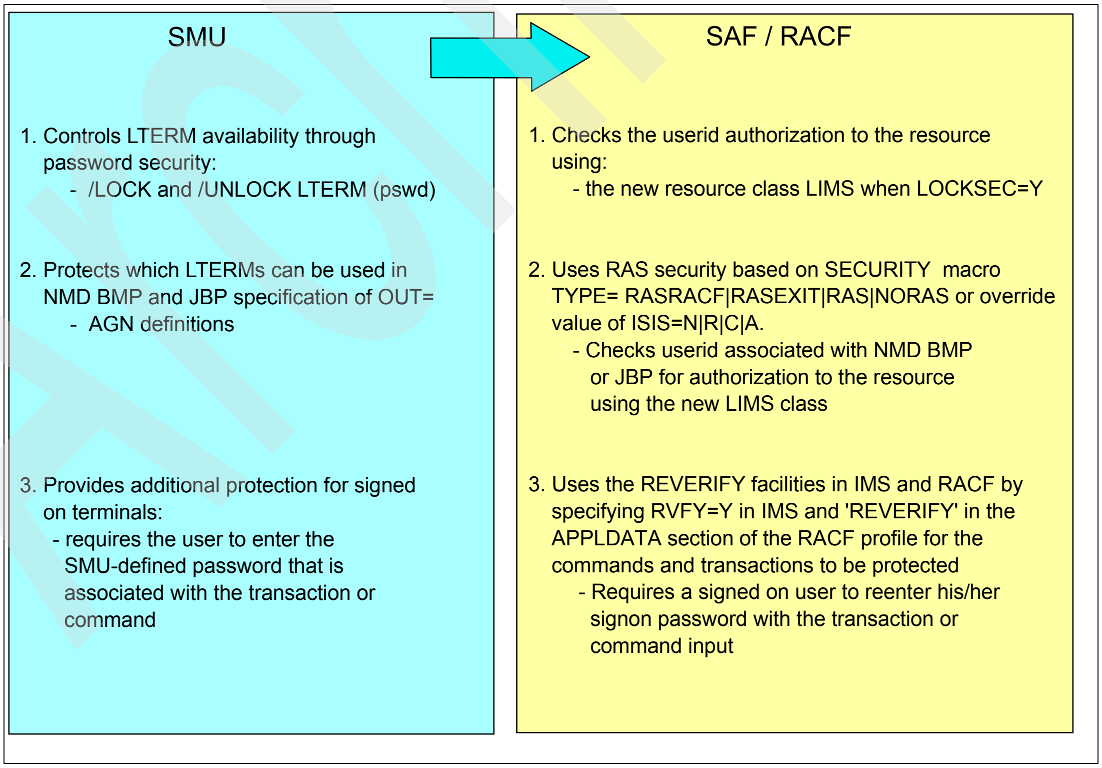
A similar scheme as the one described for restricting IMS commands to entry from specific static LTERMs can be used for IMS transactions entered from static terminals. The exception is that the transaction authorization exit (DFSCTRN0) is used instead of DFSCCMD0.

One difference in exit processing for transactions is that if RACF denies access (RACF RC=8) to the inputting user ID (for example USERA), DFSCTRN0 is not invoked. This is as it should be because if the user ID is not authorized to the transaction, additional security checking is unnecessary.

An additional consideration for transactions is that if the security reverification exit (DFSCTSE0) exists, it is invoked for transactions requested by DL/I CHNG and AUTH calls. Application programs are already coded to make the final decision for transactions requested by AUTH calls, so the major consideration here is for transactions requested by CHNG calls. If either SAF/RACF or DFSCTRN0 denied authorization to a transaction-LTERM combination, DFSCTSE0 should also be coded to deny authorization.

6.7.4 Migration considerations for LTERM-based security

Figure 6-10 show the considerations when migrating LTERM-based security from the use of SMU to the use of SAF/RACF. The first three areas show specific SAF/RACF solutions to assist in LTERM migration considerations.



SMU

1. Controls LTERM availability through password security:
   - /LOCK and /UNLOCK LTERM (pswd)

2. Protects which LTERMs can be used in NMD BMP and JBP specification of OUT=
   - AGN definitions

3. Provides additional protection for signed on terminals:
   - requires the user to enter the SMU-defined password that is associated with the transaction or command

SAF / RACF

1. Checks the userid authorization to the resource using:
   - the new resource class LIMS when LOCKSEC=Y

2. Uses RAS security based on SECURITY macro TYPE= RASRACF|RASEXIT|RAS|NORAS or override value of ISIS=N|R|C|A.
   - Checks userid associated with NMD BMP or JBP for authorization to the resource using the new LIMS class

3. Uses the REVERIFY facilities in IMS and RACF by specifying RVFY=Y in IMS and 'REVERIFY' in the APPLDATA section of the RACF profile for the commands and transactions to be protected
   - Requires a signed on user to reenter his/her signon password with the transaction or command input

Figure 6-10 SMU SAF considerations

Figure 6-11 on page 110 shows two additional areas of concern. These two areas in Figure 6-11 on page 110 are a little more complex since there is no direct SAF equivalent.

To migrate LTERM-based command security, consider using RACF in conjunction with the command authorization exit (DFSCCMD0) routine. Normal RACF command authorization checking based on user ID/group may be used. If it exists, the command authorization exit DFSCCMD0 is always invoked after RACF for commands entered from IMS terminals. The sample DFSCCMD0 exit that was described in the previous section, is coded to determine if a terminal is ETO or STATIC. If DFSCCMD0 determines that the command is from a static terminal, rather than accept the command, the exit could be changed to re invoke RACF to do one of the following:

- Check a special user-defined RACF profile.

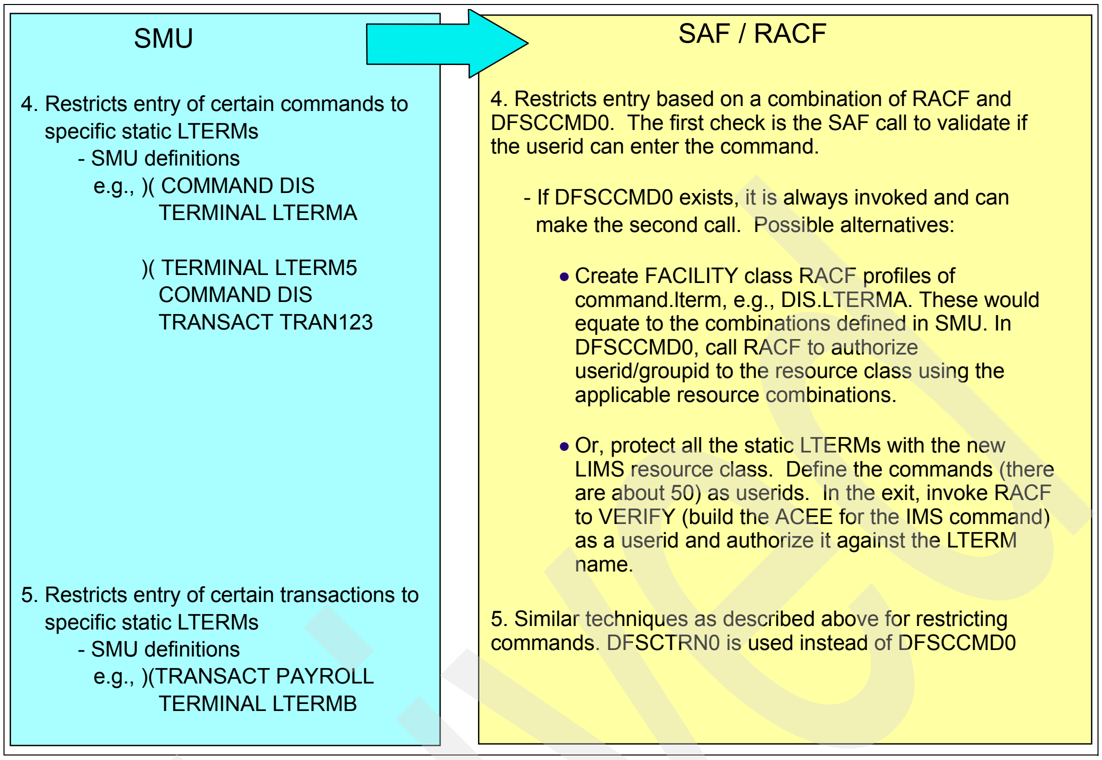
  For example, the exit could obtain the name of the static LTERM used to enter the command and construct a RACF profile name from a combination of the command verb and inputting LTERM name. Finally, the exit could invoke RACF to determine if the user ID/group is authorized to the command/LTERM profile.

- Invoke the command as a user ID.

  This alternative approach would be to RACF protect all the static LTERMS that were defined in SMU using the new RACF LIMS resource class. Additionally, since there are only about 50 commands, they could be defined to RACF as user IDs. DFSCCMD0 could then verify (build the ACEE) for the IMS command as a user ID and authorize it against the LTERM name.

Similar techniques as those described above for restricting commands could be used for restricting transactions. The difference would be that the transaction authorization exit (DFSCTRN0) would be used instead of DFSCCMD0. Note, however, that DFSCTRN0 is not called for any further security checking if the user ID is not authorized to the transaction. As with the command examples, you could also do all of the following to accomplish the same thing:

- Protect all of the static LTERMs in the new LIMS resource class provided by IMS Version 9
- Define the affected IMS transactions as RACF user IDs
- Use RACF's VLF ACEE cache so the transaction ACEEs are always in storage to eliminate I/O to the RACF DB
- Invoke RACF to VERIFY (build the ACEE) for the IMS transaction name as user ID if one did not already exist.



SMU

4. Restricts entry of certain commands to specific static LTERMs
   - SMU definitions
     e.g.,)(COMMAND DIS
        TERMINAL LTERMA

)(TERMINAL LTERM5
        COMMAND DIS
        TRANSACT TRAN123

5. Restricts entry of certain transactions to specific static LTERMs
   - SMU definitions
     e.g.,)(TRANSACT PAYROLL
        TERMINAL LTERMB

SAF / RACF

4. Restricts entry based on a combination of RACF and DFSCCMD0. The first check is the SAF call to validate if the userid can enter the command.

- If DFSCCMD0 exists, it is always invoked and can make the second call. Possible alternatives:

  - Create FACILITY class RACF profiles of command.lterm, e.g., DIS.LTERMA. These would equate to the combinations defined in SMU. In DFSCCMD0, call RACF to authorize userid/groupid to the resource class using the applicable resource combinations.

  - Or, protect all the static LTERMs with the new LIMS resource class. Define the commands (there are about 50) as userids. In the exit, invoke RACF to VERIFY (build the ACEE for the IMS command) as a userid and authorize it against the LTERM name.

5. Similar techniques as described above for restricting commands. DFSCTRN0 is used instead of DFSCCMD0

Figure 6-11 SMU SAF LTERM based

6.8 Additional migration considerations

There is a software dependency on the RACF product to include the four new RACF security classes as default classes. If the new classes have not yet been added to the RACF product, they must be added to the installation-defined class descriptor table (CDT). Macro ICHERCDE is used to define a new class.

The procedure for adding classes to this table is described in “Adding Installation-Defined Classes” in the *z/OS V1R5.0 Security Server RACF System Programmer's Guide,* SA22-7681. These classes are provided with z/OS 1.6 Security Server Defined Resource Classes.

RACF supplies jobs to update the CDT in SYS1.SAMPLIB. These new classes should also be defined to the RACF router table, ICHRFR01. The member RACTABLE has a sample for updating the router table. Example 6-6 shows how the macros can be modified to define the new IMS classes IIMS, JIMS, LIMS and MIMS.

Example 6-6 A sample update of the CDT for the new IMS classes

```
IIMS     CSECT
         ICHERCDE CLASS=IIMS,GROUP=JIMS,ID=128,RACLIST=DISALLOWED,POSIT=575,            X
               OPER=NO,FIRST=ALPHA,OTHER=ALPHANUM,DFTUACC=NONE
         ICHERCDE CLASS=JIMS,MEMBER=IIMS,ID=129,RACLIST=DISALLOWED,POSIT=575,           X
               OPER=NO,FIRST=ALPHA,OTHER=ALPHANUM,DFTUACC=NONE
         ICHERCDE CLASS=LIMS,GROUP=MIMS,ID=130,RACLIST=DISALLOWED,POSIT=576,            X
               OPER=NO,FIRST=ALPHANUM,OTHER=ALPHANUM,DFTUACC=NONE
         ICHERCDE CLASS=MIMS,MEMBER=LIMS,ID=131,RACLIST=DISALLOWED,POSIT=576,           X
               OPER=NO,FIRST=ALPHANUM,OTHER=ALPHANUM,DFTUACC=NONE
         ICHERCDE
         END

... (left out the JCL statements from here...

ICHRFR01 CSECT
         ICHRFRTB CLASS=IIMS,ACTION=RACF
         ICHRFRTB CLASS=JIMS,ACTION=RACF
         ICHRFRTB CLASS=LIMS,ACTION=RACF
         ICHRFRTB CLASS=MIMS,ACTION=RACF
         ICHRFRTB TYPE=END
         END   ICHRFR01
...
```

For environments that migrate to IMS Version 9 and choose to start out using only SMU security, a specification of RCF=N is supported. When migrating to the use of SAF begins, the RCF parameter will need to be specified appropriately. Switching from a value of “N” to anything else requires that the IMS system be cold started.

If an IMSplex contains IMS Version 9 systems along with prior release systems, the NORSCCC(MODBLKS) parameter in the DFSCGxxx PROCLIB member should be specified to disable a part of the resource consistency checking. By including the specification, the MODBLKS and MATRIX data set names are not checked for consistency which means that the IMSs in the IMSplex do not need to define the same data sets for the MODBLKS and MATRIX library. The MATRIX data sets are still required in IMS Version 9, but they may be empty if SMU security is not used.

Customers could possibly have a conflict between transaction names for AOI defined as users to RACF and current user IDs for users defined to RACF. A similar conflict could exist for MSNAMEs used for MSC security and user's user IDs.

From an operations point of view, notice that messages DFS066 (password security violation) and DFS067 (terminal security violation) are replaced by RACF ICH408I messages.

6.8.1 Performance considerations

The use of RACF security may be slower than checking a SMU security matrix table, however the frequency of the type of security checking being made available is not that high, so the performance impact should not be significant.

When taking advantage of MSC link security, several options are provided to control the level of security. The choices impact the amount of checking, and overhead that is used. In the link receive environment, the calls to SAF and DFSCTRN0 are limited to being made after the call to DFSMSCE0 instead of before and after, as in previous releases.

6.8.2 AOI considerations

There are three new return/reason codes for an ICMD call when AOI=CMD is specified:

- `0110/0054 - Command not authorized to RACF.`

  An AO application tried to issue an IMS command. Security checking determined that the command is not authorized to RACF as a user ID.

- `0110/0058 - Command not authorized to be issued by the transaction.`

  An AO application tried to issue an IMS command. Security checking determined that the command is not authorized to RACF to be issued by the transaction.

- `0110/005C - DFSCCMD0 indicated command was not authorized to be issued by transaction.`

  An AO application tried to issue an IMS command. The command authorization exit routine (DFSCCMD0) indicated the command was not authorized to be issued by the transaction.

There are also three new return/reason codes for an ICMD call when AOI=TRAN is specified:

- `0110/0044 - Transaction not authorized to RACF.`
- `0110/0048 - Transaction not authorized to issue the command.`
- `0110/004C - DFSCCMD0 indicated transaction was not authorized to issue command.`

The explanations for these are similar to the ones for AOI=CMD with transaction replacing command in the text.

The X'10' log record (described by DFSLOG10) has four new error codes defined for the SCERROR field. The codes and their meanings are the following:

160	CMD USERID NOT DEFINED TO RACF
161	CMD USERID NOT AUTH TO COMMAND
162	CMD REJECTED BY DFSCCMD0
163	CMD FAILED, DFSCCMD0 UNAVAILABLE

6.8.3 User exit interface changes

The following changes have been made for the user exit interface:

- DFSRAS00

  This is the new resource access security exit routine that provides authorization of IMS resources to IMS dependent regions. It replaces DFSISIS0 for environments that use RAS instead of AGN security with SMU.

- DFSCCMD0

  The input parameter list supports two new values for the type of caller in CCMD_RQSTTYPE: CMD FOR TRANSACTION and ICMD FOR TRANSACTION. In both cases, the user ID field will also contain the transaction name, and CCMD_UIDIND will contain CCMD_UIDT ('T').

- DFSISIS0

  In prior releases, this exit was called the resource access security exit routine. In IMS Version 9 the exit has been renamed to application group name (AGN) security exit routine to avoid confusions with DFSRAS00. Its function will be the same.

- DFSMSCE0

  At the end of the main exit parameters section, the following fields have been added: user ID, group name and a user ID indicator. Additionally, in an existing field of the link receive routing parameters, MSLRFL3 (Link Receive Exit Flag 3), the exit can specify the level of authorization as being: security based on MSNAME, security based on CTL address space, security based on user ID, security based on Exit, or no link receive security. On entry to the DFSMSCE0 user exit for link receive, the MSLRFL3 field contains the system default value from the MSCSEC= startup parameter.

6.8.4 SMU to RACF migration check list

The following is a high-level migration checklist of tasks we discussed in this chapter:

- Translate AGN definitions to RACF

  - Add the new classes to RACF
  - Define new RAS parameters (SECURITY macro and ISIS= execution parameter)
  - Create DFSRAS00 to replace DFSISIS0
  - Review JCL for AGN= specifications

- For static terminals required to sign on
  - Specify SIGNON=ALL|SPECIFIC parameter in DFSDCxxx
  - Optionally, specify OPTIONS=SIGNON on applicable TYPE/TERMINAL macros
- Enable SAF support for TCO command authorization
  - TCORACF=Y and RCF=A|S|R|B
- Review AOI requirements
  - Specify AOI parameter on TRANSACT macro where needed
  - For TYPE 1 CMD security, additionally specify AOI1 = A|N|C|R|S
- Migrate /LOCK and /UNLOCK security
  - Specify LOCKSEC=Y in DFSDCxxx
- Review MSC requirements for link receive security
  - Specify use of SAF/DFSCTRN0 and level of authorization checking in the new MSCSEC parameter in DFSDCxxx
  - Modify DFSMSCE0 if needed
  - Synchronize RACF profiles on sending and destination systems
- Determine the need to change or write exit routines

7

Type-2 commands and enhanced command environment

In this chapter we introduce the IMS Version 9 enhancement for the use of type-2 commands and the enhanced command environment.

IMS Version 9 publications use new terminology to distinguish the types of IMS commands. Classic commands are called type-1 commands. IMSplex or enhanced commands are now called type-2 commands. Type-1 commands are generally preceded by a leading slash character, and they may be entered from any valid IMS command source. Type-2 commands may be entered only through the Operations Manager (OM) API. IMS Version 9 further enhances the type-2 commands to include database resources and introduces new commands for integrated HALDB online reorganization.

7.1 Type-2 command enhancements in IMS Version 9

IMS Version 9 improves the use of type-2 commands in two ways. First, the enhanced command environment does not require a Resource Manager (RM). This allows some installations to simplify their environments and procedures. Second, type-2 commands may be used for database resources. This allows users to consolidate operations through the use of type-2 commands. In other words the environment is simplified, procedures are easier and operations are consolidated.

All type-2 commands have a common syntax. The command begins with an action verb and resource type. Parameters are enclosed in parentheses following a keyword. This avoids any potential conflicts between keywords and resource names. There are five action verbs for these commands. IMSplex action verbs available with IMS Version 9 are the following:

- QUERY or QRY verb
- UPDATE or UPD verb
- INITIATE or INIT verb
- TERMINATE or TERM verb
- DELETE or DEL verb

Each verb may be followed by a resource type. Resource types OLREORG, DB, AREA, and DATAGRP are introduced in IMS Version 9. OLREORG indicates HALDB online reorganization. DB is used for databases. AREA is used for Fast Path DEDB areas. DATAGRP is used for database groups.

Type-2 commands are issued by using the Operations Manager (OM) API. This requires the implementation of the Common Service Layer (CSL). There must be at least one OM address space in the IMSplex and an SCI address space on each LPAR in the IMSplex. IMS Version 9 does not require an RM address space for support of type-2 commands.

IMS includes a TSO implementation of the API. This is the TSO single point of control (TSO SPOC). IMS also provides a sample REXX program which uses the API. The IMS Control Center is a tool which provides a GUI implementation of the API. The IMS Control Center is executed on a Windows® or UNIX® workstation.

The type-2 commands require the use of the OM API. They cannot be entered from IMS terminals or the system console. The IMS control region AOI user exits are not invoked for type-2 commands or their responses.

Figure 7-1 on page 117 illustrates a full CSL environment. Each system in the IMSplex contains an OM, SCI, and RM address space. There is SCI function in these address spaces, the IMS control regions, and Common Queue Server (CQS) address spaces, and any instances of DBRC. CQS is used to manage the resource structure in the Coupling Facility (CF).

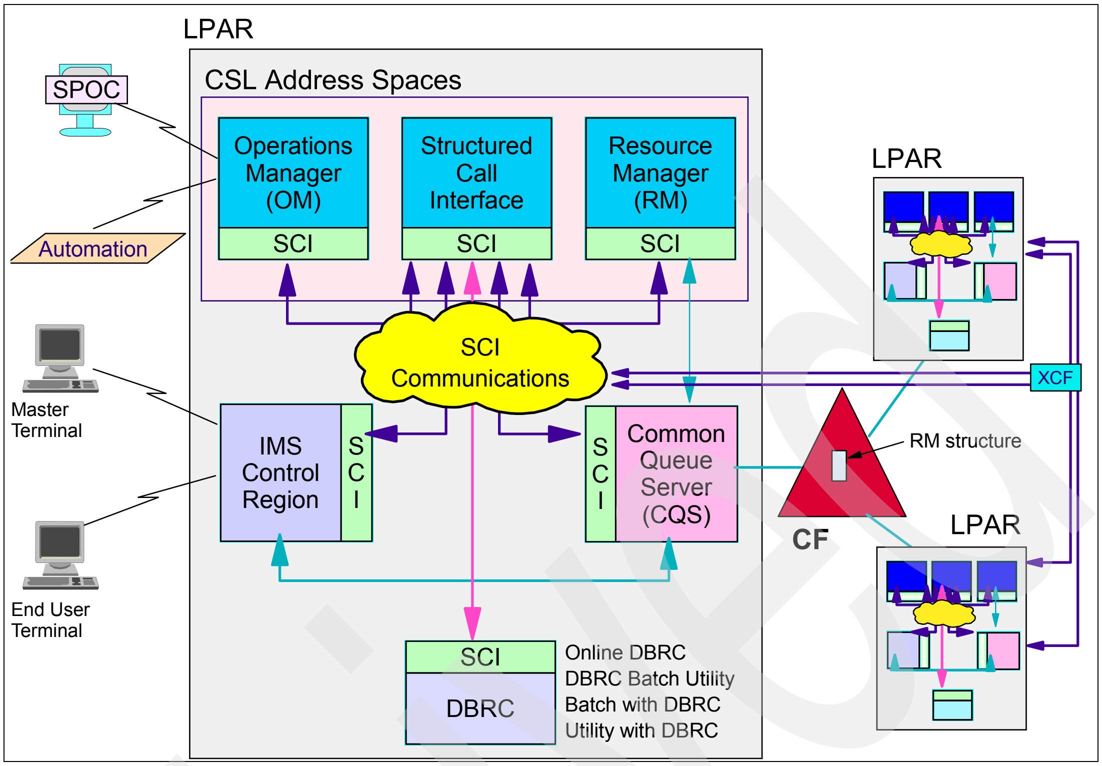


Figure 7-1 Full CSL configuration

Figure 7-14 on page 130 illustrates the minimal IMS Version 9 CSL environment required for type-2 commands. There is one OM address space and one Structured Call Interface (SCI) address space. If there were multiple systems (LPARs) in the IMSplex, each system would require an SCI address space, but only one OM address space would be required in the IMSplex.

Command clients

Type-2 commands are entered through command clients. They use the OM API for entering commands and receiving responses. Typically, they format the responses for presentation to users, although this is not required. For example, an automation package might issue commands, examine the responses, and take other actions based on the responses.

As mentioned previously, the TSO SPOC is distributed with IMS for ISPF users. The SPOC receives responses in XML format and formats them for presentation to the user. The IMS Control Center may be downloaded from the Web. It's a GUI command client for workstations and available with the IBM DB2 Universal Database™ (DB2 UDB) Administration Client, Version 8.2. Follow the links from the IMS home page for download details:

http://www.ibm.com/ims

TSO SPOC

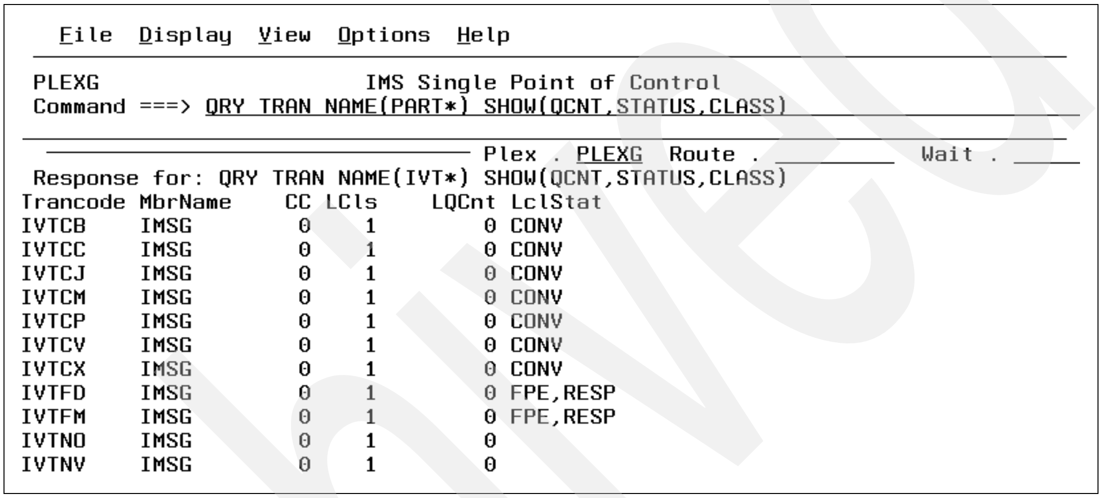
The TSO SPOC is an ISPF application. Users may set preferences and defaults, such as the IMS systems to which the commands will be sent. Users may also create groups and shortcuts for use in commands. For example, a set of IMS systems might be collected into a group so that only the group name is required to reference all of these IMS systems. Similarly, a set of characters, such as &DAR might be used as a shortcut for “DISPLAY ACTIVE REGIONS”.

The TSO SPOC may be used to enter both type-1 and type-2 commands. Type-1 commands are those classic commands which used a "/", such as /DISPLAY. Responses from both types of commands are displayed to the user. The user may search and sort this output when it is a response for a type-2 command.

The TSO SPOC saves previous commands and their responses. The user may recall these commands, edit them, and enter the modified commands. The TSO SPOC formats the returned XML responses from OM so it's readable by the user.

TSO SPOC command response

This Figure 7-2 shows a sample response screen. At the top there is an area for the entering of commands. The response to a previous command is shown below it.



```
   File  Display  View  Options  Help
 ---------------------------------------------------------------------------
  PLEXG                        IMS Single Point of Control
  Command ===> QRY TRAN NAME(PART*) SHOW(QCNT,STATUS,CLASS)
 ---------------------------------------------------------------------------
                                      Plex . PLEXG  Route . _________  Wait . _____
  Response for: QRY TRAN NAME(IVT*) SHOW(QCNT,STATUS,CLASS)
 Trancode MbrName    CC LCls    LQCnt LclStat
 IVTCB    IMSG        0    1        0 CONV
 IVTCC    IMSG        0    1        0 CONV
 IVTCJ    IMSG        0    1        0 CONV
 IVTCM    IMSG        0    1        0 CONV
 IVTCP    IMSG        0    1        0 CONV
 IVTCV    IMSG        0    1        0 CONV
 IVTCX    IMSG        0    1        0 CONV
 IVTFD    IMSG        0    1        0 FPE,RESP
 IVTFM    IMSG        0    1        0 FPE,RESP
 IVTNO    IMSG        0    1        0
 IVTNV    IMSG        0    1        0
```

Figure 7-2 TSO SPOC command response

IMS Control Center

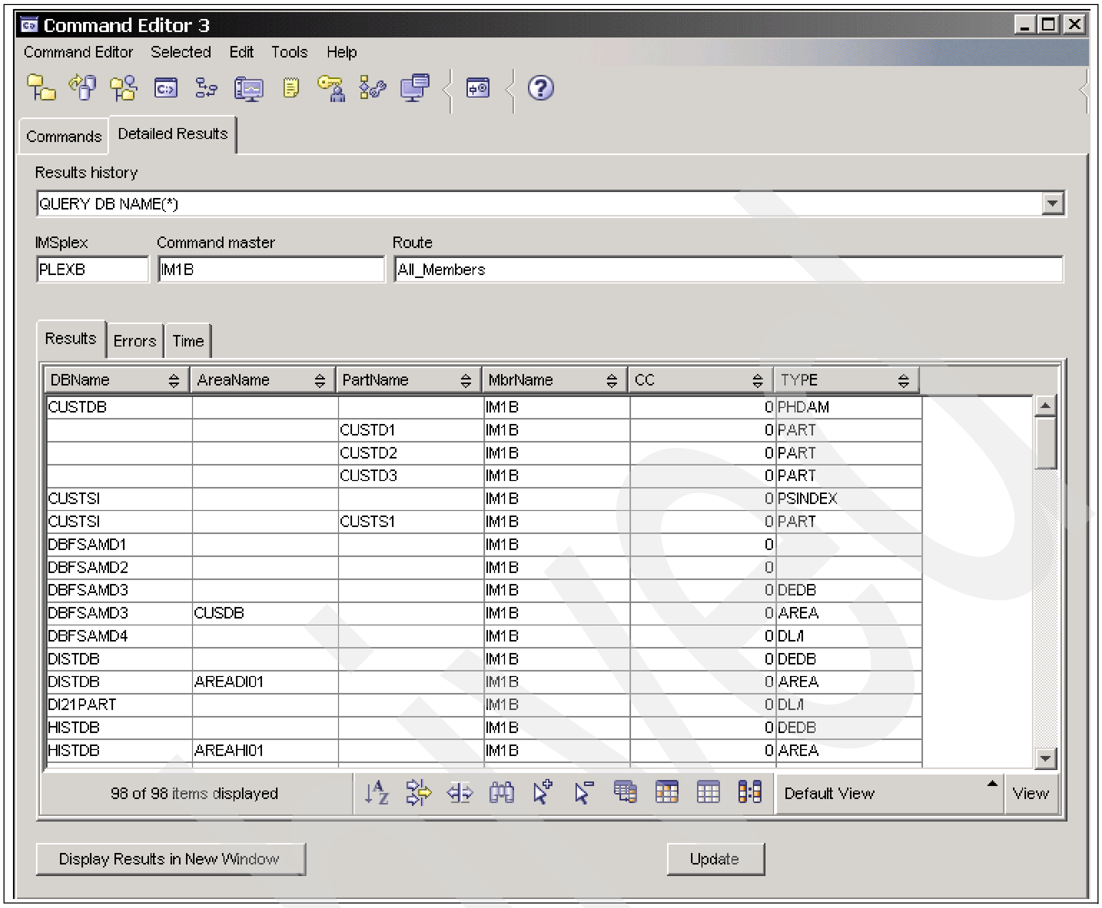
The IMS Control Center may be used to enter type-1 and type-2 commands through a GUI application on a workstation. The IMS Control Center is part of the IBM DB2 UDB Version 8 Control Center, available with the IBM DB2 Universal Database (DB2 UDB) Administration Client, Version 8.2. If you have a workstation which does not have DB2 UDB, you may download the DB2 UDB Administration Client from the Web. There is no charge for this function. Follow the IMS Control Center links from IMS home page:

    http://www.ibm.com/ims

The IMS Control Center uses TCP/IP to communicate with the z/OS system. IMS Connect or the integrated IMS Connect function of IMS Version 9 on the z/OS system is required for this TCP/IP connectivity. IMS Connect uses SCI to connect to the IMS OM address space.

IMS Control Center allows you to enter type-2 commands. These include commands introduced in IMS Version 8, such as those that query IMSplex members, and those introduced in IMS Version 9. Of course, the IMS Control Center also allows you to enter the classic IMS commands (type-1 commands).

Figure 7-3 on page 119 show you a sample IMS Control Center screen, you may sort the rows by any column. In Appendix D, "IMS Control Center" on page 299 we show you how to connect the command center to your IMS Connect.



Results history: QUERY DB NAME(*)

IMSplex: PLEXB

Command master: IM1B

Route: All_Members

DBName	AreaName	PartName	MbrName	CC	TYPE
CUSTDB			IM1B	0	PHDAM
		CUSTD1	IM1B	0	PART
		CUSTD2	IM1B	0	PART
		CUSTD3	IM1B	0	PART
CUSTSI			IM1B	0	PSINDEX
CUSTSI		CUSTS1	IM1B	0	PART
DBFSAMD1			IM1B	0	
DBFSAMD2			IM1B	0	
DBFSAMD3			IM1B	0	DEDB
DBFSAMD3	CUSDB		IM1B	0	AREA
DBFSAMD4			IM1B	0	DL/I
DISTDB			IM1B	0	DEDB
DISTDB	AREADI01		IM1B	0	AREA
DI21PART			IM1B	0	DL/I
HISTDB			IM1B	0	DEDB
HISTDB	AREAHI01		IM1B	0	AREA

98 of 98 items displayed

Figure 7-3 IMS Control Center sample response screen

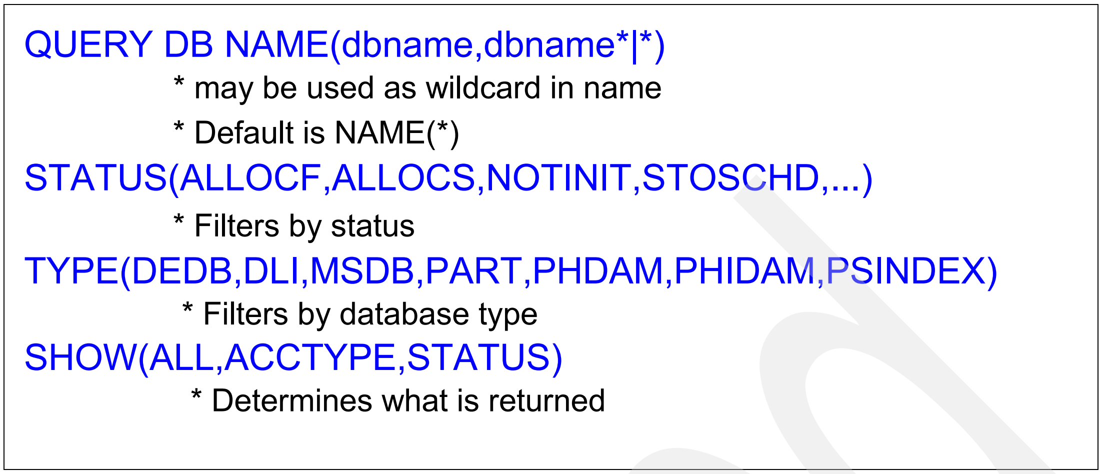
7.2 Type-2 database commands

IMS Version 9 adds database resources to the type-2 commands. In IMS Version 8, you could only enter classic database commands (type-1 commands), such as /DBR DB through the OM API: there was no type-2 equivalent command. IMS Version 9 allows you to enter DB commands using the type-2 format. The new type-2 database commands are the following:

- QUERY or QRY verb with resource types DB and AREA
- UPDATE or UPD verb with resource types DB, AREA, and DATAGRP

7.2.1 QUERY DB command

Figure 7-4 on page 120 shows the format of the QUERY DB command.



```
QUERY DB NAME(dbname,dbname*|*)
        * may be used as wildcard in name
        * Default is NAME(*)
STATUS(ALLOCF,ALLOCS,NOTINIT,STOSCHD,...)
        * Filters by status
TYPE(DEDB,DLI,MSDB,PART,PHDAM,PHIDAM,PSINDEX)
         * Filters by database type
SHOW(ALL,ACCTYPE,STATUS)
          * Determines what is returned
```

Figure 7-4 The format of QUERY DB command

The QUERY DB or QRY DB command specifies either a database or a generic database name. Generic names use * as a wildcard. NAME(*) is equivalent to the use of ALL in a type-1 IMS command. NAME(*) is also the default.

If the database specified is a HALDB master, the master database and all its partitions are returned. If the database specified is a HALDB partition, the partition database and the master database are returned. If the database is a DEDB, the database and all of its areas are returned.

The values that may be used with the STATUS keyword are:

ALLOCF, ALLOCS, BACKOUT, EEQE, LOCK, NOTINIT, NOTOPEN, OFR, OLR, OPEN, RECALL, RECOV, RNL, STOSCHD, and STOUPDS.

Most of these are the attributes which may be displayed with a `/DIS DB` command. Others are unique to QRY DB. OLR indicates that an online reorganization is active. STOSCHD indicates that the database is stopped for scheduling. This could be the result of a /STOP DB command or an UPD DB STOP(SCHD) command. STOUPDS indicates that the database is stopped for updates. This could be the result of a /DBD DB command or an UPD DB STOP(UPDATES) command. RNL indicates that the randomizer is not loaded for the DEDB.

SHOW determines what is returned by the command processing. Database name, IMS member name, database type, DEDB area names, HALDB partition names, and completion code are always returned.

QRY DB and equivalent type-1 commands

In Table 7-1 on page 121 we show a few examples of QRY DB commands and the equivalent type-1 commands.

Since NAME(*) is the default for QRY DB commands, the QRY DB STATUS(ALLOCF NOTINIT) is equivalent to a QRY DB NAME(*) STATUS(ALLOCF NOTINIT) command. It returns only those databases which either have had an allocation failure or have failed database directory (DDIR) initialization.

There is no equivalent of the QRY DB NAME(AB*) command. Wildcard commands cannot be used with /DIS DB commands.

The QRY DB NAME(AB34) SHOW(STATUS) command returns a subset of the data returned by a /DIS DB AB34 command because the SHOW keyword is used to limit the data to only the status and the information which is always returned. Database name, IMS member name, database type, HALDB partition names, and completion code are always returned.

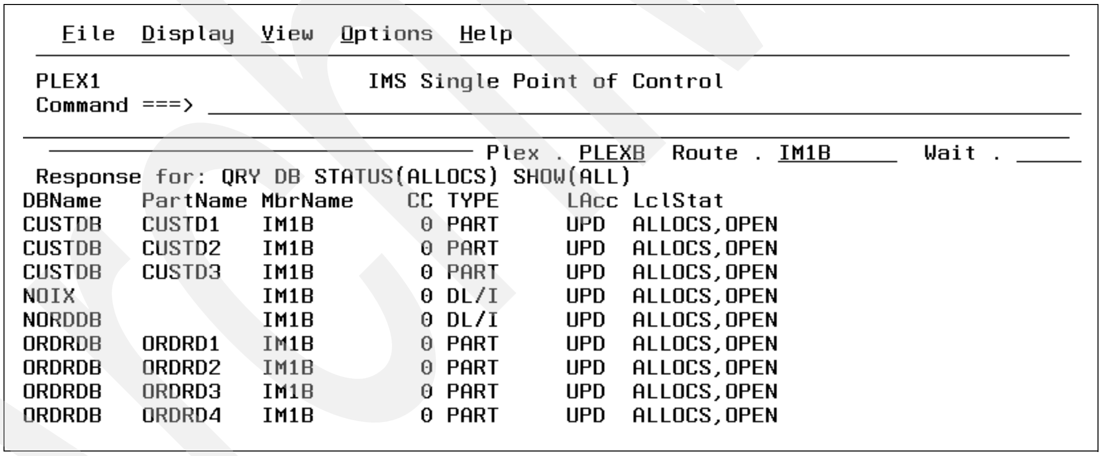
Table 7-1 QRY DB examples

QRY DB Command	Equivalent Type-1 Command
QRY DB NAME(*)	/DIS DB ALL
QRY DB NAME(AB34 AB56)	/DIS DB AB34 AB56
QRY DB STATUS(ALLOCF NOTINIT)	/DIS DB ALLOCF NOTINIT
QRY DB NAME(AB*)	No equivalent; returns only databases beginning with 'AB'
QRY DB NAME(AB34) SHOW(STATUS)	Subset of data from /DIS DB AB34

QRY DB output examples

These are two examples of the responses seen when using the TSO SPOC with QRY DB commands. 'PartName' and 'AreaName' columns do not appear if there is no data for the column.

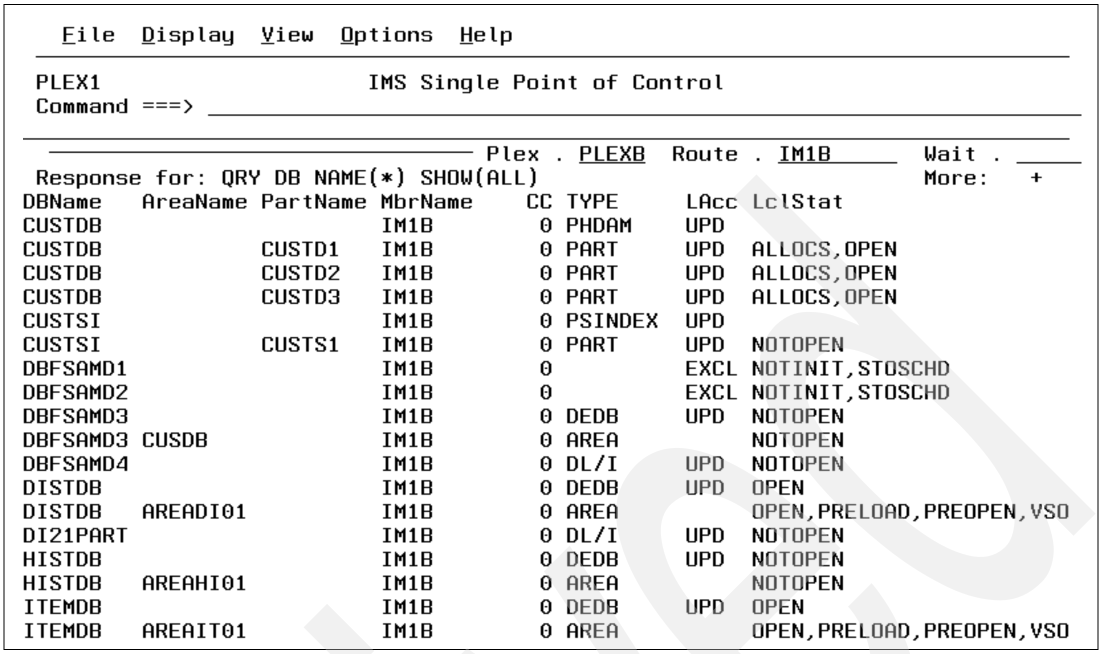
In Figure 7-5, the 'PartName' column appears because database CUSTDB and ORDRDB are HALDBs. The response includes partition names for these databases.



```
 PLEX1                     IMS Single Point of Control
 Command ===> ____________________________________________________________

 ______________________________________ Plex . PLEXB  Route . IM1B      Wait . _____
 Response for: QRY DB STATUS(ALLOCS) SHOW(ALL)
DBName    PartName MbrName    CC TYPE     LAcc LclStat
CUSTDB    CUSTD1   IM1B        0 PART     UPD  ALLOCS,OPEN
CUSTDB    CUSTD2   IM1B        0 PART     UPD  ALLOCS,OPEN
CUSTDB    CUSTD3   IM1B        0 PART     UPD  ALLOCS,OPEN
NOIX               IM1B        0 DL/I     UPD  ALLOCS,OPEN
NORDDB             IM1B        0 DL/I     UPD  ALLOCS,OPEN
ORDRDB    ORDRD1   IM1B        0 PART     UPD  ALLOCS,OPEN
ORDRDB    ORDRD2   IM1B        0 PART     UPD  ALLOCS,OPEN
ORDRDB    ORDRD3   IM1B        0 PART     UPD  ALLOCS,OPEN
ORDRDB    ORDRD4   IM1B        0 PART     UPD  ALLOCS,OPEN
```

Figure 7-5 QRY DB STATUS(ALLOCS) SHOW(ALL) issued from TSO SPOC

In Figure 7-6 on page 122, the 'AreaName' column appears because there are some DEDBs displayed. The response includes area names for these databases.



```
  File  Display  View  Options  Help
 ------------------------------------------------------------------------------
 PLEX1                       IMS Single Point of Control
 Command ===> ___________________________________________________________________
 ------------------------------------------------------------------------------
  ------------------------------- Plex . PLEXB  Route . IM1B      Wait . ____
 Response for: QRY DB NAME(*) SHOW(ALL)                          More:   +
DBName   AreaName PartName MbrName    CC TYPE     LAcc LclStat
CUSTDB                     IM1B        0 PHDAM    UPD
CUSTDB            CUSTD1   IM1B        0 PART     UPD  ALLOCS,OPEN
CUSTDB            CUSTD2   IM1B        0 PART     UPD  ALLOCS,OPEN
CUSTDB            CUSTD3   IM1B        0 PART     UPD  ALLOCS,OPEN
CUSTSI                     IM1B        0 PSINDEX  UPD
CUSTSI            CUSTS1   IM1B        0 PART     UPD  NOTOPEN
DBFSAMD1                   IM1B        0          EXCL NOTINIT,STOSCHD
DBFSAMD2                   IM1B        0          EXCL NOTINIT,STOSCHD
DBFSAMD3                   IM1B        0 DEDB     UPD  NOTOPEN
DBFSAMD3 CUSDB             IM1B        0 AREA          NOTOPEN
DBFSAMD4                   IM1B        0 DL/I     UPD  NOTOPEN
DISTDB                     IM1B        0 DEDB     UPD  OPEN
DISTDB   AREADI01          IM1B        0 AREA          OPEN,PRELOAD,PREOPEN,VSO
DI21PART                   IM1B        0 DL/I     UPD  NOTOPEN
HISTDB                     IM1B        0 DEDB     UPD  NOTOPEN
HISTDB   AREAHI01          IM1B        0 AREA          NOTOPEN
ITEMDB                     IM1B        0 DEDB     UPD  OPEN
ITEMDB   AREAIT01          IM1B        0 AREA          OPEN,PRELOAD,PREOPEN,VSO
```

Figure 7-6 QRY DB NAME(*) SHOW ALL issued from TSO SPOC

7.2.2 QUERY AREA command

The QUERY AREA command is similar to the QUERY DB command. Figure 7-7 shows the format of the QUERY AREA command.

QUERY AREA NAME(areaname,areaname*|*)

* may be used as wildcard in name

* default is NAME(*)

STATUS(EEQE,IC,MAS,NOTOPEN,OPEN,PRELOAD,...)

* filters by status

OPTION(REFRESH)

* returns refreshed IOVF statistics to the master command processor

SHOW(ALL,CI,STATUS,UTILITY)

* determines what is returned

Figure 7-7 The format of QUERY AREA command

The values that may be used with STATUS are:

EEQE, IC, MAS, NOTOPEN, OFR, OPEN, PRELOAD, PREOPEN, RECALL, RECOVERY, RECOVNEEDED, SAS, SHARED, STOPPED, UTIL, and VSO.

SHOW determines what is returned by the command processing. Area name, DEDB name, IMS member name, and completion code are always returned. ADS information is returned if available. CI, STATUS, and UTILITY may be used to limit the other information returned. If SHOW is not specified, only a list of resource names is returned.

QRY AREA and equivalent type-1 commands

In Table 7-2 these are examples of QRY AREA commands and the equivalent type-1 commands. There is no equivalent of the QRY AREA NAME(BX*) STATUS(RECALL) command. Wildcard commands cannot be used with /DIS AREA commands.

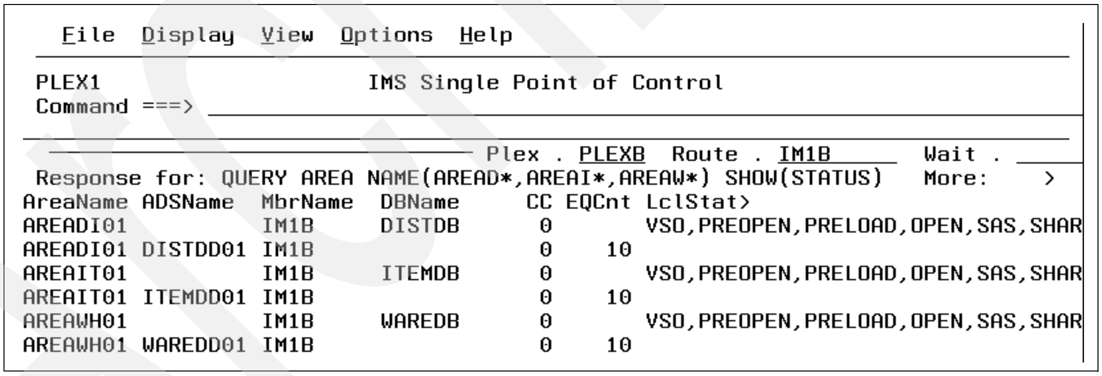
Table 7-2 QRY AREA examples

QRY AREA Command	Equivalent Type-1 Command
QRY AREA NAME(*) SHOW(ALL)	/DIS AREA ALL
QRY AREA NAME(BXJ21) OPTION(REFRESH)	/DIS AREA BXJ21 IOVF
QRY AREA NAME(*) STATUS(STOPPED)	/DIS AREA STOPPED
QRY AREA NAME(BX*) STATUS(RECALL)	No equivalent; Returns only areas beginning with 'BX' which are being recalled
QRY AREA NAME(BXJ21) SHOW(STATUS)	Subset of data from /DIS AREA BXJ21

QRY AREA output example

Figure 7-8 shows the response when using the TSO SPOC with QRY AREA command with SHOW(STATUS) for selected databases. The EQCnt column contains different data for areas and area data sets. For areas it is the number of EEQEs. For area data sets it is the number of EQEs remaining out of the ten available. If multiple area data sets are not used, EQEs are not used and the value will always be 10. The LclStat column contains statuses for areas.

With SHOW(CI), the response would have columns for SDEP information. Sdep-T is the total number of CIs defined for sequential dependents. SDep-U is the number of these CIs which are unused. Dir-T is the total number of CIs defined for independent overflow. Dir-U is the number of these CIs which are unused.

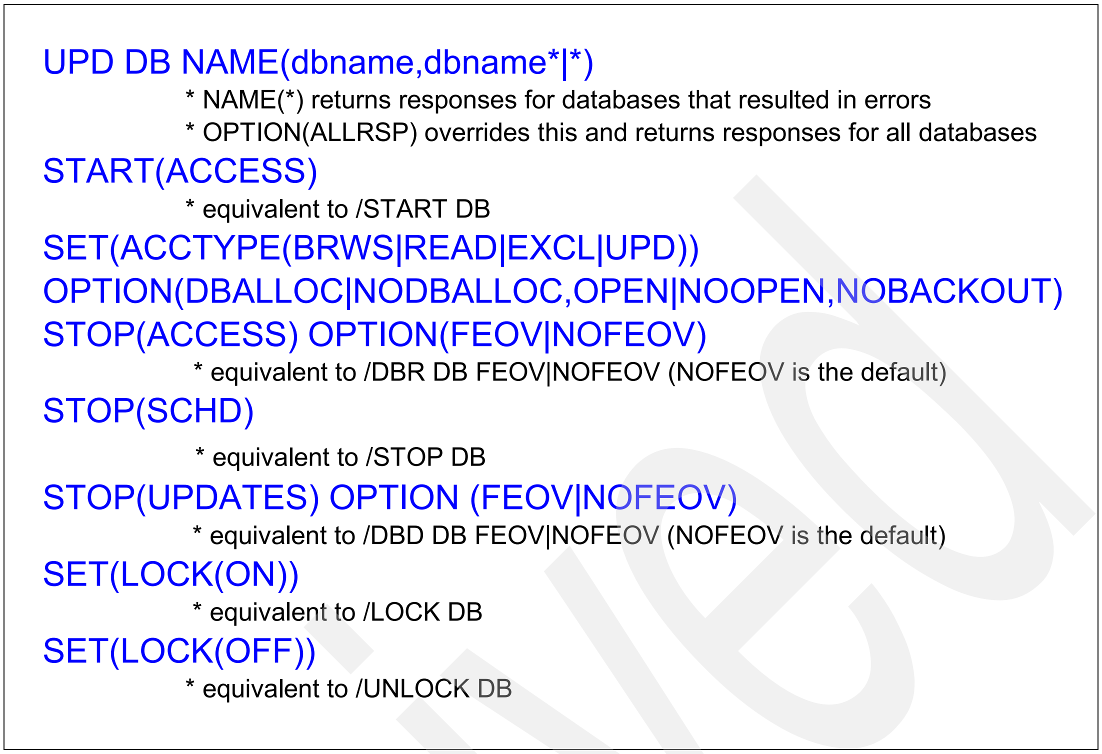


```
  File  Display  View  Options  Help
 ----------------------------------------------------------------------
 PLEX1                    IMS Single Point of Control
 Command ===> ________________________________________________________

                              Plex . PLEXB  Route . IM1B      Wait .
 Response for: QUERY AREA NAME(AREAD*,AREAI*,AREAW*) SHOW(STATUS)   More:    >
AreaName ADSName  MbrName  DBName     CC EQCnt LclStat>
AREADI01          IM1B     DISTDB      0       VSO,PREOPEN,PRELOAD,OPEN,SAS,SHAR
AREADI01 DISTDD01 IM1B                 0    10
AREAIT01          IM1B     ITEMDB      0       VSO,PREOPEN,PRELOAD,OPEN,SAS,SHAR
AREAIT01 ITEMDD01 IM1B                 0    10
AREAWH01          IM1B     WAREDB      0       VSO,PREOPEN,PRELOAD,OPEN,SAS,SHAR
AREAWH01 WAREDD01 IM1B                 0    10
```

Figure 7-8 QRY AREA output example

7.2.3 UPDATE DB command

Figure 7-9 on page 124 shows the format and keywords for UPDATE DB command.



```
UPD DB NAME(dbname,dbname*|*)
            * NAME(*) returns responses for databases that resulted in errors
            * OPTION(ALLRSP) overrides this and returns responses for all databases
START(ACCESS)
            * equivalent to /START DB
SET(ACCTYPE(BRWS|READ|EXCL|UPD))
OPTION(DBALLOC|NODBALLOC,OPEN|NOOPEN,NOBACKOUT)
STOP(ACCESS) OPTION(FEOV|NOFEOV)
             * equivalent to /DBR DB FEOV|NOFEOV (NOFEOV is the default)
STOP(SCHD)
             * equivalent to /STOP DB
STOP(UPDATES) OPTION (FEOV|NOFEOV)
             * equivalent to /DBD DB FEOV|NOFEOV (NOFEOV is the default)
SET(LOCK(ON))
             * equivalent to /LOCK DB
SET(LOCK(OFF))
            * equivalent to /UNLOCK DB
```

Figure 7-9 The format and keywords for UPDATE DB command

The use of the NAME keyword with the UPDATE DB or UPD DB command is similar to its use with other type-2 DB commands, but there is a difference in the responses when NAME(*) is specified. There is no response for databases which are updated successfully unless OPTION(ALLRSP) is specified.

The UPD DB command with START(ACCESS) is equivalent to a /START DB command. The SET(ACCTYPE(...)) specification is equivalent to the ACCESS= specification on the /START DB command. The OPTION(...) specifications are equivalent to the same keywords which are used with the /START DB command.

The UPD DB command with STOP(ACCESS) is equivalent to the /DBR DB NOFEOV command. It stops scheduling and deallocates the database(s). NOFEOV is the default for the OPTION keyword. If OPTION is not specified with STOP(ACCESS) the log will not be automatically switched to the next OLDS.

The UPD DB command with STOP(SCHD) is equivalent to the /STOP DB command. It stops scheduling for the database(s).

The UPD DB command with STOP(UPDATES) is equivalent to the /DBD DB NOFEOV command. It prevents updates to the database(s). NOFEOV is the default for the OPTION keyword. If OPTION is not specified with STOP(UPDATES) the log will not be automatically switched to the next OLDS.

NAME(*) cannot be specified with either SET(LOCK(ON)) or SET(LOCK(OFF)). Note that the UPD DB NAME(dbname) SET(LOCK) and UPD DB NAME(dbname) SET(UNLOCK) commands do not have SMU support for passwords. This differs from the /LOCK and /UNLOCK type-1 commands.

There is also a SCOPE keyword which may be used with UPD DB commands. Valid parameters with SCOPE are ACTIVE or ALL, however, they produce the same results in this release. They both apply changes to all active systems.

UPDATE DB and equivalent type-1 commands

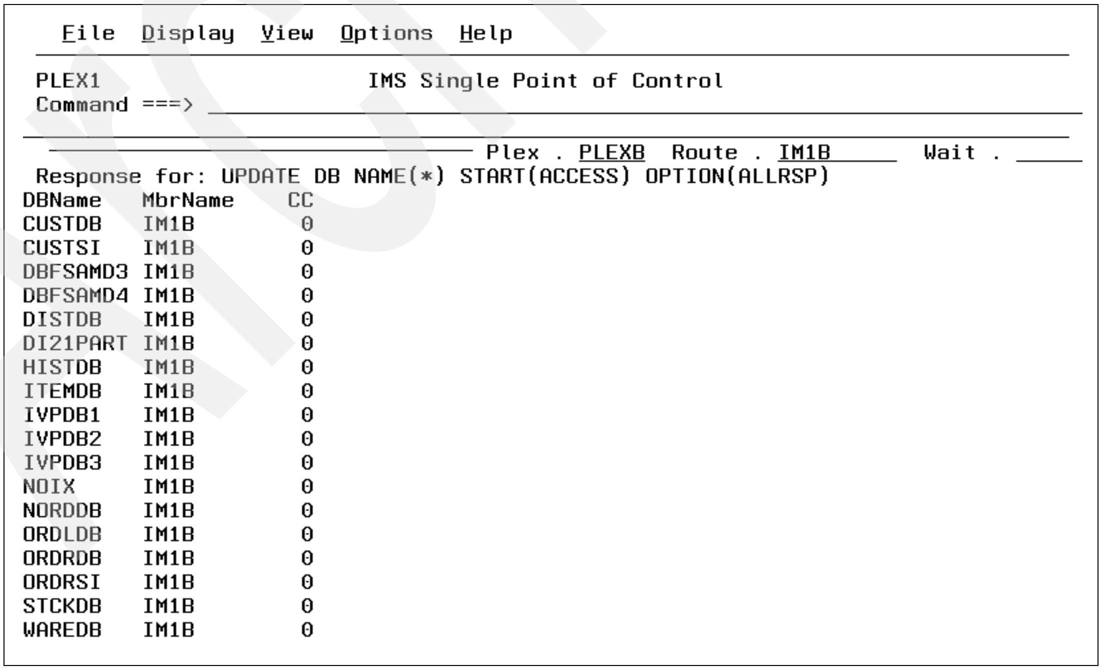
Table 7-3 shows some examples of UPD commands and their equivalent type-1 commands.

Table 7-3 UPD DB examples

UPD DB Command	Equivalent Type-1 Command
UPD DB NAME(AB34) START(ACCESS)	/START DB AB34
UPD DB NAME(AB34) STOP(SCHD)	/STOP DB AB34
UPD DB NAME(AB34) STOP(ACCESS)	/DBR DB AB34 NOFEOV
UPD DB NAME(AB34) STOP(UPDATES)	/DBD DB AB34 NOFEOV
UPD DB NAME(AB34) START(ACCESS) SET(ACCTYP(READ)) OPTION(OPEN)	/START DB AB34 ACCESS=RD OPEN
UPD DB NAME(AB*) START(ACCESS)	No equivalent; starts all databases beginning with AB

UPDATE DB output example

An example of the response is shown in Figure 7-10 where the TSO SPOC is used to issue an UPD DB NAME(*) START(ACCESS) command. OPTION(ALLRSP) is used here to produce response lines for each database started. Otherwise only the databases that resulted in errors would have been listed. The completion code of 10 indicates that the data base does not exists, resource not found. You can find the condition codes in the *IMS Version 9: Command Reference,* SC18-7814.

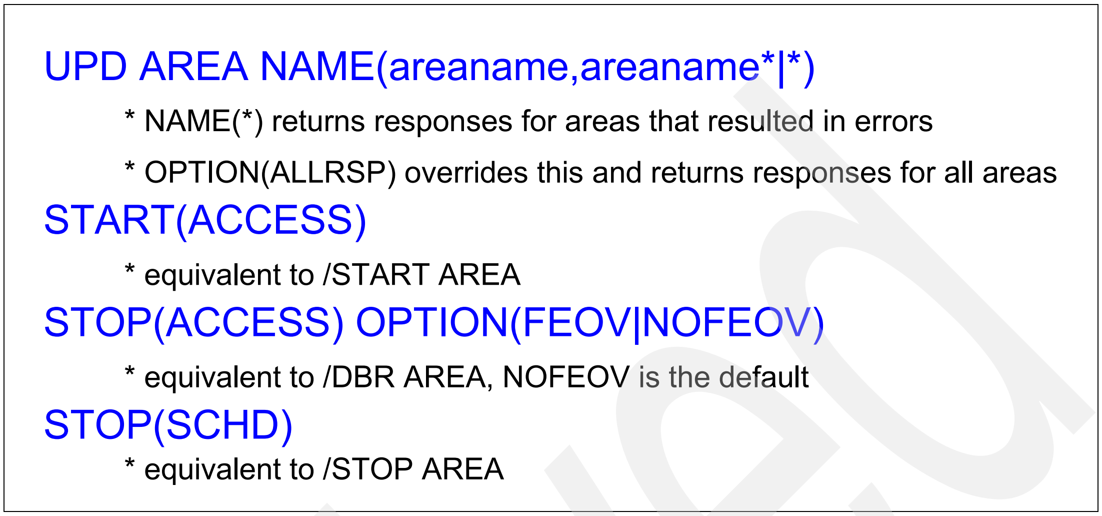


```
PLEX1                     IMS Single Point of Control
Command ===> 

                                  Plex . PLEXB  Route . IM1B      Wait . 
Response for: UPDATE DB NAME(*) START(ACCESS) OPTION(ALLRSP)
DBName   MbrName    CC
CUSTDB   IM1B        0
CUSTSI   IM1B        0
DBFSAMD3 IM1B        0
DBFSAMD4 IM1B        0
DISTDB   IM1B        0
DI21PART IM1B        0
HISTDB   IM1B        0
ITEMDB   IM1B        0
IVPDB1   IM1B        0
IVPDB2   IM1B        0
IVPDB3   IM1B        0
NOIX     IM1B        0
NORDDB   IM1B        0
ORDLDB   IM1B        0
ORDRDB   IM1B        0
ORDRSI   IM1B        0
STCKDB   IM1B        0
WAREDB   IM1B        0
```

Figure 7-10 UPD DB START(ACCESS) sample

7.2.4 UPDATE AREA command

Figure 7-11 shows the format of the UPDATE AREA command.



UPD AREA NAME(areaname,areaname*|*)

* NAME(*) returns responses for areas that resulted in errors

* OPTION(ALLRSP) overrides this and returns responses for all areas

START(ACCESS)

* equivalent to /START AREA

STOP(ACCESS) OPTION(FEOV|NOFEOV)

* equivalent to /DBR AREA, NOFEOV is the default

STOP(SCHD)

* equivalent to /STOP AREA

Figure 7-11 The format of UPDATE AREA command

The UPD AREA command is similar to the UPD DB command. It is used for DEDB areas, not databases. The keywords for the UPD AREA command have the same meaning as they have for UPD DB commands as seen in Table 7-4.

NOFEOV is the default for the OPTION keyword with STOP(ACCESS). If OPTION is not specified, the log will not be automatically switched to the next OLDS.

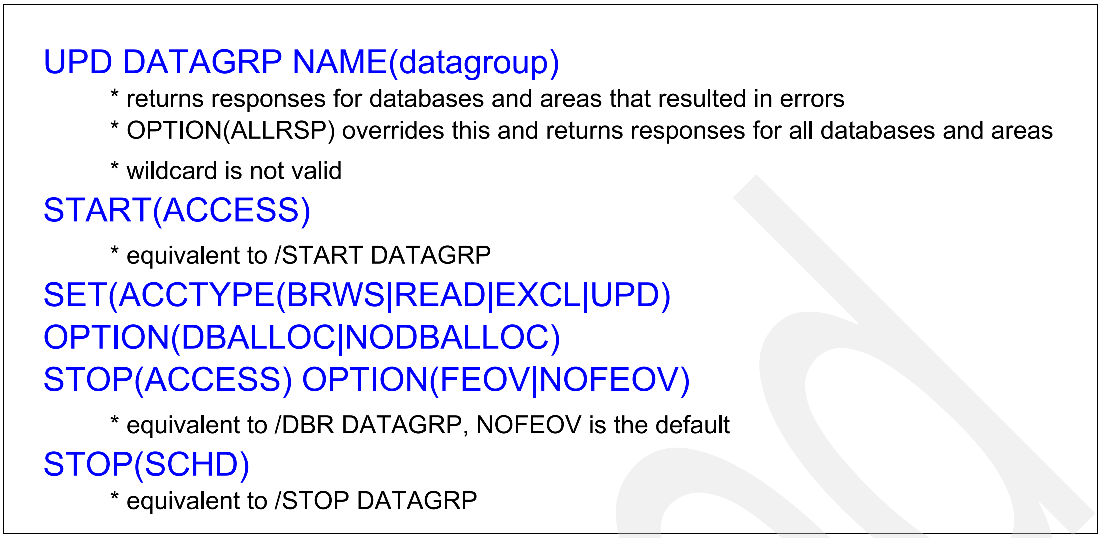
There is also a SCOPE keyword which may be used with UPD AREA commands. Valid parameters with SCOPE are ACTIVE or ALL, however, they produce the same results in this release. They both apply changes to all active systems.

Table 7-4 UPD AREA command examples

UPD AREA Command	Equivalent Type-1 Command
UPD AREA NAME(BXJ21) START(ACCESS)	/START AREA BXJ21
UPD AREA NAME(BXJ21) STOP(SCHD)	/STOP AREA BXJ21
UPD AREA NAME(BXJ21) STOP(ACCESS)	/DBR AREA BXJ21 NOFEOV
UPD AREA NAME(BX*) STOP(ACCESS)	No equivalent; Stops access for all areas beginning with 'BX'

7.2.5 UPDATE DATAGRP command

Figure 7-12 on page 127 shows the format and the keywords of the UPDATE DATAGRP command.



```
UPD DATAGRP NAME(datagroup)
      * returns responses for databases and areas that resulted in errors
      * OPTION(ALLRSP) overrides this and returns responses for all databases and areas
      * wildcard is not valid
START(ACCESS)
      * equivalent to /START DATAGRP
SET(ACCTYPE(BRWS|READ|EXCL|UPD)
OPTION(DBALLOC|NODBALLOC)
STOP(ACCESS) OPTION(FEOV|NOFEOV)
      * equivalent to /DBR DATAGRP, NOFEOV is the default
STOP(SCHD)
      * equivalent to /STOP DATAGRP
```

Figure 7-12 The format of UPDATE DATAGRP command

The UPD DATAGRP command is similar to the /START, /STOP, and /DBR commands with the DATAGRP keyword. There is no QRY DATAGRP command, just as there is no /DIS DATAGRP command.

Only one data group may be specified on a command. The asterisk cannot be used for the NAME parameter with the UPD DATAGRP command. Table 7-5 shows examples and their equivalent type-1 commands.

The NAME(datagroup) returns responses for databases and areas that result in errors. The OPTION(ALLRSP) overrides this and returns responses for all databases and areas.

For the START(ACCESS) there are the following additional keywords: SET(ACCTYPE(BRWS | READ | EXCL | UPD) OPTION(DBALLOC | NODBALLOC).

There is also a SCOPE keyword which may be used with UPD DATAGRP commands. Valid parameters with SCOPE are ACTIVE or ALL, however, they produce the same results. They both apply changes to all active systems.

Table 7-5 UPD DATAGRP command examples

UPD DATAGRP Command	Equivalent Type-1 Command
UPD DATAGRP NAME(GP35) START(ACCESS)	/START DATAGRP GP35
UPD DATAGRP NAME(GP35) STOP(SCHD)	/STOP DATAGRP GP35
UPD DATAGRP NAME(GP35) STOP(ACCESS)	/DBR DATAGRP GP35 NOFEOV

7.2.6 Command security

Security is available in OM for commands entered through it. There are two types of security available in OM, SAF (RACF) and the OM security exit. Either or both may be used. *IMS Version 9: Command Reference,* SC18-7814, lists resource names and RACF authority for IMS Commands. The “IMS Command, RACF Access Authorities and Resource Names Table” in the appendix of the command reference manual has been expanded to include the new DB commands. Table 7-6 on page 128 shows an example of an entry in that table.

Table 7-6 Resource names and RACF authority

Command	Keyword	Authority	Resource Name
UPD	DB	UPDATE	IMS.plxname.UPD.DB

The corresponding RACF definition for this could be as the one shown in Example 7-1.

Example 7-1 RACF definitions

```
REDEFINE OPERCMDS IMS.*.UPD.DB UACC(NONE)
PERMIT IMS.*.UPD.DB CLASS(OPERCMDS) ID(KNUT) ACCESS(UPDATE)
```

The OM security user exit routine is documented in *IMS Version 9: Common Service Layer Guide and Reference,* SC18-7816.

7.2.7 Log records

The type x'02' log record is written for type-1 (classic) IMS commands. Type-2 commands are logged in type x'22' instead of type x'02' log records. Type x'22' log records will be written when, for example, START(ACCESS) or STOP(SCHD) is specified on the following commands:

- UPD DB
- UPD AREA
- UPD DATAGRP

Log record x’22’ can be mapped with the DFSLOG22 macro.

7.2.8 Summary of the command enhancements

IMS Version 9 command enhancement extends type-2 commands by using the SPOC to query and update IMS database resource status. The benefit of this enhancement is the consistency with type-2 commands for other resources. Responses from multiple IMS systems are consolidated, and further enhanced to allow wildcards with NAME parameters. Responses are encapsulated in XML tags, so command clients may manipulate output for presentation as well as for automation. With the TSO SPOC and IMS Control Center support, output is easily sorted and manipulated by users.

7.3 Enhanced command environment

The enhanced command environment of IMS Version 9 allows installations to enter type-2 commands without including a Resource Manager (RM) address space in their environment. This simplifies the execution environment by eliminating these address spaces. In this environment users are able to automatically start the SCI and OM address spaces when the control region is started. This enhancement is ideal for DBCTL users and non-sysplex environments, which do not need the features of, for example, sysplex terminal management and global online change, which require Resource Manager.

7.3.1 IMS Version 8 Common Service Layer components

In IMS Version 8 the Common Service Layer (CSL) includes an SCI address space on each LPAR in the IMSplex. At least one OM address space is required in the IMSplex. Additional OM address spaces may be used to provide increased availability. At least one RM address space is required in the IMSplex. If a resource structure is included, multiple RM address spaces may be used to provide increased availability. If there is no resource structure, only one RM address space may be used. A Coupling Facility resource structure is required if the sysplex terminal management (STM) function is used. The resource structure is optional if RM is used only for global online change.

7.3.2 IMS Version 9 Common Service Layer components

IMS Version 9 removes the RM address space requirement for the Common Service Layer, unless the functionality of sysplex terminal management or global online change is required. Type-2 commands do not require an RM address space. They do require an SCI address space in each LPAR and at least one OM address space.

Online change using the INIT command does not require an RM address space, but the RM address space is recommended. Without the RM address space, the change is not “global”. It is made in only one IMS system.

Figure 7-13 is an illustration of an enhanced command environment CSL configuration. There is no RM address space or resource structure. Two of the LPARs have an OM address space. Only one is required, but the second provides increased availability.

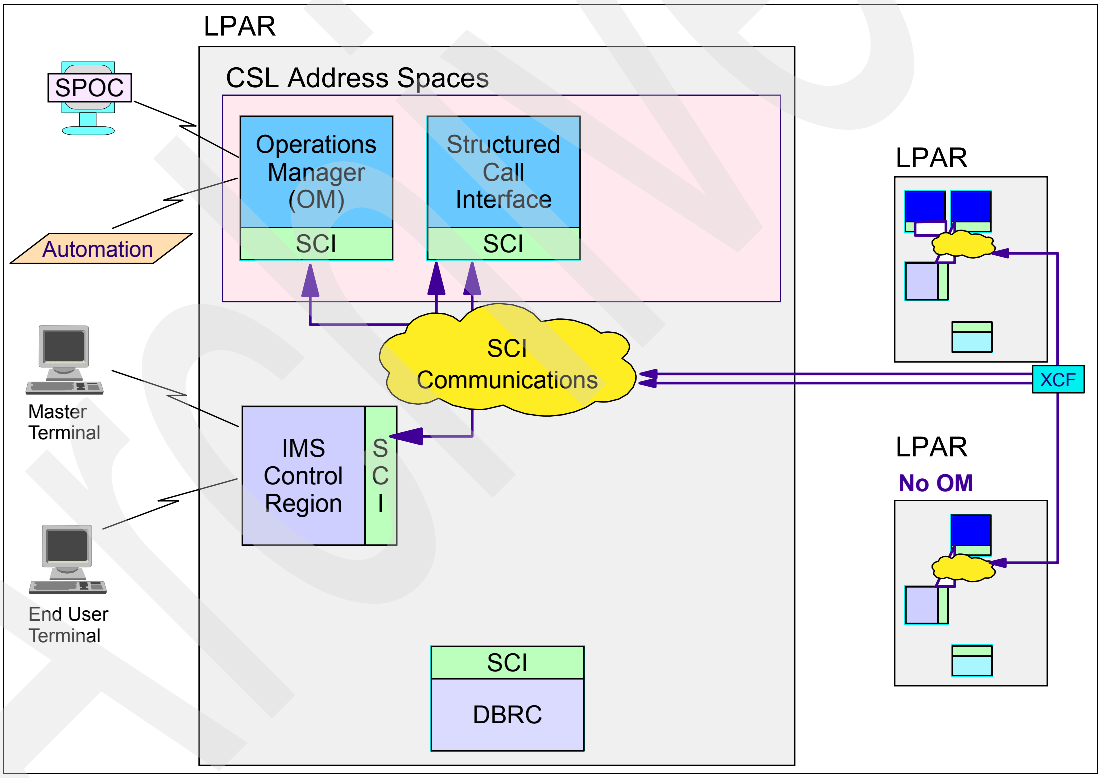


Figure 7-13 Enhanced command environment CSL configuration

Figure 7-14 on page 130 shows the minimum CSL configuration which may be used for entering type-2 commands. There is one LPAR with one OM address space and one SCI address space.

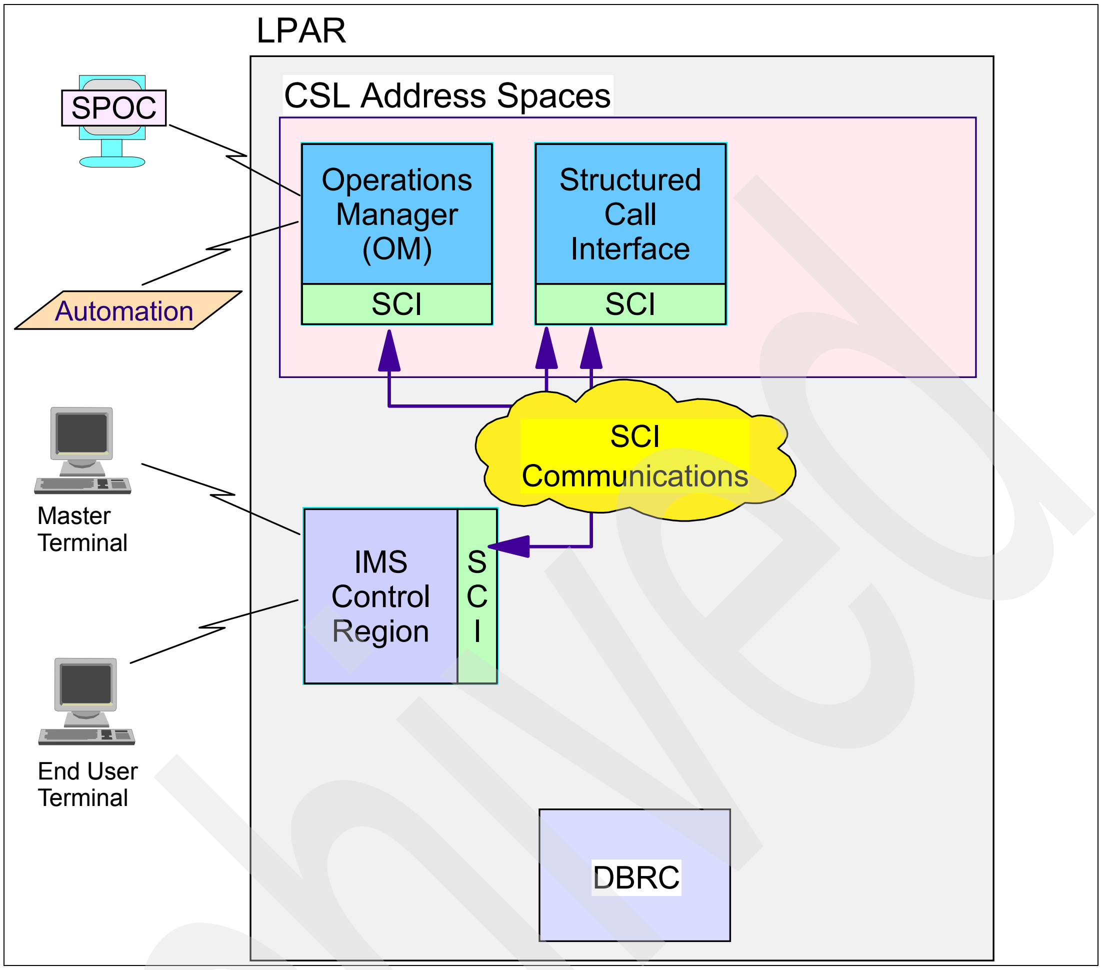


Figure 7-14 Minimum CSL configuration

7.3.3 CSL functions and requirements summary

This summarizes the minimum configurations required for the use of various functions provided by the Common Service Layer. Automatic RECON loss notification requires only SCI. An SCI address space is required on each LPAR in the IMSplex.

- SCI only required
  - Automatic RECON loss notification (ARLN)
- SCI + OM required
  - Type-2 commands
    - Online change using the INIT command
    - Dynamic LE runtime options
    - HALDB online reorganization commands
    - Other commands
- SCI + OM + RM without resource structure required
  - Global online change without resource consistency checking
- SCI + OM + RM with resource structure required
  - Sysplex terminal management
  - Global online change with resource consistency checking

7.3.4 CSL start up options and implementation

IMS Version 8 required that an installation start each CSL address space. Either an operator or automation had to start each address space. IMS Version 9 has a new option which causes the IMS control region to start the OM and SCI address spaces. This is intended for installations that are using the CSL only for type-2 commands in an enhanced command environment and, possibly, automatic RECON loss notification.

The automated start of OM and SCI address spaces is controlled by the parameters in DFSCGxxx member in IMS.PROCLIB. The parameters are the following:

- RMENV=

  RMENV=Y is the default and indicates that an RM address space will be used. It is not automatically started by the IMS control region. RMENV=Y does not allow the control region to automatically start the SCI or OM address spaces.

- SCIPROC=

  This parameter is used to specify the procedure name for the SCI address space. When it is specified, the control region will start the SCI address space if one is not already available on the LPAR, and RMENV=N is also specified.

- OMPROC=

  This parameter is used to specify the procedure name for the OM address space. When it is specified, the control region will start the OM address space if one is not already available in the IMSplex, and RMENV=N is also specified

Example 7-2 shows a sample DFSCGxxx PROCLIB member that makes it possible to use automatic start of OM and SCI address spaces when IMS control region is started.

Example 7-2 A sample contents of DFSCGxxx member

```
********************************* Top of Data **********************************
*--------------------------------------------------------------------*
* IMS COMMON SERVICE LAYER PROCLIB MEMBER                            *
*--------------------------------------------------------------------*
CMDSEC=N,                    /* NO CMD AUTHORIZATION CHECKING     */
IMSPLEX=PLEXG,               /* IMSPLEX NAME                      */
RMENV=N
OMPROC=IVP91OM,
SCIPROC=IVP91SCI
*--------------------------------------------------------------------*
*  NORSCCC=(ACBLIB,FORMAT,MODBLKS)
*  OLCSTAT=IMSTESTG.GLOBLOLC.STAT
*--------------------------------------------------------------------*
* END OF MEMBER DFSCGECE                                             *
*--------------------------------------------------------------------*
******************************** Bottom of Data ********************************
```

7.3.5 Online change alternatives

There are three ways of implementing online change.

- OLC=LOCAL specification causes IMS to use the online change functions available in IMS releases before IMS Version 8.

  Online change is invoked with /MODIFY commands and keeps statuses in the MODSTAT data set. An online change affects only one IMS. Each IMS system must have its own MODSTAT data set.

- OLC=GLOBAL specification with RMENV=N causes IMS to use the online change functions introduced in IMS Version 8.

  Online change is invoked with INIT OLC commands using OM. Since there is no RM, the online change affects only one IMS system. Each IMS system must have its own OLCSTAT data set.

- OLC=GLOBAL specification with RMENV=Y causes IMS to use the online change functions introduced in IMS Version 8.

  Online change is invoked with INIT OLC commands using OM. Since there is an RM, the online change affects all of the IMS systems in the IMSplex. The IMS systems share an OLCSTAT data set.

7.3.6 Global online change

The use of global online change is optional. This is true for both IMS Version 8 and Version 9. IMS Version 9 adds the capability to run an IMSplex without an RM address space. This may affect the capabilities of global online change.

The use of global online change requires that OLC=GLOBAL be specified in the DFSCGxxx PROCLIB member. If there is no RM address space, OLC=GLOBAL does not invoke a global online change. Instead, the OLC commands must be issued separately for each IMS system. Since online changes for different IMS systems are processed separately, each IMS system must have its own OLCSTAT data set.

The IMS system using an OLCSTAT data set, writes its name in the data set. If a restart of IMS reads the OLCSTAT data set and finds another IMS name in the data set, restart fails. The name in the data set may be changed by using the OLC utility (DFSUOLC0).

Resource consistency checking cannot be done without RM. Resource consistency checking is the function which checks to ensure that each IMS system is using the same ACBLIB, MODBLKS, MATRIX, and FORMAT data sets.

Note: Without an RM, there is no check for the use of the same data sets ACBLIB, MODBLKS, MATRIX, and FORMAT.

7.3.7 OM security user exit

Users may implement an OM security user exit. It is used to perform security checking during OM command processing. It is optional.

The exit is invoked when the CMDSEC= parameter on the OM procedure is specified as A or E. A indicates that both the OM security user exit and RACF are used for OM command security. E indicates that only the OM security user exit (and not RACF) is called for OM command security. The exit is defined as TYPE=SECURITY in the EXITDEF statement in the BPE user exit list PROCLIB member.

The input parameter list for the OM security exit has two new parameters:

- OSCX_ROUTLEN is the length of the route list. This is the list of IMS systems to which the command is routed. This list may be used in security decision.
- OSCX_ROUTPTR is the address of a copy of the route list.

The new parameters are included in the CSLOSCX mapping macro. This change applies to IMS Version 9 whether or not the enhanced command environment is used.

QUERY commands with RMENV=N

The responses to QUERY calls are affected by the lack of an RM. These effects are explained here:

- QUERY MEMBER response

  RMENVNO attribute returned when RMENV=N

- QUERY OLC response

  Information about other IMS systems is not available

  – Command is only processed by the command master
  – OLCSTAT is not shared when RMENV=N

  May need to route to the appropriate IMS

- QUERY TRAN response

  Only information local to the IMS processing the query is returned

7.3.8 Migration to OM

Instructions for implementing OM are included in the Operations Manager chapter of the *IMS Version 9: Common Service Layer Guide and Reference,* SC18-7816. The DFSCGxxx member to be used for the IMSplex is specified in the CSLG= in DFSPBxxx or as an execution parameter for the IMS control region. A warm start is sufficient, when the CSLG= parameter value is changed.

The IMSplex parameters in the DFSCGxxx member include RMENV=, CMDSEC=, IMSPLEX=, NORSCCC=, OLC=, OLCSTAT=, and LEOPT=. NORSCCC= is not applicable when RMENV=N is specified. The IMSPLEX specification must match the IMSPLEX specification for the SCI and OM address spaces.

Procedures for the OM and SCI address spaces must be created. Their initialization parameters are specified in the CSLSIxxx and CSLOIxxx PROCLIB members. As mentioned before, if SCIPROC and OMPROC are specified in the DFSCGxxx member used for the control region and the SCI and OM address spaces are not already active, the control region will start these address spaces. If the SCIPROC and OMPROC parameters are not specified, the address spaces must be started by the operator.

7.3.9 Migrating from non-IMSplex to IMSplex with RMENV=N

Follow these steps when migrating from a non-IMSplex environment to an enhanced command environment, in other words, an IMSplex environment with RMENV=N.

1. Add CSLG=xxx to IMS control region parameters

   – Specifies DFSCGxxx PROCLIB member

2. Create DFSCGxxx member with RMENV=N

   – Specify the same IMSplex name as used by SCI and OM

3. Create SCI and OM startup procedures
4. Create initialization parameter PROCLIB members for SCI and OM

   – CSLSIxxx
   – CSLOIxxx

5. Start SCI and OM or specify procedure names in DFSCGxxx
6. Shut down IMS

7. Warm start IMS

7.3.10 Migrating from RMENV=N to RMENV=Y

To migrate from the enhanced command environment (no RM address space) to a Common Service Layer including an RM address space, the RMENV parameter for the control region must be changed so that RMENV=N is not specified. Either RMENV=Y may be specified or the RMENV= parameter may be removed.

The IMS control region will not start the SCI and OM address spaces when RMENV=Y is specified. If this was being done with RMENV=N, operations procedures must be modified to start these address spaces. The RM address space is started and then IMS is restarted with a warm start. A cold start of IMS is not required.

Follow these steps for migrating from enhanced command environment to CSL with Resource Manager:

1. Update DFSCGxxx member to specify RMENV=Y, or take the default by removing RMENV=N
2. Create RM startup procedure
3. Create initialization parameter PROCLIB member CSLRIxxx for RM
4. If previously using IMS to start SCI and OM, update operations procedures to start SCI and OM
5. Start RM
6. Shut down IMS
7. Warm start IMS

7.3.11 Enhanced command environment summary

To summarize the previous pages, the enhanced command environment is an IMSplex with SCI and OM address spaces but without Resource Manager (RM) address spaces. This environment may be used to enter commands through the OM interface. These commands may be type-1 IMS commands or new type-2 commands including commands to initiate HALDB online reorganization, commands to dynamically modify LE runtime options, and commands to invoke the online change process introduced in IMS Version 8.

The enhanced command environment is simpler than the full IMSplex environment. Since the IMS control region can automatically start the OM and SCI address spaces, the enhanced command environment is simpler to operate.

8

Integrated IMS Connect support

The functionality of IMS Connect Version 2 Release 2 (program number 5655-K52) — one of IBM DB2 and IMS Tools products — is included as part of IMS Version 9. This functionality is included in the IMS systems services Function Modification Identifier (FMID) HMK9900.

You can use IMS Connect 2.2 to provide communications between one or more TCP/IP clients and one or more IMS Version 7 or IMS Version 8 clients. For communications with an IMS Version 9 client, you can use the integrated IMS Connect functionality with IMS Version 9. IMS Connect 2.2 will continue to be supported for IMS Version 7 and IMS Version 8 clients, but future enhancements to the IMS Connect functionality will be made available only with IMS Version 9 or later.

8.1 IMS Connect overview

Integrated IMS Connect function provides advanced security and transactional integrity in TCP/IP and local/390 access to IMS and the Internet, as IMS Connect itself has been providing. It supports high performance communications between one or more TCP/IP or local/390 clients, and one or more IMS systems. It provides commands to manage that environment and assist with workload balancing. It reduces the design/coding effort for client applications, and it provides easier access to IMS applications and operations.

Integrated IMS Connect function used with the WebSphere Development Tooling and the IMS Connector for Java, can significantly ease the development of on demand business solutions that access IMS transactions. These solutions can be deployed in IBM WebSphere Application Servers, allowing you to use Web applications, J2EE applications, or Web Services to quickly transform static Web sites into sources of dynamic Web content.

Integrated IMS Connect function, used with IBM DB2 Version 8 Universal Database Control Center, allows a single, graphical user interface to control both IMS and DB2, easing IMS operations. For more information about setting up the DB2 and IMS Control Center, refer to Appendix D, “IMS Control Center” on page 299.

IMS Connect runs in an address space on z/OS system, and performs router functions between TCP/IP clients and local option clients with IMS. Request messages received from TCP/IP clients, using TCP/IP connections, or local option clients, using the z/OS Program Call (PC), are passed to IMS through cross-system coupling facility (XCF) sessions. Then IMS Connect receives response messages from IMS and passes them back to the originating TCP/IP or local option clients. Figure 8-1 shows IMS Connect functions for TCP/IP clients.

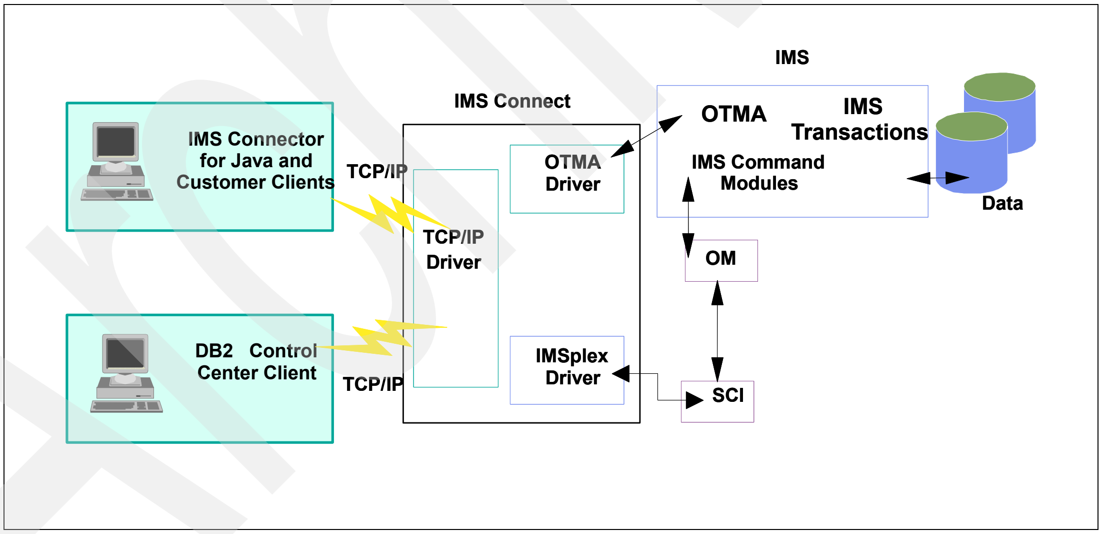


Figure 8-1 IMS Connect overview

8.1.1 IMS Connect enhancements in IMS Version 9

For IMS Version 9, IMS Connect is an integrated function, which offers a functional replacement for the IMS Connect tool (program number 5655-K52). Basically, the integrated IMS Connect function in IMS Version 9 provides the same functionality and enhancements to IMS as provided by the stand-alone tool, IMS Connect Version 2.2.

For IMS Connect Version 2.2 and IMS Version 9 integrated IMS Connect function, the improvements and enhancements include:

- Commit mode 0 persistent socket support
- Purge not deliverable support
- New and enhanced IMS Connect commands
  - IMS Connect commands can now be issued through the z/OS modify interface. This is a z/OS command. The response to it is presented by z/OS to the issuer of the command, which enhances consistency and ease of use
  - Query member command enhancements add IMS Connect IP address and version
- Cancel timer support
- New RESUME_TPIPE single with wait option
- The MAXSOC= parameter has changed and requires APAR PQ90051
- IMS Connect event recording support
- Improved performance and availability reporting
- IMS Connector for Java adds:
  - Commit Mode 0 persistent socket for improved performance
  - Socket timeout for enhanced usability
  - Retry for improved availability

IMS Connect 2.2 and the integrated IMS Connect function in IMS Version 9 are the last releases which support user message exits, HWSIMSO0 and HWSIMSO1. These two user message exits will not be available in future.

8.2 IMS Connect clients

Using IMS Connect, you can develop applications as a:

- Roll Your Own (RYO) client
- IMS Connector for Java client

A RYO client can be written in any language which supports the TCP/IP interface. You need to know the protocol provided by IMS Connect and OTMA to develop this TCP/IP socket application. The advantage of a RYO client compared to an IMS Connector for Java client is that the RYO client has similar functionality to and flexibility of MQ-IMS Bridge client applications, which support the OTMA protocol and the message routing.

IMS Connector for Java development environment is included in WebSphere Studio Application Developer Integration Edition and the runtime component is provided with IMS. The IMS Connector for Java client may have a little less flexibility than a RYO client, depending on your requirements. But you can build an IMS Connector for Java client without in-depth knowledge of the protocol of IMS Connect and OTMA, and you can use utilities which aid programmer productivity.

Functions regarding the connectivity using IMS Connect introduced here are not a new function of IMS Version 9, but of IMS Connector for Java Version 2.2.2 for WebSphere Studio Application Developer Integration Edition (WSADIE) Version 5.1.1.

For additional information about OTMA and IMS Connect, see the following IBM Redbook*:* *IMS e-business Connectors: A Guide to IMS Connectivity*, SG24-6514.

8.3 IMS Connector for Java client overview

IMS Connector for Java is based on the J2EE Connector Architecture (J2C) to connect to an IMS system, which is the standard architecture of J2EE. In this architecture, IMS is a so called Enterprise Information System (EIS) as is CICS, thus the IMS resource adapter is provided based on J2C architecture. However, from the IMS point of view, there is no difference between RYO clients and IMS Connector for Java clients. They both issue the transaction code through IMS Connect and OTMA.

IMS Connector for Java provides its own functions which RYO clients do not have. For example, you can use the local option if your J2EE application component is running on WebSphere Application Server for z/OS. With this option, you can submit IMS transaction messages that participate in global transaction processing. This transaction processing is coordinated by Resource Recovery Services (RRS) on z/OS and WebSphere Application Server for z/OS. IMS Connector for Java is RRS-compliant and is designed specifically to work with RRS so that the Java transaction manager in WebSphere Application Server and IMS, as the resource manager, can work together to make consistent changes to multiple protected resources. The XA protocol is not used by IMS Connector for Java when running global transactions with the local option.

Note that different resource adapters are provided depending on platforms, and that some functions of OTMA messaging protocol or flow are not supported depending on IMS version, IMS Connect version and IMS Connector for Java version. It is necessary to check your components and their versions to know which function is available at that time.

With IMS Version 9 and IMS Connector for Java Version 2.2.2, the send only message flow is supported as a new function. See Figure 8-2.

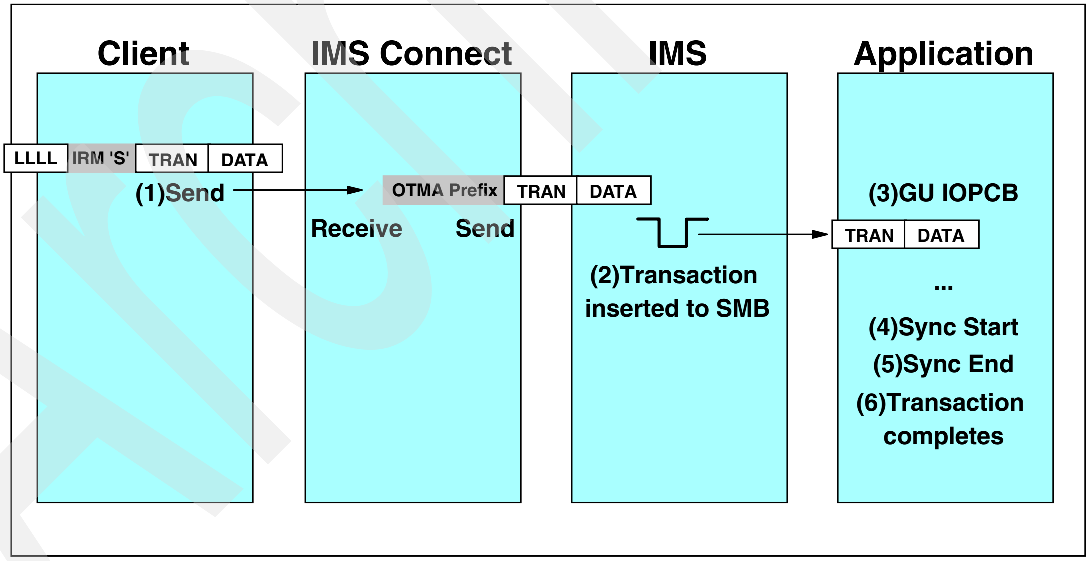


Figure 8-2 Commit-then-send and send-only transaction flow

The message type is specified as a send-only transaction in the IRM header. With IMS Connector for Java, you can specify a send-only flag as an interactionVerb property based on J2C architecture. So you do not need to be aware of which field in the IRM header is used to specify it.

8.3.1 IMS Connector for Java functions

As IMS Connector for Java is based on J2C, its functions are J2C compliant. You specify the information regarding connection, for example IMS Connect connected to or OTMA commit mode, on ConnectionSpec and InteractionSpec which are provided by J2C as standard. These Specs wrap the detail for the interaction protocol between OTMA, IMS Connect and a client application. Thus you do not need to study which fields of the message have what kind of information, thus making IMS Connector for Java easier compared to developing a RYO client.

ConnectionSpec and InteractionSpec contain the following information:

- Connection information to IMS Connect, IMS security information
- OTMA commit mode, message flow information
- IMS internal timeout value
- Information of conversational transaction

IMS Connector for Java provides the following functions as J2C System Contract, which are available only for IMS Connector for Java clients:

- Socket connection management (for example, connection pooling, timeout)
- Component-managed, Container- managed sign on security, or SSL as security
- Global transaction
- Trace function

For more detail of these functions, refer to manual *IMS Connector for Java 2.2.2 and 9.1.0.1 Online Documentation for WebSphere Studio Application Developer Integration Edition 5.1.1,* SC09-7869, which is available at the following Web site:

http://www.ibm.com/software/data/db2imstools/imstools/imsjavcon.html

8.3.2 IMS Connector for Java clients

Regarding IMS Connector for Java original functions, WebSphere tooling today enables IMS transactions using COBOL, C, and MFS-based applications as Web Services. Using IMS transactions as Web Services leverages customers' past investment in applications development. It can also eliminate or greatly reduce new programming effort, reduce end-to-end business process transformation, and facilitate application integration with partners, suppliers, and customers.

Using IMS Connect and IMS Connector for Java of WSADIE, you can develop the following type of applications with distributed Web access across platforms:

- J2EE Connector Architecture (J2C) Common Client Interface (CCI) applications
- Enterprise Service applications

IMS Connector for Java CCI applications have the advantage that you can utilize the CCI API which covers up the access logic detail required to connect to IMS, and you can develop applications in a simple MVC model. However, you may need to code applications manually including logic to build input and output messages and to convert character sets.

If you choose an Enterprise Service application, GUI-based application development functions provided by WSAD IE are available. You may just follow the steps provided by the GUI tools, in which case Enterprise Service applications corresponding to Web Services are generated automatically. Note the simple MVC-model-based applications are not generated using these GUI tools, and this method requires the skills regarding servlet, JSP and Web Service components like XML, SOAP, WSDL, WSIF and EJB.

8.4 IMS transactions as Enterprise Service applications

Here we introduce the way to see IMS transactions as Enterprise Service applications. From the client application's point of view, IMS transactions appear as an exchange of message which have their own layout as required by IMS or an IMS online application. So client applications are not necessarily aware of the process inside of IMS online applications. They recognize IMS transactions simply as an interaction of messages. The format of messages can be given by:

- I/O area of COBOL or C applications' source
- MFS source

This exchange of messages is known as a service. Once you define an IMS transaction as a service, access to an IMS system from any platform or protocols is allowed with mechanisms on z/OS or OS/390®.

With Enterprise Service, all of the information regarding IMS transaction message exchange, that is, IMS service is defined using Web Service Description Language (WSDL). WSDL is the open standard format to describe a Web Service. IMS Enterprise Service provides the way to invoke this Web Service, with the Web Service Invocation Framework (WSIF). WSIF is the API is used to invoke a Web Service and enables Web Service to be accessed from Java, RMI/IIOP, and SOAP. IMS Enterprise Service uses IMS Connector for Java and IMS Connect, and can therefore utilize the functions provided by J2C.

8.5 IMS Enterprise Service component

You describe all of the information about sending and receiving IMS transactions in WSDL. See Figure 8-3 on page 141.

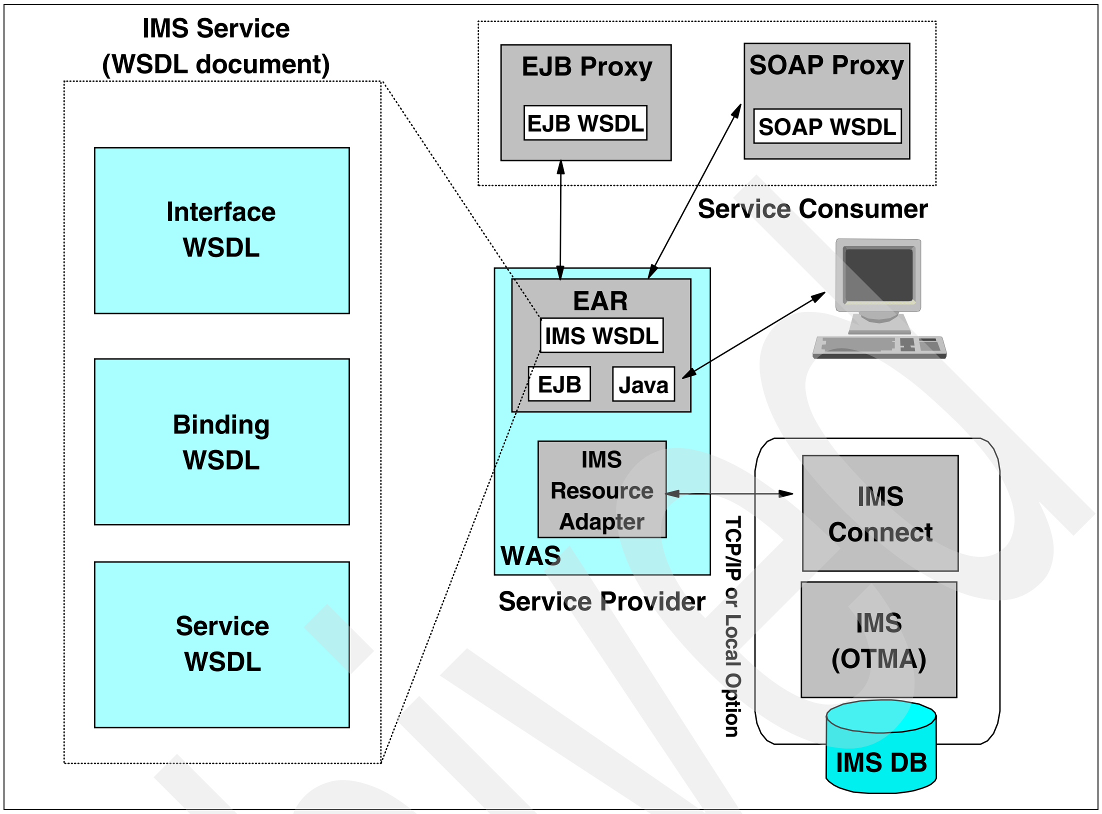


Figure 8-3 IMS Enterprise Service components

WSDL is the origin standard used to describe Web Service information, but with an Enterprise Service, you can describe information about not only Web Services but also various protocols like J2C, JMS, SOAP, and EJB and so on. When WSDL describes an IMS Service it has input and output message format information, an encoding specification, and J2C binding information corresponding to InteractionSpec and ConnectionSpec.

IMS Enterprise Service uses this information described in WSDL, and performs input and output processing of IMS transactions. WSIF is a Java API package which supports various protocols and transports such as SOAP, original Java, EJB, or COM objects, and executes services based on WSDL. WSIF is enhanced to correspond to J2C so that you can modify InteractionSpec or ConnectionSpec property described in WSDL. Compared to Web Services based on SOAP, IMS Enterprise Service can deal with more protocols because it has the benefits of WSIF.

IMS Enterprise Service is used by classes generated automatically from WSDLs. These classes have WSIF code to invoke Service whose detailed processing is covered up, so that clients can transparently invoke Service simply by passing arguments to the methods in Proxy.

You can deploy IMS Enterprise Service as an EJB, then RMI/IIOP or SOAP can access this Service. In this case, those protocols use Binding WSDL and Proxy by WSIF.

8.5.1 Interface WSDL

In the Interface WSDL, as message elements and type elements, message formats of IMS transactions are expressed by XML schema. This description can automatically be generated from COBOL and C application source or MFS source. Only fundamental information is described here, so it is common regardless of how it is deployed.

8.5.2 Binding WSDL

In the Binding WSDL, J2C binding information is described to use the IMS resource adapter. It has physical format elements and format elements which describe encoding information for IMS transaction messages. This information is used to convert EBCDIC-Unicode, endian, or used for byte operation. Physical format element and formant elements can automatically be generated from COBOL and C application source or MFS source.

8.5.3 Service WSDL

In the Service WSDL, the location of Service is described. In the case of J2C binding, Service location information corresponds to the J2C ConnectionSpec. You can utilize J2C binding in a managed environment without EJB deployment because the connection factory resource name can be written here directly.

8.6 MFS-based transaction as Enterprise Service application

With valid MFS source, MFS-based IMS enterprise service can be developed using MFS Web Services Importer inside WebSphere Studio Application Developer Integration Edition. The service runs on WebSphere Application Server and uses MFS Web Services Runtime, IMS Connector for Java, and IMS Connect.

8.6.1 MFS Web Services Importer

Available in WebSphere Studio Application Developer Integration Edition, the importer wizard takes users through parsing the MFS source file, selecting the device type, device feature, input MID, input logical page number, and output MODs, and populating the transaction information into the WSDL files. Infopops provide explanations for each control on the wizard pages. Multiple MODs can be chosen for multiple possible output types. System default MODs are provided in the selection by default. After running through the wizard, XML schema representation of the chosen MID and MODs is stored in the Interface WSDL. The information described in the WSDL files is used to generate Java beans and perform input and output processing. A Java bean represents each MID or MOD.

8.6.2 MFS Web Services Adapter

The MFS Web Services Adapter is packaged with IMS Connector for Java. Invoked by WSIF on input, the adapter transforms input bean into host input data stream and returns populated output bean from host output data stream. On output, the adapter transforms host output data stream to populate the output bean, as specified by the mapname in the OTMA header which is the same as the MOD name for the IMS application. See Figure 8-4 on page 143

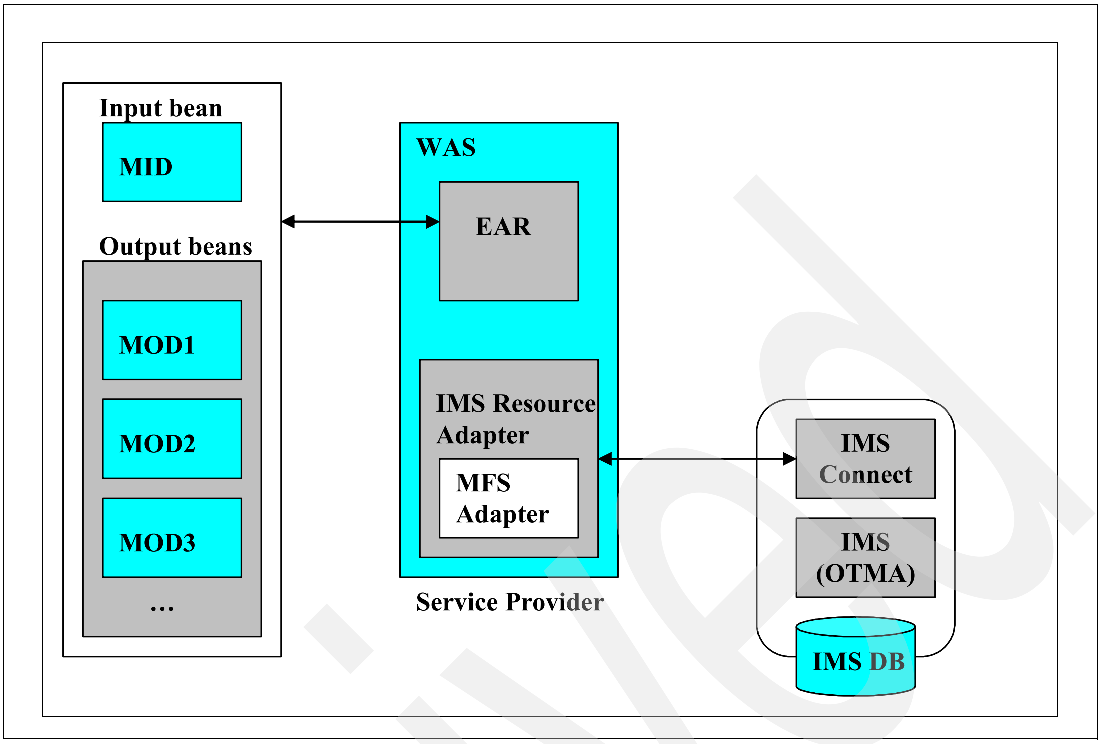


Figure 8-4 MFS Web Services adapter

For more information, refer to *IMS Connector for Java 2.2.2 and 9.1.0.1 Online Documentation for WebSphere Studio Application Developer Integration Edition 5.1.1,* SC09-7869, which is available at the following Web site:

http://www.ibm.com/software/data/db2imstools/imstools/imsjavcon.html

9

XML storage in IMS databases

In this chapter we introduce the enhanced ability of IMS to provide a storage facility for XML documents. XML stands for Extensible Markup Language, and throughout this chapter we will use XML only. This enhancement is provided as part of the IMS Java function. See Chapter 10, “IMS Java enhancements” on page 159 for other functionality regarding IMS Java.

This chapter includes:

- XML database overview
- Storing XML information in IMS databases
- XML schema definition language
- Decomposed storage
- Intact storage
- DLIModel Utility
- JDBC interface

9.1 IMS Version 9 XML database

Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML plays an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. In IMS we use XML in similar way to how we use the DBD. It describes the fields and segments of i.e. a WEB format versus a IMS Database.

An increasing amount of information is being transferred inside and between organizations in the form of XML documents. IMS databases are ideally suited to the storage and retrieval of information in XML documents, given that both are structured hierarchically. IMS Version 9 provides support for:

- Storage and retrieval of XML documents in IMS databases
- Composition of XML documents from existing IMS databases
- Creation of IMS segments from XML documents (decomposition)
- Intact storage of XML documents into IMS databases (without decomposition)

IMS also provides tools to assist you to define the metadata for mappings between your IMS database definitions and those required for XML. This is provided through an extension of the DLIModel Utility.

It is now a simple process to exchange data between your existing IMS databases and XML documents. It is straightforward to use existing IMS databases to create XML documents, which can be done without the need to alter any of your existing IMS applications.

IMS Java application programming support has also been extended to the following environments:

- IMS TM
- CICS
- DB2
- WebSphere Application Server

All DEDBs and full function databases, including HALDBs support these XML enhancements. Secondary indexes and logical relationships can be used for retrieval of information from IMS databases, although there are some restrictions on the insertion of segments through these facilities.

9.1.1 Storing XML information in IMS databases

There are two ways of storing XML documents in IMS databases:

- Decomposed storage

The use of *decomposed storage* is a data-centric approach to storing the information in IMS database. In this case, the XML data is converted to traditional IMS data types, when stored into the database. Since the data is not stored as XML, the information in these IMS databases may be read and updated by existing IMS applications.

- Intact storage

The use of *intact storage* is a document-centric approach to storing the information into your IMS database. In this case, the XML data is not converted when stored in an IMS database. The data is stored as XML, and includes all the XML tags as well as the information. This data cannot be manipulated by non-XML applications.

9.2 XML database overview

An XML schema can be created by the IMS DLIModel Utility. This utility reads PSB and DBD source, (XML Metadata Interchange Format) XMI files, and control statements. WebSphere Studio Application Developer Integration Edition (WSADIE) may be used to create these XMI files from COBOL copybooks. The XML schema is used by an IMS Java application to compose an XML document from an IMS database record, or to compose IMS database segments from an XML document.

The XML schema is also used by IMS Java applications to compose and decompose XML documents, as shown in Figure 9-1.

The XML Metadata Interchange Format (XMI) specifies an open, information interchange model that gives developers working with object technology the ability to exchange programming data over the Internet in a standardized way. The XMI standard will allow developers to leverage the Web to exchange object-oriented data among tools, applications, and repositories. The standard will also allow developers to build secure, distributed applications in a team development environment.

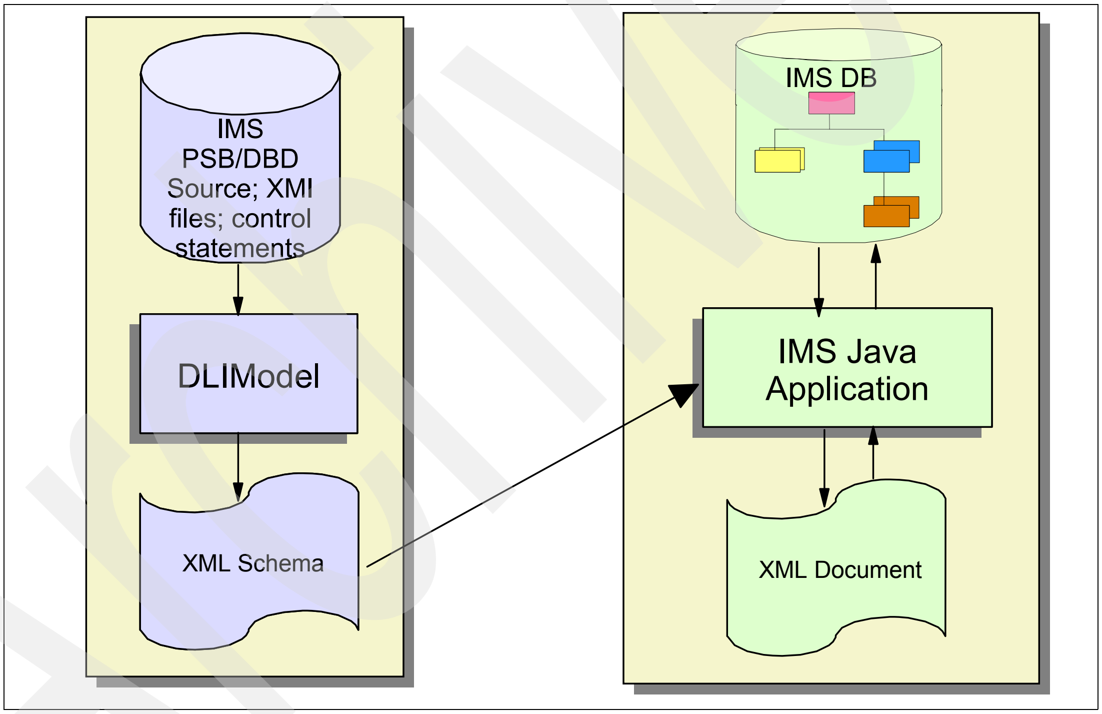


Figure 9-1 DLIModel Utility

IMS XML DB uses XML schema

An XML schema defines what data may be present in an XML document. The schema can be used to create an XML document from IMS database data, by describing how to copy data from the segments into an XML document.

The schema may also be used to store a document in a database. The XML data can either be stored in a decomposed or an intact format in the database.

9.2.1 XML schema definition language

An XML schema definition language is a language for defining the legal building blocks of a valid XML document. An XML schema defines the following:

- The elements and attributes that can appear in a document
- Which elements are child elements
- The order and number of child elements
- Whether an element is empty or can include text
- Data types for elements and attributes
- Default and fixed values for elements and attributes

The schema also defines an agreed-upon communication contract for exchanging XML documents. The generated XML schema must be made available at runtime in order to provide the XML structure of the data retrieved from the database or conversely of an incoming XML document being stored into IMS.

9.2.2 Decomposed storage

An XML document must be parsed and validated in order to extract the information from the document. The data must then be converted to traditional IMS types:

- EBCDIC characters (picture strings)
- Numeric types (COMP-1, COMP-2, and so on.)

The stored data is immediately usable by traditional IMS applications, without any further transformation. These traditional applications need have no knowledge of XML, since the data is stored in decomposed format in the database. Because the XML document does not have to be regenerated when the data is retrieved from the database, the retrieval of the XML data is typically faster than when it is stored without its XML tagging.

The definition of the mappings of data in an XML document to segments and fields in an IMS database is performed by the XML schema, as shown in Figure 9-2 on page 149.

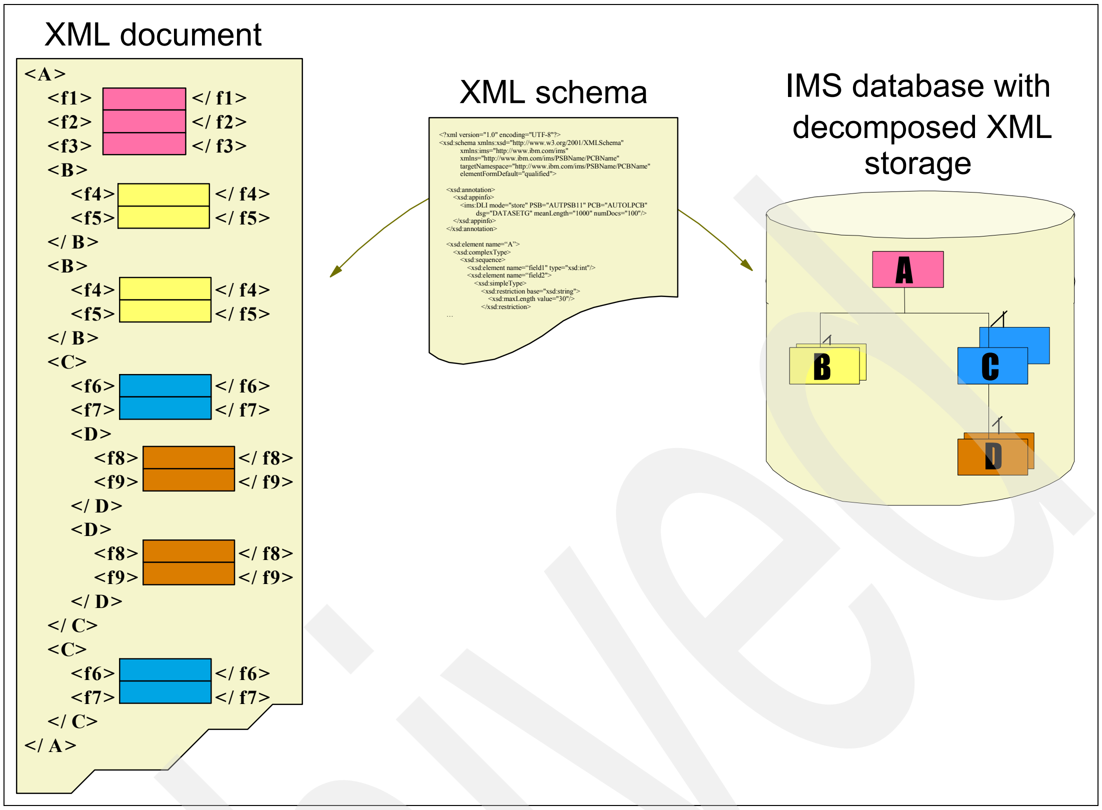


Figure 9-2 Mapping of XML to IMS DB

The transformation of data between an XML document and the IMS database record is performed by the *retrieveXML* and *storeXML* functions. An example of this is shown in Figure 9-3.

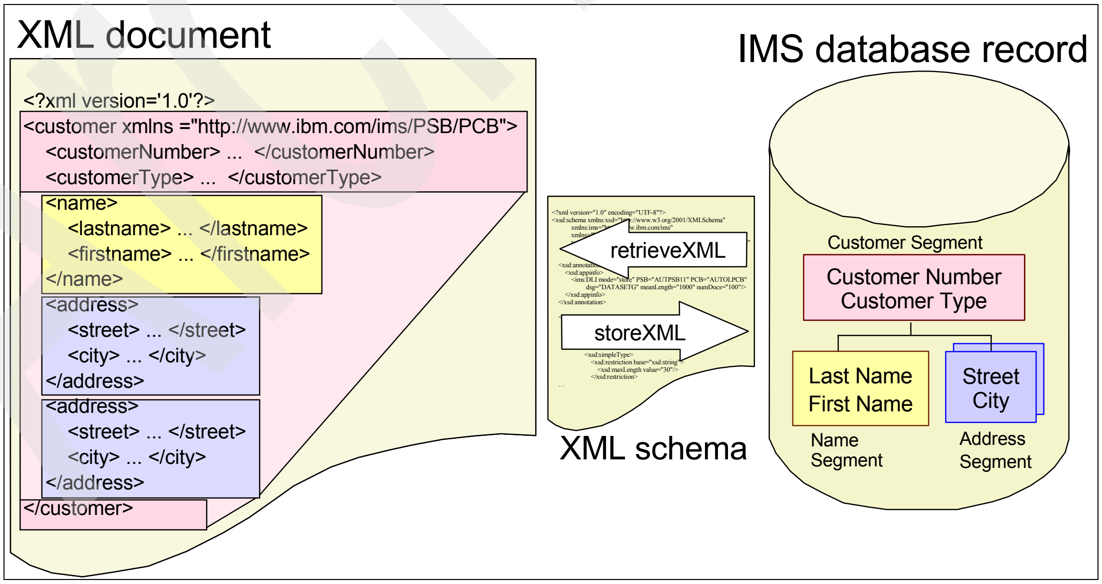


Figure 9-3 The transformation of information from XML to IMS DB

An example of a generated XML schema is shown in Example 9-1 on page 150.

Example 9-1   A generated XML schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
       xmlns="http://www.myNamespace.net"
       targetNamespace="http://www.myNamespace.net"
       elementFormDefault="qualified">

   <xsd:element name=“A”>
     <xsd:complexType>
       <xsd:sequence>
         <xsd:element name=“field1" type="xsd:int"/>
         <xsd:element name=“field2">
           <xsd:simpleType>
             <xsd:restriction base="xsd:string">
               <xsd:maxLength value="30"/>
             </xsd:restriction>
           </xsd:simpleType>
         <xsd:element name=“B” minOccurs=“0” maxOccurs=“unbounded”>
         </xsd:element>
         <xsd:element name=“C” minOccurs=“0” maxOccurs=“unbounded”>
           <xsd:element name=“D” minOccurs=“0” maxOccurs=“unbounded”>
           </xsd:element>
   …
</xsd:schema>
```

An XML document may also be stored in a subset of a database record, from a non-root segment and its dependents, as shown in Figure 9-4. The subset must be a segment and its dependents. The highest level segment in the subtree is the XML root element segment. All of its dependents are included in the XML document. Of course, the XML root element segment may be the database root segment. In this case, the XML document would correspond to the entire database record.

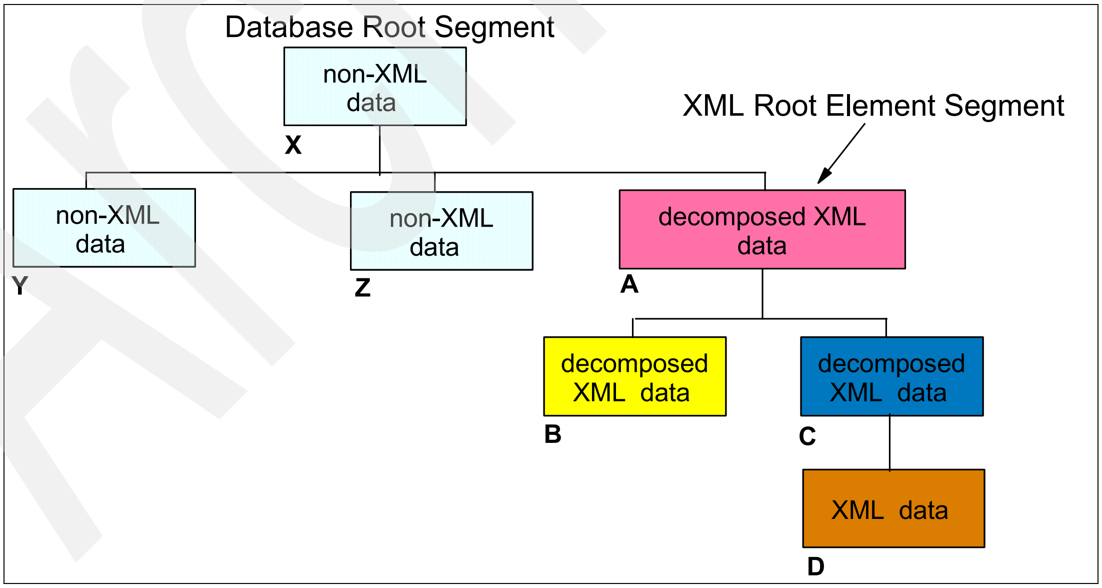


Figure 9-4   An XML document may be stored in a subset of a database record

9.2.3 Intact storage

Information from an XML document can be inserted, retrieved or deleted from the document intact into an IMS database. This can be a new or existing database. The data is stored as XML, and includes both the XML tags and the data fields. The data in these databases can only be manipulated by applications that are knowledgeable of XML, since traditional applications will not understand the XML format.

For intact storage, the data is stored in segments based on the number of bytes in the document, without regard to XML tags. There are two segment types per document, one "root" segment and as many overflow segments as necessary to store the entire document. There is no correspondence between XML elements and database segments. The root segment contains a fixed number of bytes of XML data. Each overflow segment also contains a fixed, but different, number of bytes. The illustration in Figure 9-5 shows that elements from the XML document may span database segments.

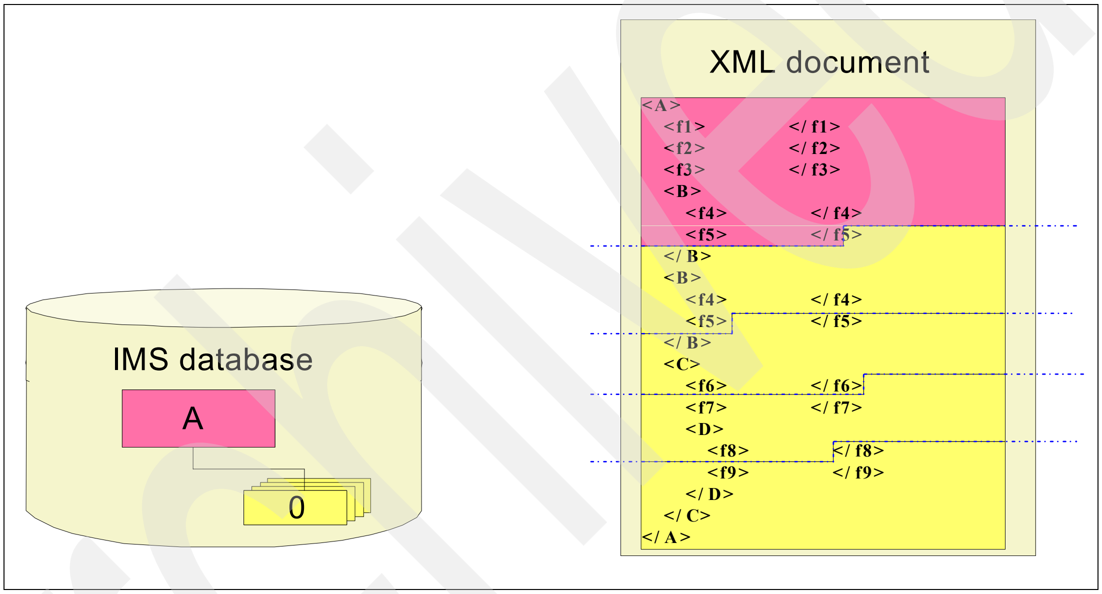


Figure 9-5 Intact storage of an XML document into an IMS database

We will see later that the IMS database may also contain special segments used for secondary indexes.

When intact storage is used, the XML document is not parsed or validated when it is stored in an IMS database unless secondary indexing is used. The XML document is stored in the database as Unicode. Traditional IMS applications do not process XML or Unicode, so they are not able to process XML documents when they are stored intact. These documents are accessible to applications which can process XML and Unicode. Data is not converted to traditional IMS types, but is stored as Unicode. These documents are only accessible by XML-enabled applications.

Secondary indexing for intact XML documents

When an XML document is stored intact in an IMS database, one or more *side segments* may be used for secondary indexing. A side segment is a source segment, which has had its segment data converted to traditional data types. Data may come from either an element or an attribute of the XML document - this data remains in the document root or overflow segments. The target segment is segment above source segment in the database hierarchy. In Figure 9-6 on page 152, the S1, S2 through Sn segments contain the data used for the secondary indexes. The target of the secondary index is segment A.

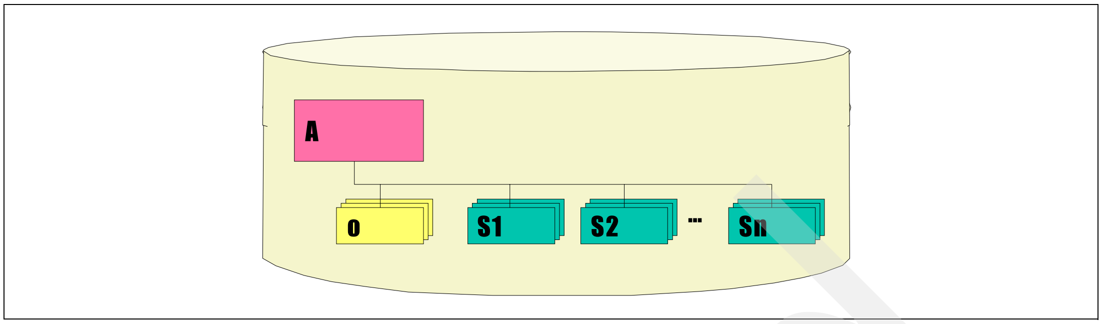


Figure 9-6 Secondary indexing for intact stored XML documents

An XML document may also be stored in a subset of the database record. In this situation, the XML root element segment is different to the database root segment. This is shown in Figure 9-7.

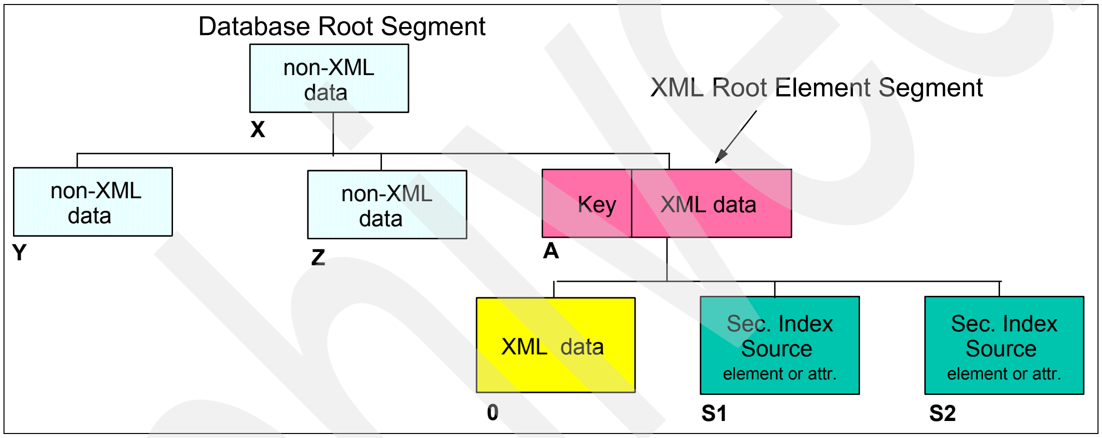


Figure 9-7 An intact XML document may be stored in a subset of a database record

Mixing decomposed and intact storage

A schema may include both decomposed and intact storage. The intact part must be a complete subtree of elements in the XML element hierarchy, as shown in Figure 9-8 on page 153.

This means that the intact segments will be a subtree of database segments containing XML data.

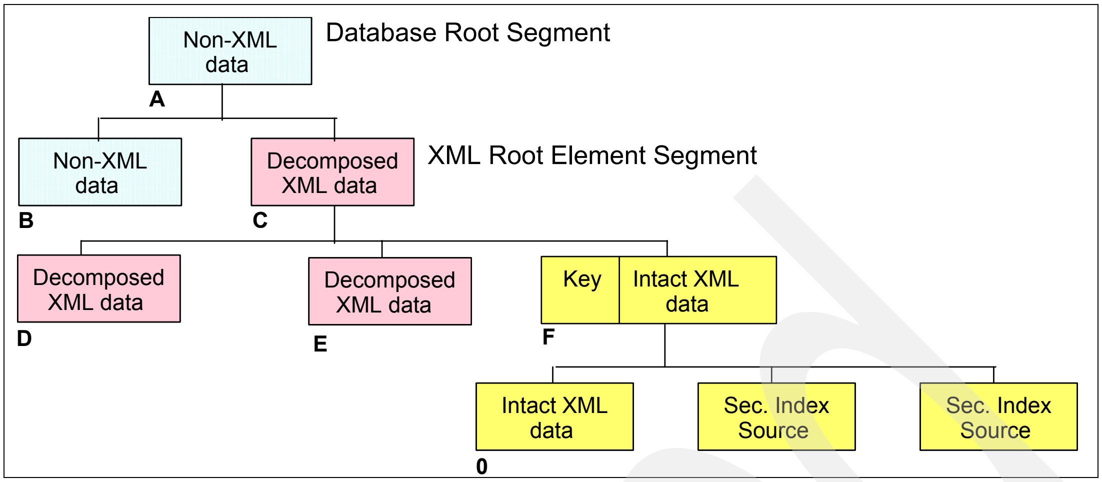


Figure 9-8 Mixing decomposed and intact storage of an XML document

9.2.4 DLIModel Utility

The DLIModel Utility is used to create the XML schema for an XML document. This is a new function of the utility for IMS Version 9. The schemas are created from PSBs, DBDs, and COBOL copybook XMI files, as shown in Figure 9-9.

The DLIModel Utility was first introduced with IMS Version 7, to create IMS Java metadata classes from PSBs and DBDs. The utility also incorporates additional field information from XMI input files that describe COBOL copybook members, since many DBDs do not completely describe the fields in the database. The utility can be run as a z/OS batch job, or from a command prompt under UNIX System Services.

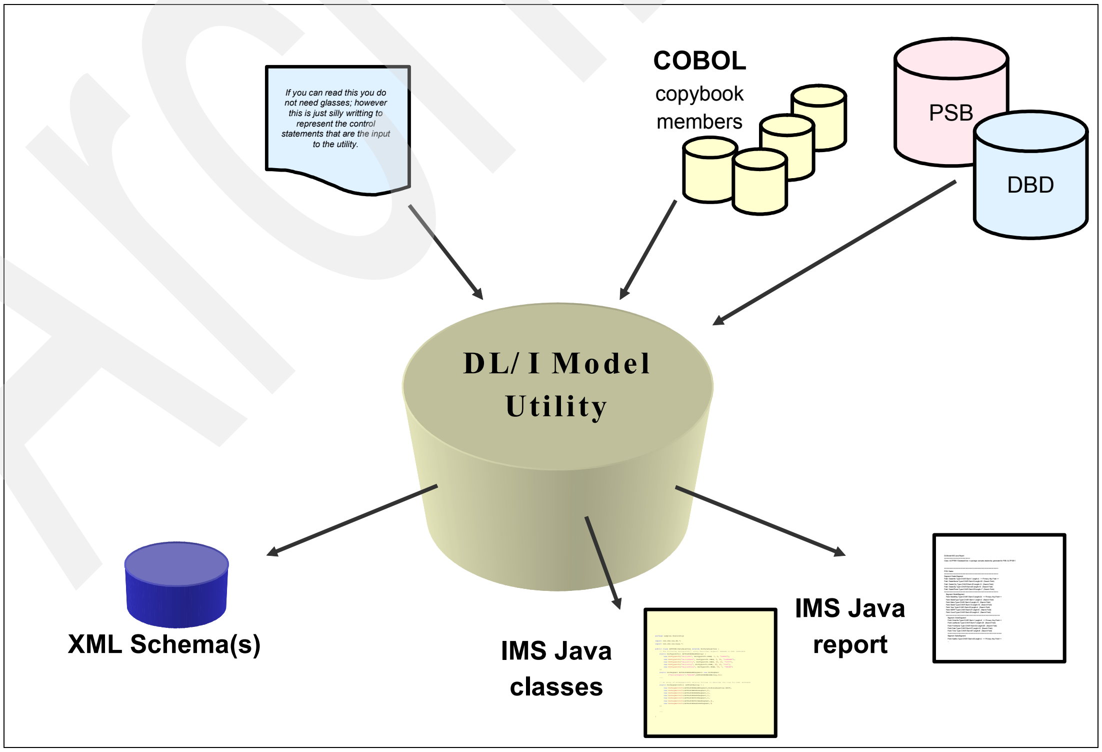

If you can read this you do not need glasses; however this is just silly writting to represent the control statements that are the input to the utility.
COBOL
copybook
members
PSB
DBD
DL/I Model
Utility
XML Schema(s)
IMS Java
classes
IMS Java
report

Figure 9-9 The DLIModel Utility

There are a number of control statements that can be specified for this utility, with new options for the creation of XML schemas. The OPTIONS statement has new parameters, indicating whether an XML schema is to be generated (`GenXMLSchemas=YES|NO`), as well as the path where the schema is to be stored (`XMLSchemaPath=path`).

The PCB statement also has new parameters:

- `GenXMLSchema=RETRIEVE|STORE|NO`
- `XMLRootElement=segmentname`

The SEGM statement has a new parameter:

- `XMLStorageType=DECOMPOSED|INTACT`

And the FIELD statement has a new parameter:

- `XMLType=ELEMENT|ATTRIBUTE|NONE`

GenXMLSchema=STORE checks for the following PCB limitations:

- The PCB must be based on a physical DBD
- The PCB PROCOPT includes I or A
- No segments in the hierarchy may be logical children
- The hierarchy must be based on a primary access path
  - Not a secondary index access path (PROCSEQ)
- All segments must have their key fields present
  - that is, they have not been excluded by field-level sensitivity
- No segments may have PROCOPT=K.

Example 9-2 on page 155 is an example of a the control statements that could be used to generate a schema using the DLIModel Utility. The utility is also used to create metadata for IMS Java. The additional keywords and parameters used to create a schema are:

- GenXMLSchemas=yes on the OPTIONS statement

  This specifies that XML schemas will be generated.

- GenXMLSchema=store of the PCB statement for PCB1

  This specifies that an XML schema will be generated for this PCB and that a check will be made to ensure that the PCB may be used for storing XML documents.

- XMLRootElement=DEALER

  This specifies that the DEALER segment will be the XML root element segment.

- XMLType=attribute on the FIELD statement for DLRNO

  This specifies that the data in the DLRNO field will be mapped to an attribute in the XML document.

Two defaults are documented in comments:

- XMLStorageType=decomposed is the default on the SEGM statement for the DEALER segment.

  This specifies that decomposed storage is used for the data mapped to this segment.

- XMLType=element is the default on the FIELD statement for DLRNAME.

  This specifies that the data in the DLRNAME field will be mapped to an element in the XML document.

The CITY, ZIP, and PHONE fields also use the default for XMLType. They will be mapped to elements. Defaults are used except on the FIELD statement for MAKE. XMLType=none specifies that this field in the database is not mapped to the XML document.

FIELD statements are not required if you have a COBOL copybook XMI file which describes the fields. This is shown in the last SEGM statement. A FIELD statement is used in the example to specify that the COLOR field is not mapped to the XML document. This is done by specifying XMLType=none. A COBOL copybook XMI file may be generated from a COBOL copybook by using WebSphere Studio Application Developer Integrated Edition (WSADIE).

There are other options for statements. These are documented in the *IMS Version 9: Utilities Reference: System,* SC18-7834 manual in its explanation of the DLIModel Utility. For example, one may code FIELD statements which completely describe fields including their offsets and sizes in segments. One would do this if the fields were not defined in the DBD or an XMI file.

Example 9-2   DLIModel control statements used to generate an XML schema

```
OPTIONS PSBds=IMSVS.TEST.PSBSRC
        DBDds=IMSVS.TEST.DBDSRC GenJavaSource=YES
        GenXMLSchemas=yes
PSB psbName=f402psb JavaName=AUTPSB11
PCB pcbName=PCB1 JavaName=Dealer GenXMLSchema=YES
        XMLRootElement=DEALER

// Physical Segments for DEALERDB
SEGM DbdName=DEALERDB SegmentName=DEALER JavaName=DealerSegment
          //defaults to XMLStorageType=decomposed
FIELD Name=DLRNO   JavaType=INTEGER JavaName=DealerNo
          XMLType=attribute
FIELD Name=DLRNAME JavaType=CHAR JavaName=DealerName
          //defaults to XMLType=element
FIELD Name=CITY    JavaType=CHAR JavaName=DealerCity
FIELD Name=ZIP     JavaType=CHAR JavaName=DealerZip
FIELD Name=PHONE   JavaType=CHAR JavaName=DealerPhone
SEGM DbdName=DEALERDB SegmentName=MODEL
      JavaName=ModelSegment
FIELD Name=MODTYPE JavaType=CHAR JavaName=ModelType
FIELD Name=MODKEY JavaType=CHAR JavaName=ModelKey
FIELD Name=MAKE   JavaType=CHAR JavaName=Make
      XMLType=none
FIELD Name=MODEL  JavaType=CHAR JavaName=Model
FIELD Name=YEAR   JavaType=Date TypeQualifier=yyyy
      JavaName=Year
FIELD Name=MSRP   JavaType=PACKEDDECIMAL
      TypeQualifier=S999999V99 JavaName=MSRP
FIELD Name=COUNT  JavaType=SMALLINT JavaName=ModelCount

SEGM DBDName=DEALERDB SegmentName=STOCK JavaName=StockSegment
       CobolXMI=imsjava/mdlex4/AutoStock.xmi
FIELD Name=COLOR   JavaType=CHAR JavaName=COLOR
      XMLType=none
```

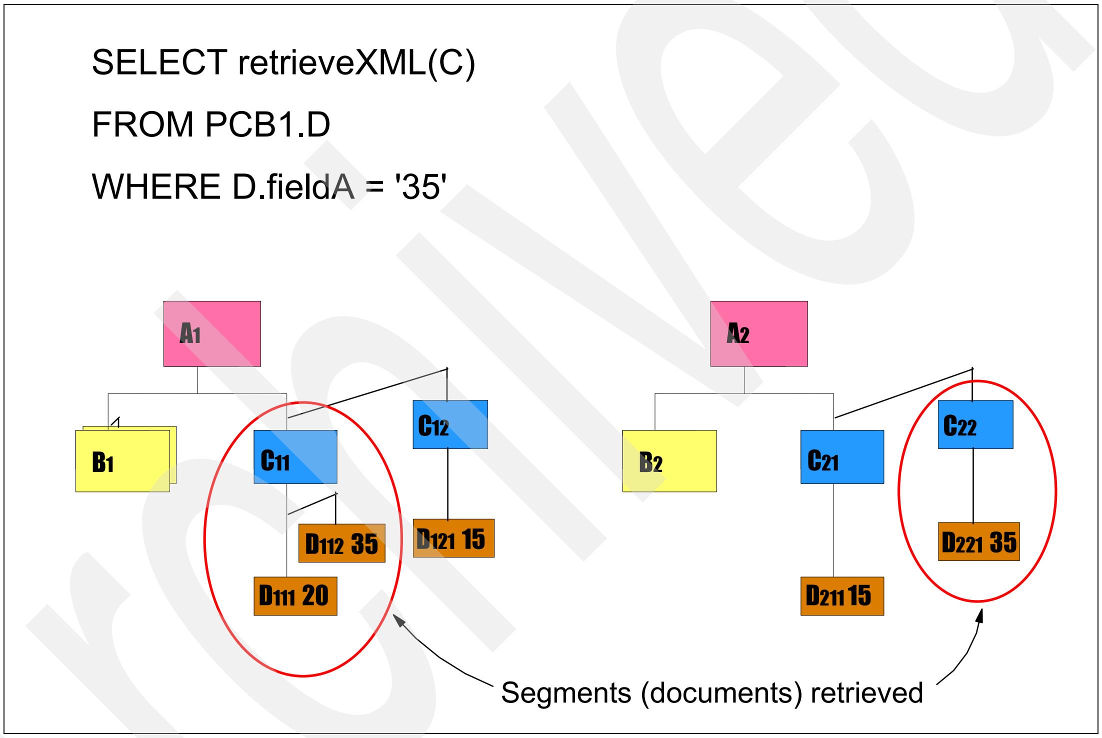
9.3 JDBC interface

The Java Database Connectivity interface (JDBC) has been extended with two user defined functions (UDF). They are *retrieveXML()*, and *storeXML().*

The retrieveXML statement is included in the select list of a JDBC SELECT operation, to retrieve XML document(s) from an IMS database. The information is returned as a character large object (CLOB) in the returned JDBC ResultSet per row.

The storeXML() statement is included in a JDBC INSERT statement, which stores an XML document in an IMS database. The position of the record in the database is determined by the WHERE clause.

An example of the use of RetrieveXML() is shown in Figure 9-10. The XML root element segment is C. It is specified in retrieveXML(C). The WHERE clause requests documents which have a value of 35 in a fieldA instance. Segment D112 meets this qualification, so the document stored in instance C11 of the C segment and its dependents is returned. Segment D221 also meets this qualification, so the document stored in instance C22 and its dependents is returned.



```
SELECT retrieveXML(C)
FROM PCB1.D
WHERE D.fieldA = '35'
```

Segments (documents) retrieved

Figure 9-10   RetrieveXML()

An example of the use of storeXML() is shown in Figure 9-11 on page 157. The XML root element segment is C. The INSERT statement requests that the document be stored under instance of segment A which has a value of 62 in fieldA. The document is stored in segments C22, D221, and D222. You may notice that the WHERE clause is not like standard SQL which does not use a WHERE clause with INSERT.

```
INSERT INTO PCB1.C (storeXML())
VALUES (?)
WHERE A.fieldA = '62'
```

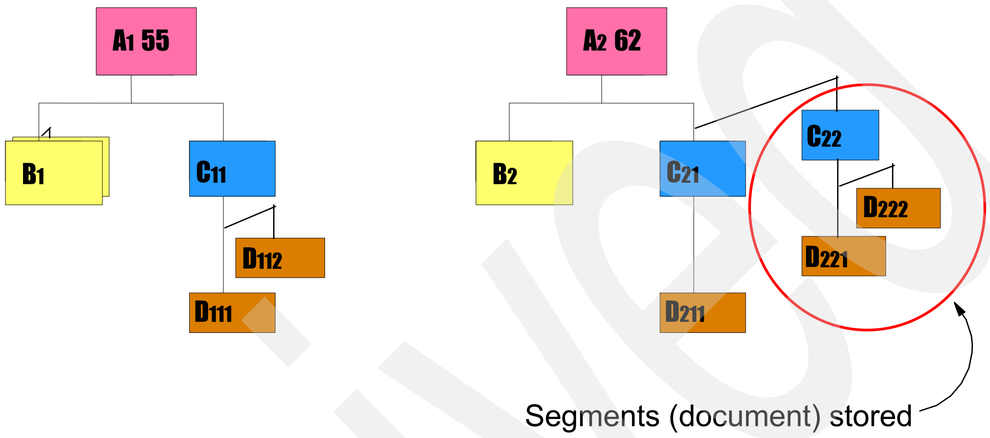


Figure 9-11 StoreXML()

9.4 XML database summary

IMS XML DB allows you to view existing or new IMS databases as XML databases, offering easy exchange of data between IMS DBs and XML documents. IMS XML DB offers two storage modes:

- Decomposed storage
  - Ideal for data-centric XML documents requiring full searching or legacy interoperability
  - Existing IMS databases may be used to create XML documents
  - Existing IMS applications are unaffected
- Intact storage
  - Ideal for document-centric XML documents or requiring increased retrieval performance
  - Search capability only through use of side segments and secondary indices

The DLIModel utility has been extended to:

- Assist in building XML metadata mapping to IMS
- Generate XML schema definitions for XML documents

The IMS Java JDBC interface has been extended to:

- Include a retrieveXML keyword for XML retrieval
- Include a storeXML keyword for XML insertion

10

IMS Java enhancements

IMS Version 9 provides enhancements for the IMS Java environment. In this chapter, we focus specifically on IMS Java enhancements as well as remote database access which uses WebSphere Application Server for z/OS Version 5.0 and WebSphere Application Server on any other platforms. IMS Java enhancements also include XML storage in IMS Databases, and is dealt with exclusively in Chapter 9, “XML storage in IMS databases” on page 145.

In this chapter we provide an overview of the IMS Java environment first, and then we introduce IMS Java enhancements in IMS Version 9:

- IMS Java environment overview
- IMS-DB2 interoperability
- Symbolic checkpoint and restart in a JBP region
- Java-COBOL interoperability
- IMS Java GSAM support for JBP applications
- JDBC and SQL enhancements
- Remote database access services using WebSphere Application Server

You can find IMS Java updates for each IMS version at the following URL:

http://www.ibm.com/software/data/ims/imsjava/javenhance.html

10.1 IMS Java environment overview

Since IMS Version 7, IMS has provided the capability to allow Java programs to access IMS resources. The supported environments include IMS application programs running in a Java message processing region (JMP) or a Java batch processing region (JBP), as well as Java programs running in other environments such as DB2 stored procedures, JCICS, and WebSphere Enterprise Java Beans (EJBs). This is shown in Figure 10-1.

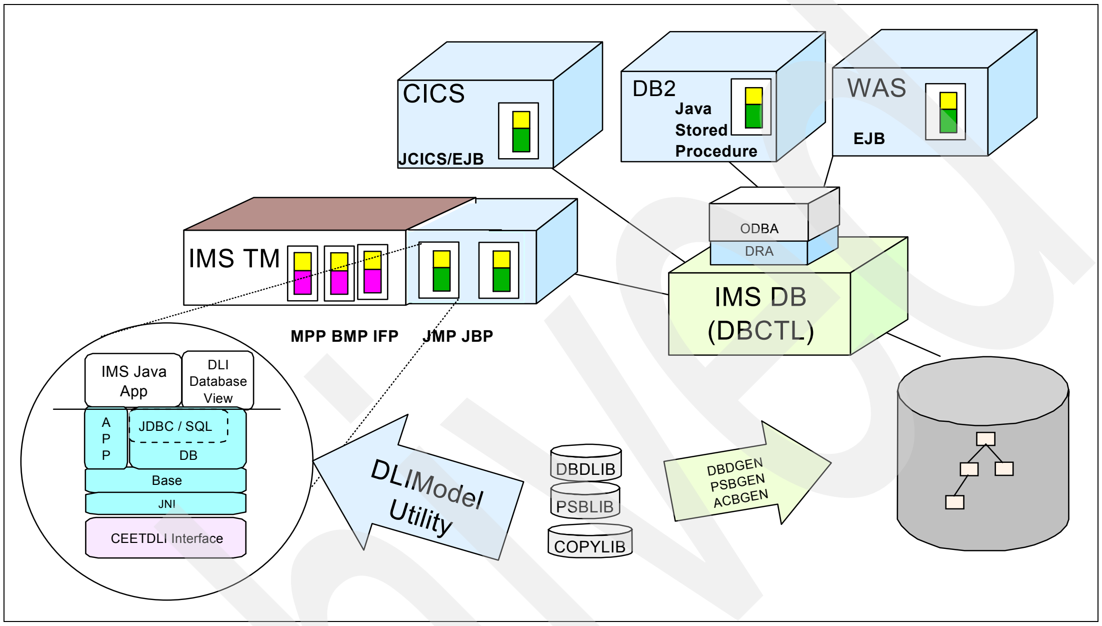


Figure 10-1 Java in the IMS environment

IMS Java is delivered in a layered set of class libraries. This concept allows for the high-level classes to focus strongly on ease-of-use, and the lower-level classes to provide complete access to IMS services and serve as the implementation vehicle for the higher-level classes. The diagram in Figure 10-2 shows the layered set of class libraries.

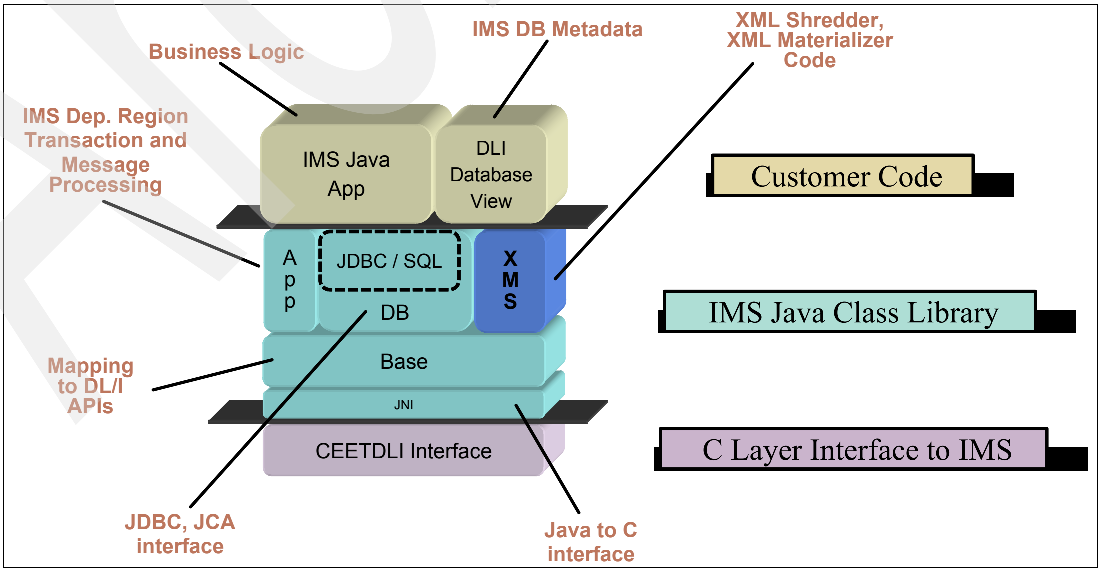


Figure 10-2 The layered set of class libraries used for Java with IMS

Access to IMS data from a Java program is provided by Java Database Connectivity (JDBC), which is the standard Java API for accessing data. IMS JDBC support allows SQL statements to be used to access an IMS database, and processes the data returned by the database. Execution of query statements establishes and opens a connection to the database, executes the query to obtain results, processes the results, and then closes the connection. IMS Version 9 no longer ships the IMS Java API specification on z/OS, but this specification is available in the Web.

10.2 IMS and DB2 interoperability

IMS Version 9 has extended the support for Java programs running in a JMP or JBP region. These programs are now able to access DB2 databases, as well as IMS databases. This support is provided as part of the base IMS Version 9, and through the service process for IMS Version 8 with UQ80615.

The support for access to DB2 uses the DB2 resource recovery service attach facility (RRSAF), and requires the **SSM** and **RRS** parameters in the IMS control region. With RRSAF, the dependent region builds an attachment thread to DB2 UDB for z/OS using Resource Recovery Services (RRS). RRS coordinates the commits of the updates that the JMP or JBP applications make to both IMS and DB2 resources. IMS is a participant, not the coordinator, of these updates and commits.

Unlike other dependent regions, JMP and JBP regions do not use the external subsystem attach facility (ESAF) which MPP, BMP, IFP, DLI, DBB, and DBCTL BMPs programs uses when they access external subsystems such as DB2 UDB for z/OS.

Only one active DB2 connection at a time is supported, although future support for multiple connections is a known requirement. Access to DB2 through JDBC and SQLJ is supported. DB2 UDB for z/OS provides different JDBC drivers:

- JDBC/SQLJ driver for DB2 for OS/390 and z/OS with JDBC 2.0
- JDBC/SQLJ driver for DB2 for OS/390 and z/OS with JDBC 1.2
- DB2 Universal JDBC driver

The JDBC/SQLJ drivers are type 2 JDBC drivers which allow access to DB2 databases only when IMS is on the same z/OS image as DB2 UDB for z/OS. DB2 Universal JDBC driver supports Universal Driver type 4 connectivity which allows access to DB2 databases from IMSs that are on different z/OS images from one DB2 UDB for z/OS is on. It also provides the type 2 implementation.

10.2.1 Setting up the IMS to DB2 connection using RRSAF

There are several parameters in your IMS system that need to be specified to connect your IMS system to DB2 using RRSAF:

- In your IMS.PROCLIB, create an SSM member which specifies the connection to DB2. The member name must follow the same naming conventions as when attaching DB2 with the traditional external subsystem attach facility (ESAF), like xxxxnnnn: xxxx is of IMSID=, nnnn is of SSM=. Refer to *IMS Version 9: Installation Volume 2: System Definition and Tailoring,* GC18-7823 for more detail of SSM PROCLIB member. The following three parameters must be specified in this member:

  ```
  SST=DB2,SSN=db2name,COORD=RRS
  ```

- Add a DFSJVMMS member to your IMS.PROCLIB, specifying a DB2 JDBC driver path, to the ZIP file and to the ZIP file name, to the trusted middleware path, for example:

```
-Dibm.jvm.trusted.middleware.class.path=>
/usr/lpp/db2/db2710/classes: >
/usr/lpp/db2/db2710/classes/db2j2classes.zip
```

- Add a DFSJVMEV member to your IMS.PROCLIB, specifying a DB2 JDBC driver path to the LIBPATH environment variable, for example:

  `LIBPATH=/usr/lpp/db2/db2710/lib`

- In your IMS control region startup parameters, add the subsystem name with the **`SSM=name`** parameter, as well as telling IMS to use Resource Recovery Services through the **`RRS=Y`** parameter.
- For the DFSJMP or DFSJBP procedure of the regions that access DB2, add the DFSDB2AF DD pointing to the DB2 libraries, which must be authorized.

10.2.2 Accessing DB2 UDB for z/OS databases from JMP or JBP applications

Accessing DB2 UDB for z/OS data from a JMP or JBP application is similar to accessing IMS data. When writing a JMP or JBP application that accesses DB2 data, consider both the differences from IMS database access and the differences from accessing DB2 data in other environments:

- You can have only one active DB2 UDB for z/OS connection open at any time.
- For type 2 JDBC drivers, you must use the default connection URL in the application program, for example, `jdbc:db2os390:` or `db2:default:connection`.
- For type 4 JDBC drivers, you can use a specific connection URL in the application program.
- To commit work, you must use this method:
  - `IMSTransaction.getTransaction().commit()`
    - For JMP, this method commits all work; SQL call and connection closures.
    - For JBP, this method commits SQL calls.
- To roll back work, you must use this method:
  - `IMSTransaction.getTransaction().rollback()`
- Because RRS is the coordinator, you cannot use these methods:
  - `Connection.setAutoCommit()`
  - `Connection.commit()`
- You must always call `IMSTransaction.getTransaction().commit()` after closing a connection to DB2 UDB for z/OS to commit the connection closure.
- You cannot use COBOL to access DB2 UDB for z/OS in a JMP or JBP region.

JMP application that accesses DB2 UDB for z/OS data

When a JMP application accesses only IMS data, it needs to open a database connection only once to process multiple transactions. However, a JMP application that accesses DB2 data must open and close a database connection for each message processing, with `IMSTransaction.getTransaction().commit()` method.

JBP application that accesses DB2 UDB for z/OS data

Like a BMP application that accesses IMS data, a JBP application that accesses DB2 data connects to a database, performs database processing, periodically commits, and disconnects from the database at the end of the application. However, the application accessing DB2 data must issue a final commit after closing the database connection. Thus, application issues `IMSTransaction.getTransaction().commit()` as periodic commit, closes the connection, then issues `IMSTransaction.getTransaction().commit()` again.

10.3 IMS Java symbolic checkpoint and restart for JBP

IMS Version 9 extends the IMS symbolic checkpoint and restart capability to the IMS Java environment. It is supported only for JBP applications, and allows them to take application checkpoints, so they can restart from the last checkpoint after a failure. The program can save critical application Java objects as a part of the checkpoint processing. IMS restores those objects at restart, as well as the database positions at the time of the checkpoint.

The support for IMS Java symbolic checkpoint and restart is provided through two new methods in the IMSTransaction class, checkpoint() and restart(). These methods perform analogous functions to the existing IMS system service calls: (symbolic) CHKP and XRST, respectively.

The `IMSTransaction.restart()` method is used at the initial application start, to notify IMS to enable checkpoint and restart capability. At a program restart, the program must provide the checkpoint id to use from either a restart call parameter, similar to providing the id to the XRST call in IMS, or the CKPTID= parameter through the JBP region JCL. If both methods are used in the same program, the CKPTID= parameter takes precedence. The method will obtain the checkpointed objects from an earlier checkpoint.

The `IMSTransaction.checkpoint()` method allows the program to invoke an IMS checkpoint and supply a checkpoint id. The program can optionally provide String checkpoint id, or an object (`com.ibm.ims.application.SaveArea`) containing the Java objects whose state is to be saved with the checkpoint.

The IMS Java symbolic checkpoint and restart capability is based on the existing symbolic checkpoint and restart facility in IMS, as provided for BMP applications. It implements a Java interface to the traditional IMS checkpoint and restart facility and shares the functional capabilities of that facility.

Symbolic checkpoint and restart support in IMS Java allows user applications in JBP regions to:

- Take application checkpoints, and to restart from the last checkpoint after a failure.
- Save critical application Java objects as part of the checkpoint, and to restore those objects at restart.
- Save database positions as part of the checkpoint and verify repositioning of those databases at restart.

> **Note:** IMS Java is adding the capability to do symbolic checkpoints and restarts in a JBP much like the capability that has been available for years in an IMS BMP. What is being documented is the set of calls to do this. The considerations for the use of these calls in a JBP are the same as the considerations for the same calls in a BMP.

10.3.1 Checkpoint call formats

The formats of the Java checkpoint call provided by IMSTransaction class are:

- **`public void checkpoint() throws IMSException`**
- **`public void checkpoint(SaveArea objectsToSave) throws IMSExeption`**

- **`public void checkpoint(String chkpId) throws IMSException`**
- **`public void checkpoint(String chkpId, SaveArea objectsToSave) throws IMSException`**

All of these checkpoint methods are used to perform a checkpoint call for JBP application that uses symbolic checkpoints. JBP application using checkpoint and restart function calls these methods at intervals to cause checkpoints to be written to the IMS log. The difference is:

- The first method without parameters is used in the cases where the application needs to take checkpoint without having to specify a checkpoint id and also without having to save any application objects, which is described as SaveArea object later. When using this method for taking checkpoints, the application should use the 4-character constant "LAST" as the checkpoint id to restart from.
- If the application needs to save application objects, it should use one of the following forms of the checkpoint method:
  - `checkpoint(SaveArea objectsToSave)`
  - `checkpoint(String chkpId, SaveArea objectsToSave)`
- If the application needs to specify a checkpoint id, it should use one of the following forms of the checkpoint method:
  - `checkpoint(String chkpId)`
  - `checkpoint(String chkpId, SaveArea objectsToSave)`

Checkpointed objects may be instances of standard Java classes (for example, Integer, String, and so on), or user defined application classes (for example, Customer, Part).

The checkpoint() method uses Java serialization to prepare the objects for checkpointing by IMS. This means that all objects that are passed in the SaveArea must implement the Java serializable interface. Additionally, each invocation of checkpoint() may specify different numbers or types of objects in its SaveArea parameter. Note, however, that the application must be able to interpret the returned objects at restart time. If different objects are to be saved on each checkpoint, consider including descriptive information in the saved data (i.e., in an object). A size limit is set on the objects to be checkpointed. The default total length of all the objects in serialized form may not exceed 64KB. An IMSException is thrown if the objects in a checkpoint exceed this limit. The default total length of all objects in serialized form is 64K which may be changed by issuing the **`setCheckpointSizeLimit()`** method.

The other methods available for the **`com.ibm.application.SaveArea`** are:

`addObject`	Adds Java objects to be checkpointed
`getObject`	Gets the Java objects back
`verifyPCBPosition`	Verifies the database position after restart
`setCheckpointSize`	Sets the maximum possible checkpoint length for the application to use.

10.3.2 Restart call formats

The formats of the Java restart call provided by IMSTransaction class are:

- **`public SaveArea restart() throws IMSException`**
- **`public SaveArea restart(String chkpId) throws IMSException`**

These methods are equivalent to the IMS XRST call. As is the case for non-Java programs, the ID specified can be one of following:

- 1 to 8 character checkpoint ID

- 14 character time stamp ID from IMS message DFS0540I
- The literal “LAST”

These restart method is used in two ways:

- At normal program start, these methods just tell IMS that subsequent checkpoint calls are symbolic checkpoint calls. You may use restart method that takes no parameters, or may use restart method with null checkpoint id parameter. The CKPTID= in the JBP region procedure also must be null.
- If the application is restarting from a checkpoint, pass a restart checkpoint id either with `restart(String id)` method, or with `restart()` method and valid CKPTID= parameter in the JBP region procedure. If the String id parameter and the CKPTID= parameter are set to a non-null value, the CKPTID= parameter in the EXEC statement takes precedence.

The invocation of the restart() method should be done only once and should be early in the execution of the program. It does not need to be the first call in the program but it should precede any invocation of the IMSTransaction.checkpoint method. The restart method call are not within the scope of a restart.

A restart is performed operationally in a very similar manner to existing non-Java BMP restarts. The job is resubmitted, supplying a checkpoint ID in one of two ways, either as a restart call parameter or in a JCL parameter. Both approaches require appropriate user operational procedures to manage master terminal messages, checkpoint ids, and so on. The first approach requires a suitable application program design to obtain the checkpoint ID by some means, and to provide it to the restart() invocation. These are user responsibilities that are the same as for non-Java IMS programs that use the checkpoint and restart facility. The second approach is perhaps an easier mechanism and takes advantage of specifying the ID in the CKPTID= field of the PARM= parameter on the EXEC statement in the program’s JCL.

During the restart, a SaveArea object is returned that contains the Java objects that were saved with the checkpoint. The objects are contained in the returned SaveArea object in the same sequence as they were saved by a checkpoint method. It is an application responsibility to retrieve these objects from the SaveArea object, to cast them to the correct application object types, and to prepare the application for continued execution. If no application objects were saved with this checkpoint, the method returns a null reference is returned.

Under the covers, the SaveArea class issues the system service call INQY with the FIND subfunction using the PCB name to get the PCB address. This address is used to check the PCB status code to determine if the database has been repositioned properly after the restart() call.

IMS repositions the application databases if possible. It is an application responsibility, however, to follow the restart with one or more invocations of the verifyPCBPosition method to confirm that repositioning was successful for each significant PCB.

The **`SaveArea.verifyPCBPosition`** method verifies that IMS was able to reposition a database(s) during restart. There are two variations of this call:

- specify the name(s) of the PCB(s) as String with this format:

  ```
  public boolean verifyPCBPosition(String pcbName) throws IMSException
  ```

- specify the name(s) of the PCB(s) as an array of Strings parameter with this format:

  ```
  public boolean verifyPCBPosition(String [ ] pcbNames) throws IMSException
  ```

The design of this support contains an arbitrary default upper limit on the total serialized length of the checkpointed objects, which is set at 64KB. The limit exists because at restart time the application (and the IMS Java checkpoint and restart code) do not know the length of the serialized object data on the IMS log. This necessitates making an assumption about the length when preparing a receiving memory extent for the IMS XRST call. The assumed length must be at least as large as the data on the log. The arbitrary upper limit could be increased to more than 64KB, but the trade-off is that at restart time IMS Java must create a byte array of this length in the dependent region. An application program obtains the SaveArea object by invoking the static **`SaveArea.getSaveArea()`** method.

Since the IMS Java restart() call is converted to an IMS XRST, an attempt is made to reposition all databases to the position that was held when the last checkpoint was taken. This is done by including each PCB and PCB key feedback area in the checkpoint record. Issuing XRST causes the key feedback area from the PCB in the checkpoint record to be moved to the corresponding PCB in the PSB that is being restarted. Then IMS issues a GU call, qualified with the concatenated key (using the C command code), for each PCB that held a position when the checkpoint was taken.

After the XRST call, the PCB reflects the results of the GU repositioning call, not the value that was present when the checkpoint was taken. The GU call is not made if the PCB did not hold a position on a root or lower-level segment when the checkpoint was taken. A GE status code in the PCB means that the GU for the concatenated key was not fully satisfied. The segment name, segment level, and key feedback length in the PCB reflect the last level that was satisfied on the GU call. A GE status code can occur because IMS is unable to find a segment that satisfies the segment search argument that is associated with a Get call for one of the following reasons:

- The call preceding the checkpoint call was a DLET call issued against the same PCB.

  In this case, the position is correct because the not-found position is the same position that would exist following the DLET call. The functionality and operational characteristics of using the symbolic checkpoint and restart should be similar to those of IMS applications such as BMPs that use the existing checkpoint and restart facility of IMS.

- The segment was deleted by another application program between the time the program terminated abnormally and the time it was restarted.

  A GN call issued after the restart returns the first segment that follows the deleted segment or segments.

The above explanation assumes that position at the time of checkpoint was on a segment with a unique key. XRST cannot reposition to a segment if that segment or any of its parents have a non unique key.

For a DEDB, the GC status code is received when position is not on a segment but at a unit-of-work (UOW) boundary. Because the XRST call attempts to reestablish position on the segment where the PCB was positioned when the symbolic checkpoint was taken, the XRST call does not reestablish position on a PCB if the symbolic checkpoint is taken when the PCB contains a GC status code.

10.4 IMS Java GSAM support for JBP applications

With IMS Version 9, JBP applications can process GSAM databases. This support, along with symbolic checkpoint and restart support for JBP applications, gives JBP applications more of the functionality of BMP applications.

GSAM databases are useful if you need a JBP application that accesses z/OS data sets to use the IMS symbolic checkpoint call. Because GSAM databases are supported in a DCCTL environment, you can also use them when a JBP application needs to process sequential non-IMS data sets. From a JBP application, you can open, read, write, and close a GSAM database. There are three new IMS Java classes for processing GSAM databases:

- com.ibm.ims.db.GSAMConnection
- com.ibm.ims.db.GSAMRecord
- com.ibm.ims.application.RSA.

See the IMS Java API Specification for details on these classes.

10.5 Java-COBOL interoperability

From within JMP or JBP applications, you can call an object-oriented (OO) COBOL application with IMS Enterprise COBOL for z/OS and OS/390 Version 3 Release 2. And, you can build an OO COBOL application containing a main routine that can invoke Java routines. This functionality is provided through IMS Version 7 (with UQ75641) and Version 8 (with UQ74520). Refer to *IMS Version 9: IMS Java Guide and Reference,* SC18-7821, for more details.

10.5.1 Enterprise COBOL as a back-end application in a JMP or JBP region

From an OO COBOL class, the Enterprise COBOL compiler generates a Java class definition with native methods and the object code that implements the native methods. After compiling the class, you can create an instance and invoke the methods of the class from a Java program that runs in a JMP or JBP region.

When Java is the front-end language, you must perform all message-queue and message-synchronization processing in Java. In the back-end application, you can access IMS databases by either using Java or calling a COBOL routine.

10.5.2 Enterprise COBOL as a front-end application in a JMP or JBP region

You can write an application for an IMS Java dependent region entirely with OO COBOL, but a more likely use for a front-end COBOL application is to call a Java routine from a COBOL application.When running within the JVM of an IMS Java dependent region, Enterprise COBOL run-time support automatically locates and uses this JVM to invoke methods on Java classes.

This COBOL application with a main routine that runs in a JMP or JBP region has the same requirements as a Java program. For example, it must commit resources before reading subsequent messages or exiting the application. This COBOL GU call does not implicitly commit resources when the program is running in an JMP or JBP region as it does when the program is running in an MPP region. Regarding message processing and transaction synchronization, a CHKP call in an IMS Java dependent region does not automatically retrieve a message from the message queue.

10.6 JDBC and SQL enhancement

These enhancements are supported as UQ74475 for IMS Version 8. As additional information, we introduce the future enhancement of SQL syntax.

10.6.1 JDBC enhancement

IMS Version 9 supports Java Database Connectivity (JDBC) including the ability to obtain scroll insensitive result sets. This is the new types of result sets provided as java.sql.ResultSet interface's static constants, ResultSet.TYPE_SCROLL_INSENSITIVE and ResultSet.TYPE_FORWARD_ONLY.

TYPE_SCROLL_INSENSITIVE indicates the type for a ResultSet object that is scrollable but generally not sensitive to changes made by others. This result set contains data which is not sensitive to underlying changes in the database while the result set is open. Result sets of this type may also be traversed in reverse order, by absolute position, or by relative position.

TYPE_FORWARD_ONLY indicates the type for a ResultSet object whose cursor may move only forward. This is the default result set type unless another type is explicitly requested or an SQL query is processed that contains aggregates, ORDER BY, or GROUP BY clause.

Refer to *IMS Version 9: IMS Java Guide and Reference,* SC18-7821 for more detail about supported JDBC interfaces.

10.6.2 SQL enhancement

Structured Query Language (SQL) enhancements for new SQL keywords and aggregate functions are supported in IMS Version 9, which are SUM, AVG, MIN, MAX, COUNT, ORDER BY, and GROUP BY.

For the SUM and AVG functions, arguments must be numbers and the data type of the result is the same as the data type of the argument. There are some exception, that the result is a long if the argument type is a byte, short, or an integer. Or that the result is a double-precision floating point if the argument type is a single-precision floating point.

For the MIN, MAX functions, arguments values can be any built-in type except BIT, BLOB, and BINARY. The data type of the result is the same as the data type of the argument. For the COUNT function, the argument can be any field name in a segment.

The DISTINCT keyword is not supported as an argument to any of the functions.

The field names specified in a GROUP BY or ORDER BY clause must exactly match the field name specified in the SELECT list.

The AS clause is used to rename any field in a SELECT list. If an AS clause is used, the new field name must be used to reference the field. In the event that an aggregate function is used without an AS clause, the resulting column name is the function name, followed by an underscore, followed by the field name. For example, MAX(age) will result in a column name of MAX_age. The name MAX_age is then to be used for all subsequent references to the field like ResultSet resultSet.getInt("MAX_age").

10.6.3 SQL syntax enhancement

IBM intends to enhance SQL syntax pertain to the usage of fields in an SQL WHERE clause. This is going to be the enhancement after the general availability of Version 9.

At this time, only searchable fields, which mean that they are defined in an IMS DBD, are allowed in a WHERE clause. However, there are situations where a searchable field is further broken down into subcomponents via COBOL copybooks, depending your applications.

For example, the field 'Address" can be broken down into 'city', 'state' and 'zip code'. This case is supported by IMS Java, however, we do not support using the fields city, state, or zip code in the WHERE clause itself. We are adding support to allow for this in near future. We call these fields (‘city’, ‘state’, and ‘zip code’) Non-searchable fields. Non- searchable fields will be able to be used in a WHERE clause with the following rules adhered:

- All subfields of a searchable field must be provided
- All subfields must be provided consecutively, although they can be in any order
- All subfields must be ANDed with one another
  - No sense to say:

    ```
    WHERE City = X OR State = Y
    ```

  - These fields are going to be concatenated together to form the actual searchable field
- The ‘=’ operator is the only supported operator when a subfield is compared to a value
  - No sense to say:

    ```
    WHERE City > someValue AND State < another Value
    ```

  - These fields are going to be concatenated together to form the actual searchable field

With this support, you can simply concatenate the fields together themselves to form the key field. For example, to continue with the notion of an ‘Address’ field broken down into ‘City’, ‘State’, and ‘Zip’, it seems it would be quite easy to just combine the three into one String object and use it in the SQL query:

```
SELECT *
FROM MyPCB.MySegment
WHERE Address = ‘Los Gatos California 95030’
```

as opposed to:

```
SELECT *
FROM MyPCB.MySegment
WHERE City = ‘Los Gatos’ AND State = ‘California’ AND Zip = ‘95030’
```

Imagine if the City field was a double-precision (double in Java), State was a single-precision (float in Java), and Zip was an integer, as we have seen that the subfields are not necessarily of the same type. It is extremely difficult to manually convert all of those data types, especially since Java doubles and floats use the IEEE 754 standard and z/OS or OS/390 does not. It makes the first query quite difficult. However, with this support you can simply use the second query above and IMS Java library will handle all of the type conversion on behalf of the applications.

10.7 Remote database services

IMS Version 9 provides IMS Java remote database services in which you can develop and deploy applications that run on WebSphere Application Server on non-z/OS platform and access IMS databases remotely through EJB on WebSphere Application Server for z/OS.

Through several IMS versions, you have been able to write applications that run on WebSphere Application Server for z/OS and access IMS databases. See Figure 10-3 on page 170.

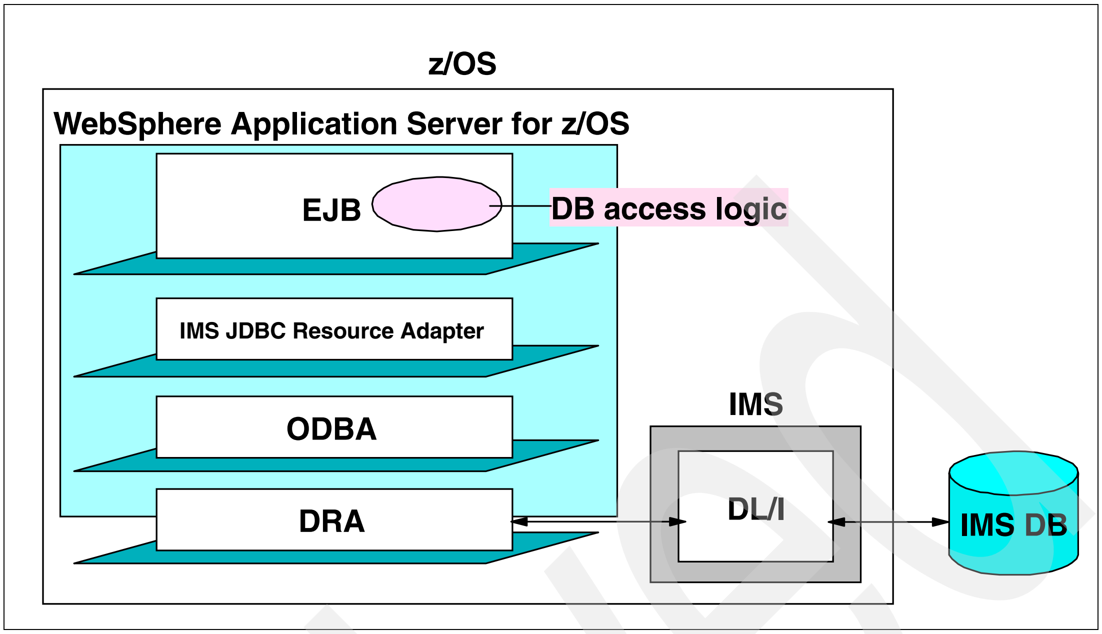
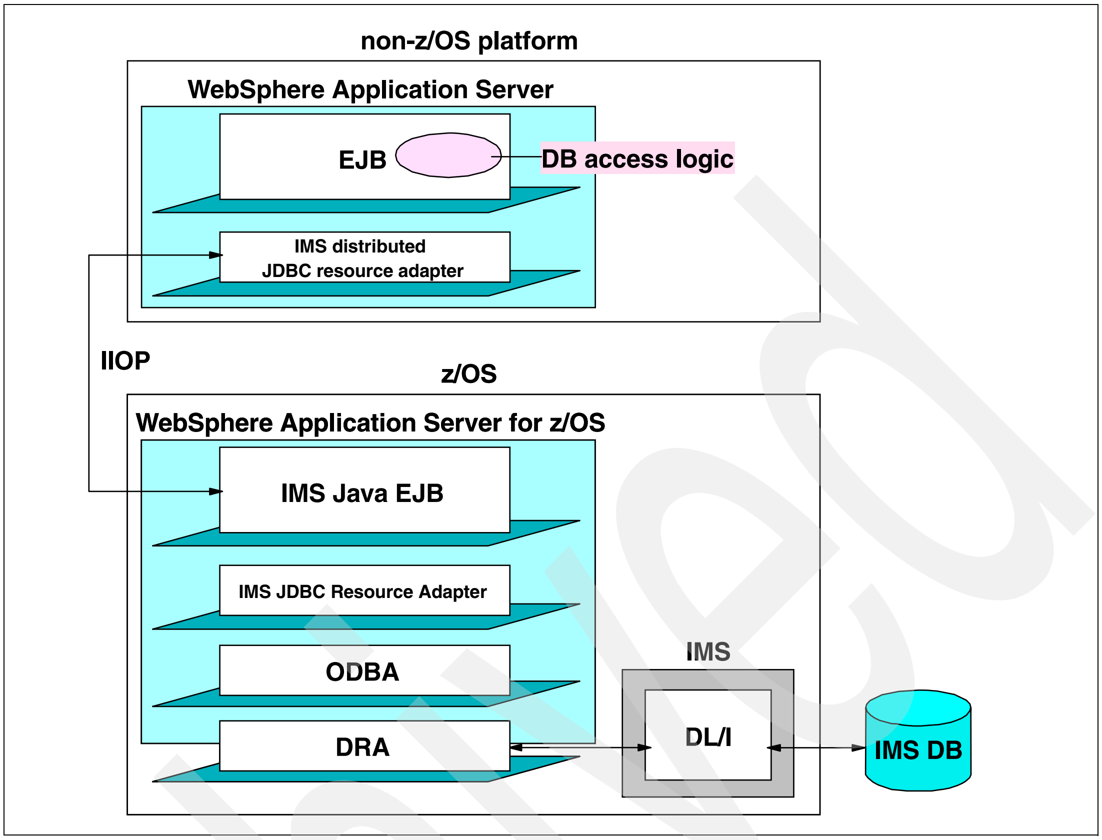

z/OS
WebSphere Application Server for z/OS
EJB
DB access logic
IMS JDBC Resource Adapter
ODBA
DRA
IMS
DL/I
IMS DB

Figure 10-3 WebSphere Application Server for z/OS EJB using IMS Java

This shows an EJB on WebSphere Application Server for z/OS is accessing IMS data. JDBC or IMS Java hierarchical interface calls (which are also classes and methods provided by IMS Java, whose formats are similar to DL/I calls), are passed to the IMS Java layer, which converts the calls to DL/I calls. The IMS Java layer passes these calls to IMS Open DataBase Access (ODBA), which uses the database resource adapter (DRA) to access the DL/I region in IMS. Note you still need to develop a z/OS application to access IMS data, that is, DB access logic is still on z/OS.

For more detail about ODBA and DRA, refer to manuals *IMS Version 9: Application Programming: Database Manager,* SC18-7809 and *IMS Version 9: Application Programming: Design Guide,* SC18-7810.

With IMS Java remote database services provided by IMS Version 9, you can develop a non-z/OS application to access IMS data, that is, IMS database access logic is on non-z/OS platform. See Figure 10-4 on page 171.



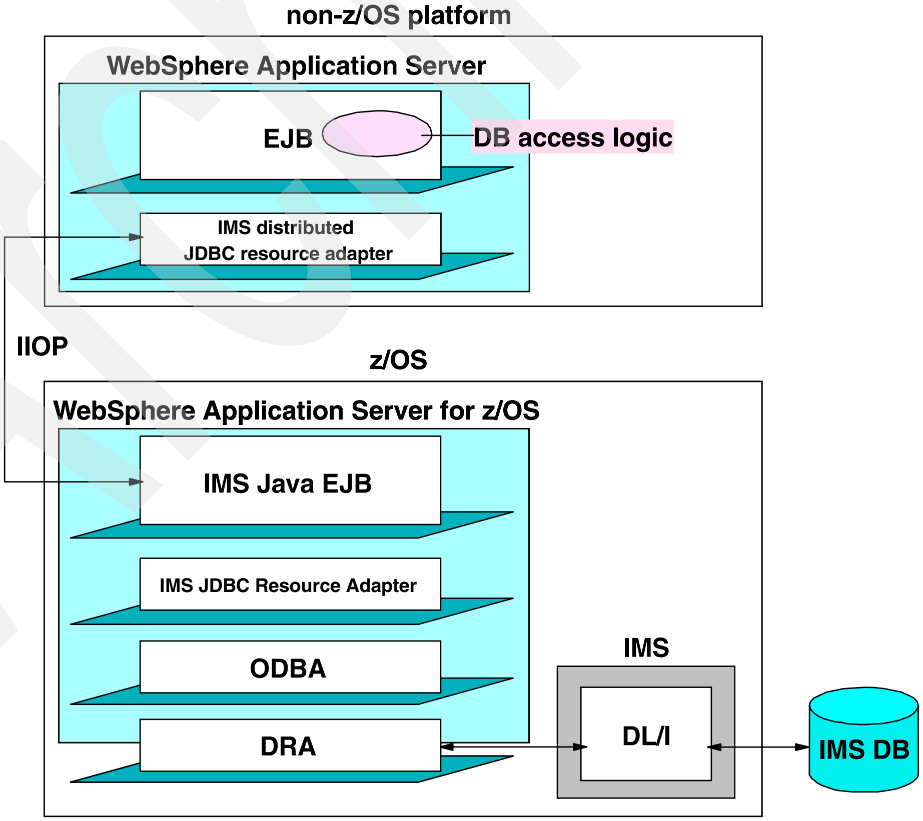


Figure 10-4 IMS Java remote database services and WebSphere Application Server components

The EJB, that contains your business logic, and is deployed on WebSphere Application Server on non-z/OS platform, passes database access requests to IMS distributed JDBC resource adapter provided by IMS Version 9 as imsjavaRDS.rar file. This resource adapter contains a type 3 JDBC driver. Then Internet Inter-ORB Protocol (IIOP) is used between WebSphere Application Server for z/OS and WebSphere Application Server running on another platform. IIOP allows the servers to exchange data, which is securely transferred across the internet using the SSL (Secure Sockets Layer). IMS Java-supplied EJB on z/OS side receives the data through IIOP, then passes transaction information to the IMS JDBC resource adapter that is deployed on the z/OS platform. The IMS JDBC resource adapter passes data to ODBA, which uses the DRA.

For system requirements, installing, configuring the servers, and running sample applications, refer to *IMS Version 9: IMS Java Guide and Reference,* SC18-7821.

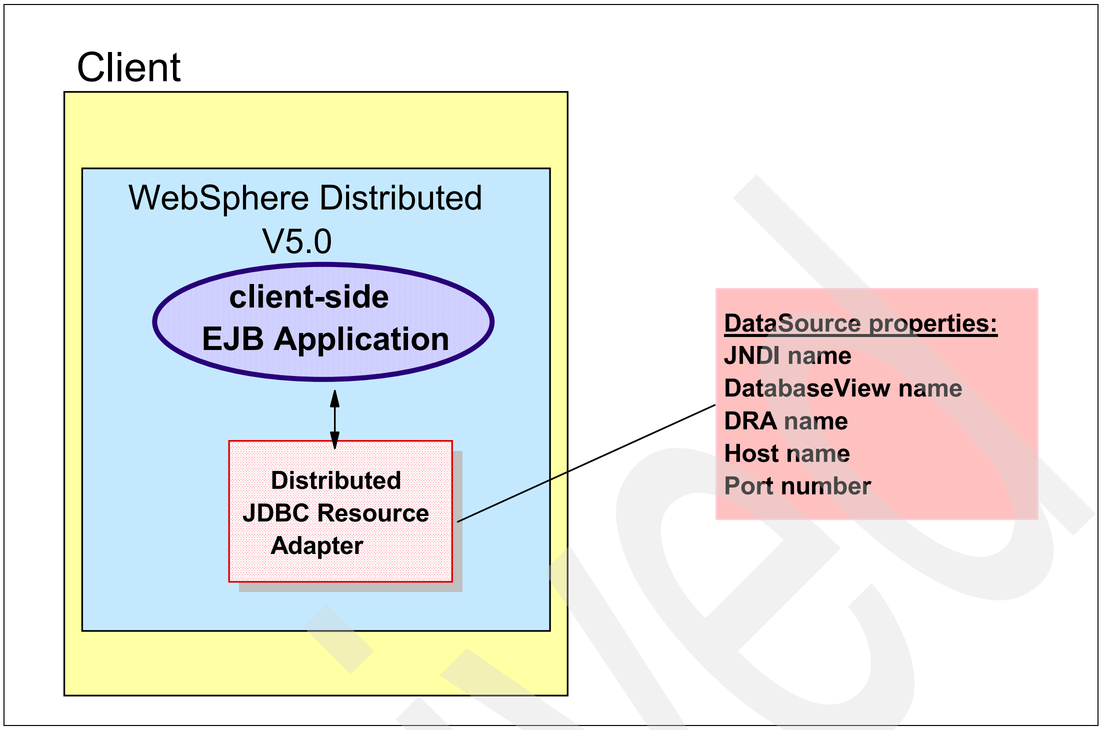
10.7.1 Remote database services components

IMS Java remote database services is a set of J2EE components that provide remote access to IMS data through IMS ODBA.The client-side and server-side components support retrieval, update, delete, and insert activity to the IMS databases. These requests are sent (transparently to the application) across the network and processed in IMS. This support provides an architected solution that allows EJB applications deployed on Distributed WebSphere Application Server to issue JDBC calls to access IMS Databases.

Client-side component

For the client-side of the connection, IMS Java RDS provides an IMS distributed JDBC resource adapter. To condition the distributed WebSphere Application Server V5.0 server for JDBC access to IMS, the IMS JDBC resource adapter must first be installed. This component contains a type 3 JDBC driver which interprets the JDBC request to access and manipulate the IMS data. A type 3 JDBC driver is defined as a standard net-protocol fully Java technology-enabled driver that translates JDBC API calls into a DBMS-independent net protocol which is then translated to a DBMS protocol by a server.

After it is installed, a J2C Connection Factory instance can be deployed. This is a DataSource object deployed in JNDI (Java Naming and Directory Interface) that can be used to obtain a JDBC 2.0 connection. The DataSource object defines properties that pertain to the actual target data source for the connection. See Figure 10-5 on page 173.


Client
WebSphere Distributed V5.0
client-side EJB Application
Distributed JDBC Resource Adapter
DataSource properties:
JNDI name
DatabaseView name
DRA name
Host name
Port number

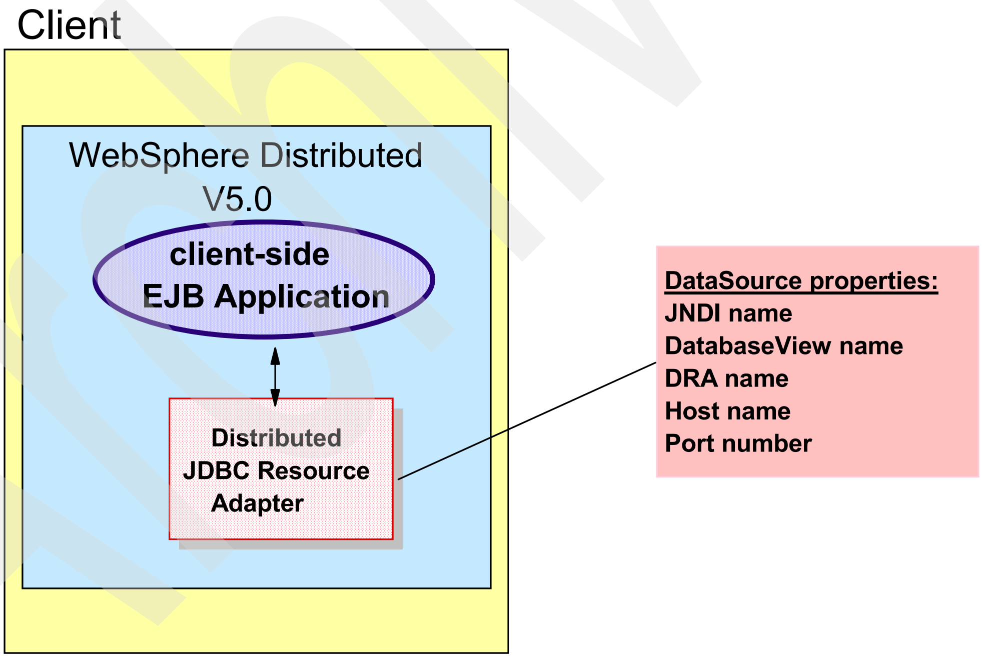

Client
WebSphere Distributed V5.0
client-side EJB Application
Distributed JDBC Resource Adapter
DataSource properties:
JNDI name
DatabaseView name
DRA name
Host name
Port number

Figure 10-5 Remote database services: client-side components

The Datasource properties consist of:

- Java Naming and Directory Interface (JNDI) name

  JNDI name in J2C Connection Factories

- DatabaseView name

  Fully qualified name of the DLIDatabaseView subclass

- DRA table name

Target IMS system

- Host name

  Name of the target host or IP Address

- Port number

  Internet Inter-ORB Protocol (IIOP) port number of the target host WebSphere Application Server z/OS

Once the connection is obtained, then a customer-written application running as an EJB in that distributed server can use the JDBC 2.0 API to access IMS DB data. To the application EJB, JDBC access appears local, that is, the application does not have to be written with any knowledge that network traffic is involved.

Server-side component

For the server-side (IMS side) of the connection, a server EJB is delivered as part of the IMS Java RDS support. This EJB must reside on a WebSphere Application Server for z/OS V5.0 environment because it uses the IMS JDBC resource adapter. The reason for the WebSphere Application Server for z/OS requirement for the IMS JDBC resource adapter is that this environment provides a mechanism to invoke a custom service when the J2EE server is brought up and down. The existing IMS JDBC resource adapter (originally introduced to provide JDBC access from WebSphere Application Server for z/OS - PQ57320 for IMS Version 7) uses the custom service. When the server is brought up, the IMS adapter initializes the ODBA environment and correspondingly terminates it when the server is brought down. As such, once the server is brought up, every application running in the server can use an already initialized ODBA environment. In ODBA environment, and for WebSphere Application Server for z/OS, Resource Recovery Services (RRS) is used, so the RRS component requires the use of the `RRS=Y` in your IMS startup parameters. See Figure 10-6.

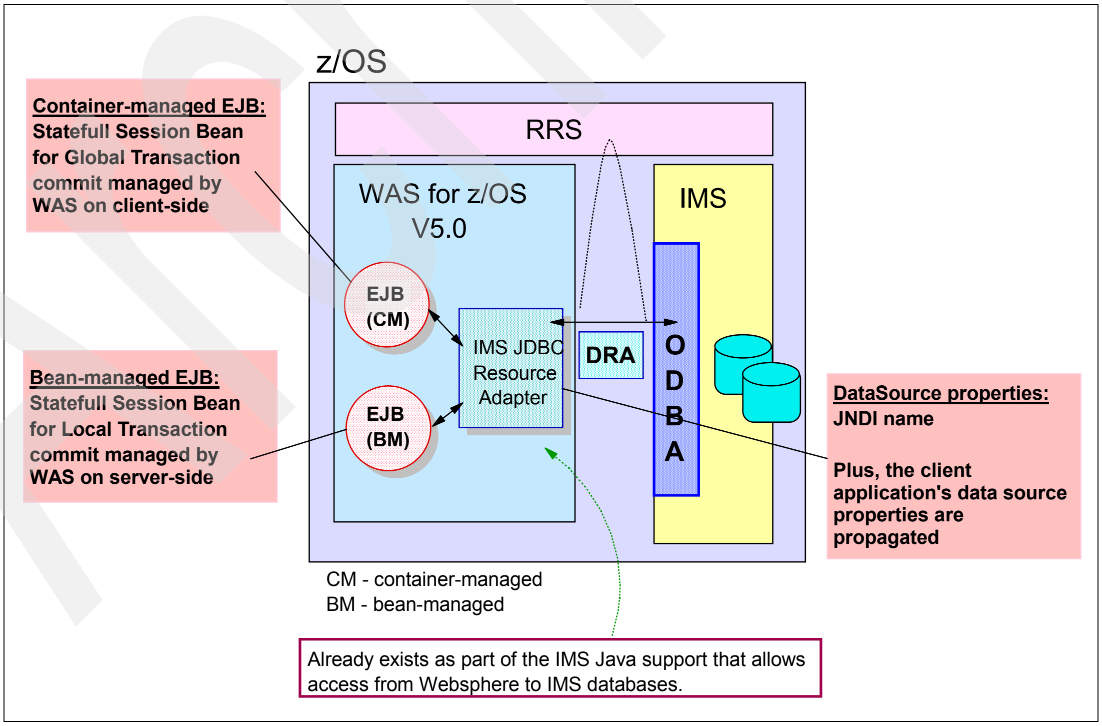


Figure 10-6 Remote database services: server-side components

The server EJB that is provided is a stateful session bean that allows it to act as a server-side extension of the client's IMS distributed JDBC resource adapter. The primary responsibility of the server EJB is to maintain the state of a client application with respect to IMS and forward all database requests to the IMS JDBC resource adapter. The server EJB is actually invoked as one of two different types: a container-managed EJB (CM EJB) and a bean-managed EJB (BM EJB).

When the server EJB runs as a container-managed EJB, it supports a client EJB application that has requested global transaction semantics. This means that the client application operates under one transaction context, or unit of work for all its database activity. In this case the distributed J2EE application server has the responsibility for being the coordinator of the global transaction.

On the other hand, when the server EJB runs a a bean-managed EJB, it supports a client EJB application that is using local transaction semantics. Each database connection operates under its own transaction context. When a commit is issued against work done on one connection (database), this does not affect work on another connection. In this scenario, WebSphere Application Server for z/OS becomes the coordinator of the transaction. This is because ODBA requires a transaction context to be present. What really happens is that the BM EJB starts a global transaction context on the server side to communicate with IMS while maintaining the local transaction aspect with respect to the client application's connection.

Client-server interaction

A diagram of the client-server interaction components is shown in Figure 10-7.

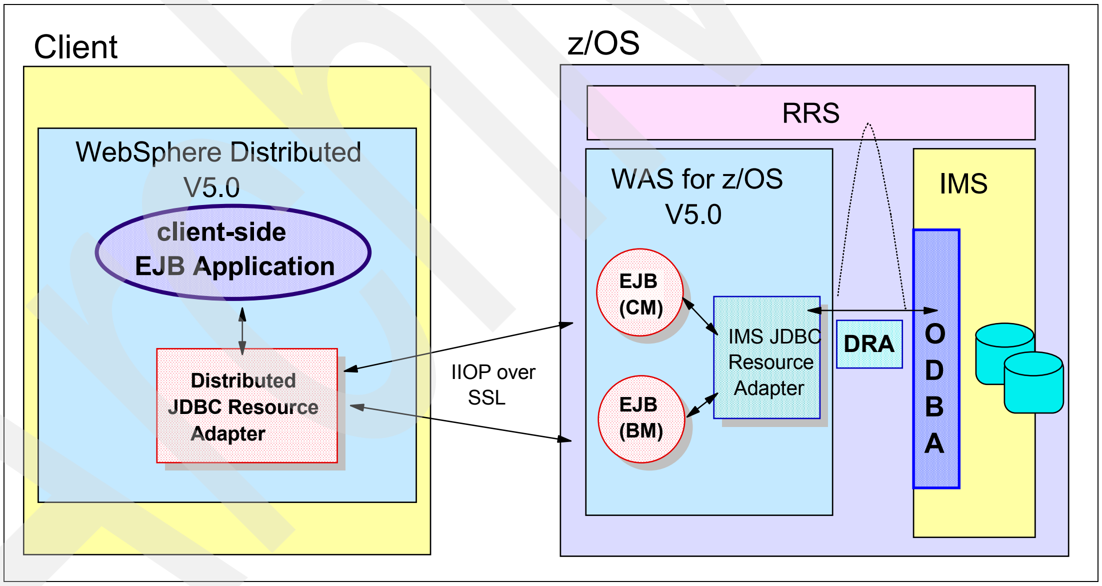

Client
WebSphere Distributed V5.0
client-side EJB Application
Distributed JDBC Resource Adapter
IIOP over SSL
z/OS
RRS
WAS for z/OS V5.0
EJB (CM)
EJB (BM)
IMS JDBC Resource Adapter
DRA
IMS
ODBA

Figure 10-7 Client-Server interaction

This figure shows the flow of a request from a client-side EJB application issuing a JDBC call all the way to where it is serviced in IMS. The IMS distributed JDBC resource adapter hides the remote access interaction from the client-side EJB, takes the request and establishes communication with the server EJB on the z/OS platform. All activity is kept within the distributed resource adapter until a JDBC connection statement is executed. At this point, an IIOP connection is established with the server-side EJB. IIOP is a transport mechanism standard that supports seamless interoperation between J2EE distributed objects.

The IMS distributed JDBC resource adapter chooses an EJB type (CM or BM) based on whether or not a transaction context exists. A transaction context exists and a CM EJB will be created if the client-side EJB application had previously issued a getTransaction() method requesting the use of global transaction semantics. Otherwise, local transaction semantics are used and a BM EJB will be created. Information from the client-side DataSource (DRA name, DLIDatabase View subclass) is also propagated to the EJB. Once the instance of the server-side EJB (CM or BM) is created, a connection to ODBA is requested. This results in the loading of the metadata classes and the allocation of the PSB. Note that the database metadata classes, that are produced by the DLIModel Utility provided through several IMS versions, are stored on the mainframe for use by the IMS JDBC resource adapter. They do not need to be moved to the distributed platforms. Once the connection is established, an indication of success or failure is sent back to the client.

The IMS distributed JDBC resource adapter, on a successful connection indication, creates the Statement object. Once this object is handed to the client EJB, a JDBC query can be executed.

10.7.2 Security

There are several resources that need to be properly secured in an IMS Java remote database services environment.

Access to client-side EJB

In a J2EE environment, the security context (or the identity) on the execution thread is determined by a deployment property of the application component known as "run-as". The run-as value can be one of the following:

- System Identity which specifies that the application component runs as the identity of the server region id.
- Caller Identity which specifies that the application component runs as the identity of its caller.
- Role which specifies that the application component runs as a particular identity that is defined by the administrator in the J2EE server.

Note that run-as values of Caller Identity and Role are not supported in this implementation. For information about run-as options and other security issues, see the WebSphere Application Server information center.

Network security

Secure Socket Layer (SSL) is used to protect the communication between WebSphere Application Server on distributed platforms and WebSphere Application Server on z/OS. The client application which uses the IMS distributed JDBC resource adapter must be deployed with a run-as identity of System. This is currently a requirement for interoperation between a distributed application server and WebSphere Application Server for z/OS. As such, the client application should be protected with restricted access so only authorized users can access it on the distributed platform. Communication between distributed WebSphere Application Server and WebSphere Application Server for z/OS. No support for user ID and password that is provided from either the application or the authentication alias should be protected with restricted access.

There is no support for passing the caller identity of the client application. At this point, the use of SSL and IIOP requires the identity to be switched to "system identity" when communicating between distributed and z/OS application servers.

Security on WebSphere Application Server for z/OS

Security regarding WebSphere Application Server for z/OS can be separated into two topics, server-side EJB and between IMS.

Server-side EJB

The distributed application server propagates the run-as property of the client-side EJB, and, WebSphere Application Server for z/OS places an appropriate security identity on the thread that will be used to access IMS. The server-side EJB defaults to the "run-as" identity value of system, which is the server region ID of WebSphere Application Server for z/OS. You can change the "run-as" property in the deployment descriptor of the server-side EJB before installing it.

Between IMS

The ODBA environment requires a previously verified access control environment element (ACEE), which WebSphere Application Server for z/OS places on the execution thread. ACEE is a control block which is built when a call to RACF or equivalent security product is issued. In the WebSphere Application Server for z/OS environment, the IMS JDBC resource adapter uses Sync-to-Thread processing to ensure that a security context is place on the thread during execution to access an IMS database. This places an ACEE on the execution thread, based on the "run-as" property of the server-side EJB.

10.7.3 Installation steps

There are actions both on the client and the server-side for setting up the IMS Java remote database services.

Client-side steps

On the client side, the imsjavaRDS.rar file that is the IMS distributed JDBC resource adapter is to be installed. After it is installed and before it can be used, a J2C Connection Factory (DataSource) instance needs to be deployed. This is a DataSource object deployed in JNDI that will be used to obtain a JDBC 2.0 connection. It must contain the values for: DRA table name, DLIDataBaseView subclass (that is, metadata classes) name, IIOP port number, and server-side host name.

Additionally, client-side applications can access multiple IMS databases simultaneously by deploying multiple J2C Connection Factory instances or DataSources, each with a unique set of properties specific to an IMS database. Note that it is not actually necessary to deploy multiple DataSource objects to access multiple databases. An application can configure a DataSource instance for access to a particular database after it has been looked up in the JNDI namespace, as opposed to fully configuring the DataSource prior to deployment into JNDI.

Server-side steps

On the server-side, the imsjava91.rar file that is the IMS JDBC resource adapter is to be installed. This needs to be installed in WebSphere Application Server for z/OS V5.0 along with configuring the custom service that allows the ODBA environment to be initialized or terminated when the server comes up or down. The RAR file contains the following:

- ra.xml, which is the heart of the rar file. It includes things like which Java classes (located in the jar file) which are to be used for the Connection Factory. The return value will be of type DataSource, and indicates to the installation tool which values must be set, for example, DRA Name, DatabaseView.

- The IMSJdbcCustomService.xml file, which enables and points to a class in the Jar file with custom services, in this case the calling of CIMS INIT and CIMS TALL call. CIMS INIT initializes ODBA thread, and CIMS TALL terminates it.
- The howto.html file, which is a verbose description including steps on how to deploy the rar file.

Then the EAR file named imsjavaRDS.ear that contains the two IMS Java-provided EJBs is to be installed. These two EJBs are stateful session beans and act as server-side extensions of the IMS Distributed JDBC resource adapter.

For more details of installing steps, settings, and running sample applications, refer to *IMS Version 9: IMS Java Guide and Reference,* SC18-7821.

11

Knowledge Based Log Analysis (KBLA)

This chapter discusses the new IMS Version 9 Knowledge Based Log Analysis (KBLA) functions which are supplied as a standard part of the IMS product. KBLA has an ISPF panel driven interface which may be used to invoke existing log and trace utilities as well as several new programs which provide more interpretative and user friendly information. KBLA may significantly reduce the need to reference the utilities manuals and avoid JCL or control statement errors. It should also reduce or eliminate the need to cross reference with control block DSECTS.

This chapter contains the following:

- Invoking KBLA
- A description of the KBLA tasks
- KBLA setup
- Supported IMS log utilities
- New IMS log and trace utilities
- Examples of using the new functions

11.1 Overview of KBLA

Knowledge Based Log Analysis is actually a set of tools and utilities to assist with finding and interpreting information about the IMS log or IMS trace data sets. The tedious and error prone process of coding JCL and control cards is replaced with a set of ISPF panels and help text for invoking both the pre existing log utilities and the new KBLA functions. These functions make the task of analyzing log and trace data much easier. Several options are available for formatting the data depending on the level of detail desired.

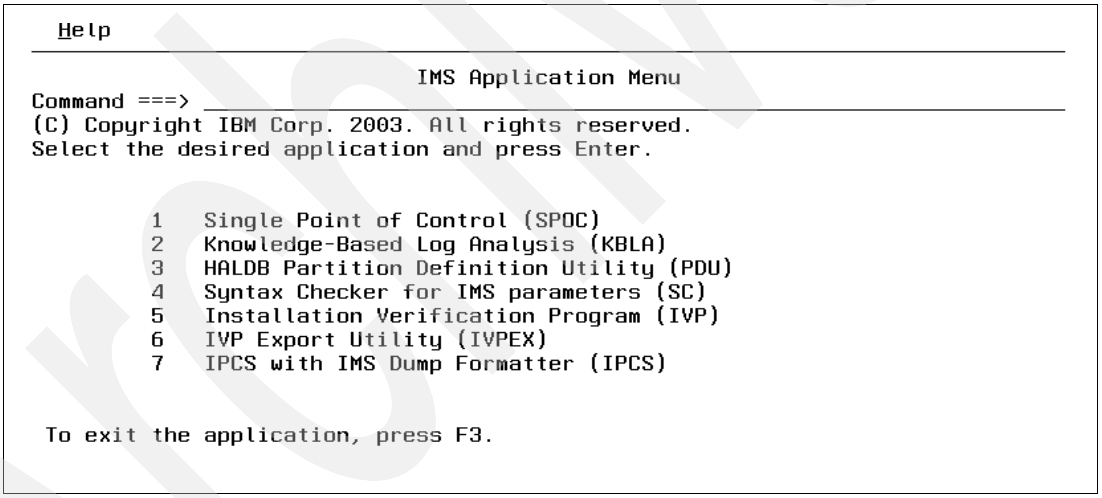
While KBLA is a valuable tool and part of the IMS Version 9 product it is not considered to be a replacement for the IMS Problem Investigator product which is a separately priced tool. IMSPI provides some similar function but is more interactive in nature.

11.2 Invoking KBLA

KBLA may be invoked by the following TSO command where *hlq* is the high level qualifier for the IMS execution libraries:

```
EX 'hlq.SDFSEXEC(DFSKBSRT)' 'HLQ(hlq)'
```

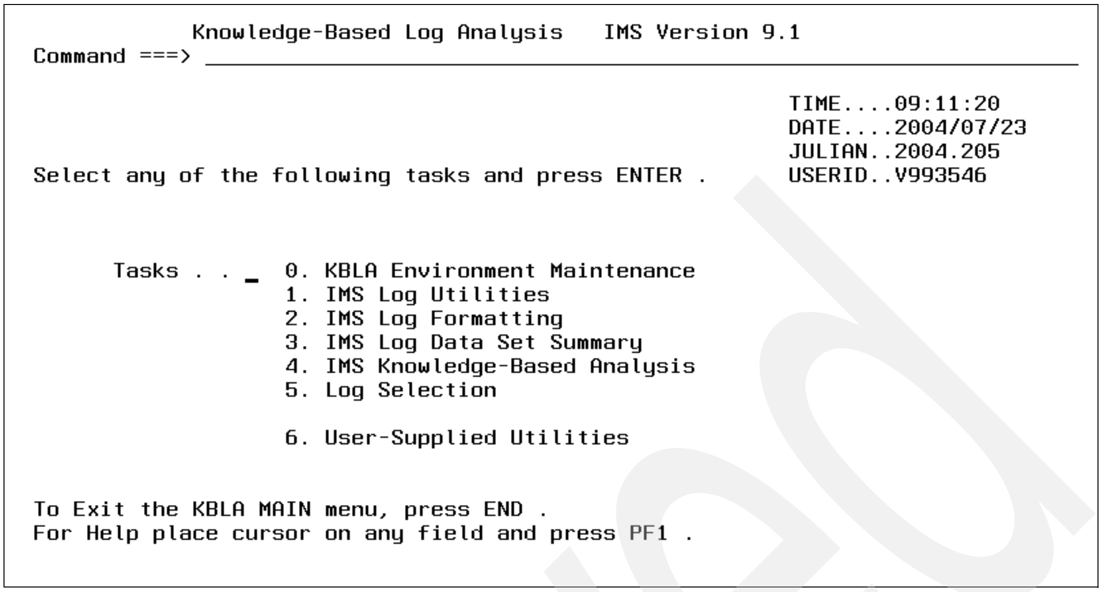
The above command is also imbedded within the IMS Application menu and may be invoked by selecting option 2 from the menu shown in Figure 11-1.



```
  Help
 ------------------------------------------------------------------------------
                             IMS Application Menu
 Command ===> __________________________________________________________________
 (C) Copyright IBM Corp. 2003. All rights reserved.
 Select the desired application and press Enter.


         1   Single Point of Control (SPOC)
         2   Knowledge-Based Log Analysis (KBLA)
         3   HALDB Partition Definition Utility (PDU)
         4   Syntax Checker for IMS parameters (SC)
         5   Installation Verification Program (IVP)
         6   IVP Export Utility (IVPEX)
         7   IPCS with IMS Dump Formatter (IPCS)


  To exit the application, press F3.
```

Figure 11-1 IMS Application Menu

Selecting option 2 from the IMS Application Menu will take you to the primary KBLA menu shown in Figure 11-2 on page 181.



```
             Knowledge-Based Log Analysis   IMS Version 9.1
Command ===> ___________________________________________________________________

                                                        TIME....09:11:20
                                                        DATE....2004/07/23
                                                        JULIAN..2004.205
Select any of the following tasks and press ENTER .     USERID..V993546



      Tasks . . _  0. KBLA Environment Maintenance
                   1. IMS Log Utilities
                   2. IMS Log Formatting
                   3. IMS Log Data Set Summary
                   4. IMS Knowledge-Based Analysis
                   5. Log Selection

                   6. User-Supplied Utilities


To Exit the KBLA MAIN menu, press END .
For Help place cursor on any field and press PF1 .
```

Figure 11-2 KBLA primary menu

The following is a list of the Task selections available and a short description of each selection. You may press PF1 on any of the KBLA panels for a more complete description of the task or individual fields.

- 0. KBLA Environment Maintenance

  You will need to go here first to perform some setup functions including required data set names.

- 1. IMS Log Utilities

  Here you will find an interface to various standard IMS log utilities including a new user data scrub utility.

- 2. IMS Log Formatting

  This selection will allow you to extract, search, and format log records using specific criteria.

- 3. IMS Log Data Set Summary

  This option will read the log and provide summary information about the records, timestamps, and so on.

- 4. Knowledge-Based Analysis

  This selection is for a set of tools to process log or trace data and provide usable information for debugging or performance analysis.

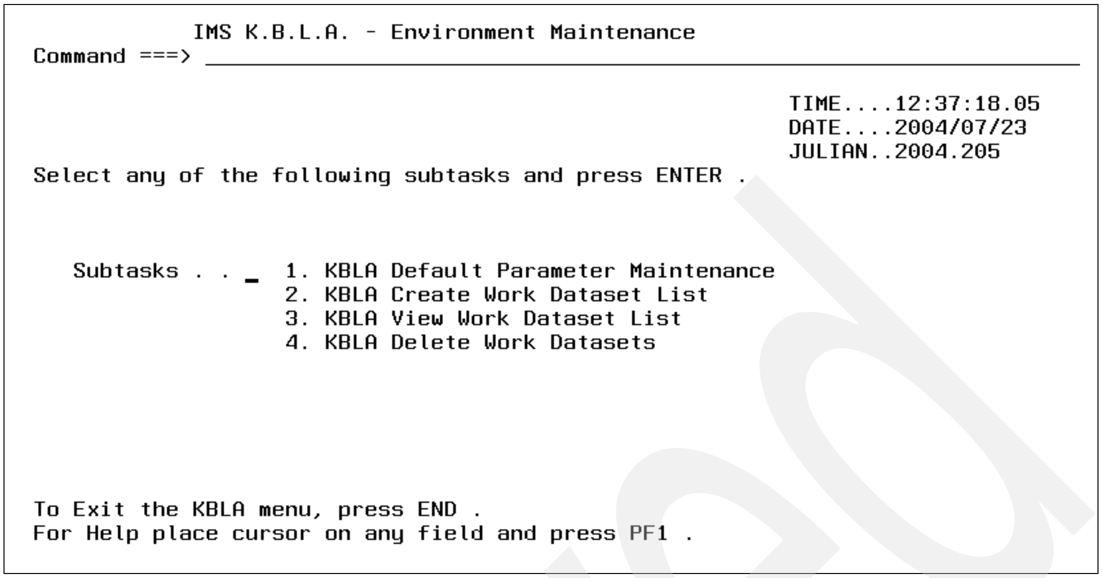
- 5. Log Selection

  This function allows you to specify which log data sets to process.

- 6. User Supplied Utilities

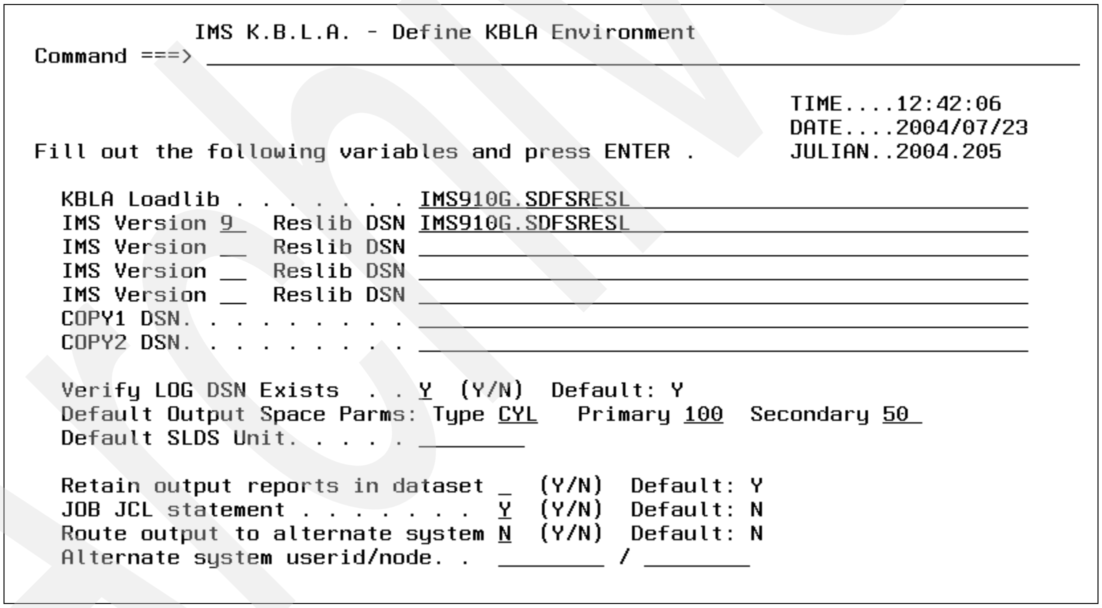
  This selection allows you to add your own functions to be initiated from KBLA.

The first time you invoke KBLA you should select Task 0 to perform some setup functions. Selecting task 0 will take you to the KBLA maintenance panel shown in Figure 11-3 on page 182. From there select option 1 to setup the default values to be used for various KBLA functions and fill in the values similar to those shown in Figure 11-4 on page 182.



```
            IMS K.B.L.A. - Environment Maintenance
Command ===> ___________________________________________________________________

                                                    TIME....12:37:18.05
                                                    DATE....2004/07/23
                                                    JULIAN..2004.205
Select any of the following subtasks and press ENTER .


   Subtasks . . _  1. KBLA Default Parameter Maintenance
                   2. KBLA Create Work Dataset List
                   3. KBLA View Work Dataset List
                   4. KBLA Delete Work Datasets






To Exit the KBLA menu, press END .
For Help place cursor on any field and press PF1 .
```

Figure 11-3 KBLA Environment Maintenance panel



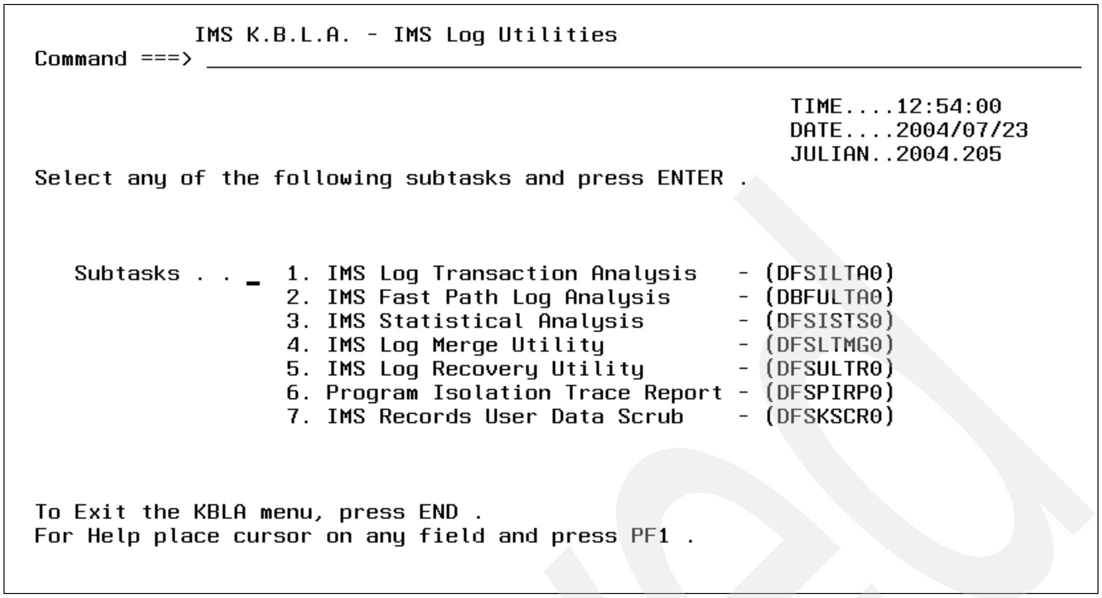
```
            IMS K.B.L.A. - Define KBLA Environment
Command ===> ___________________________________________________________________

                                                    TIME....12:42:06
                                                    DATE....2004/07/23
Fill out the following variables and press ENTER .  JULIAN..2004.205

  KBLA Loadlib . . . . . . . IMS910G.SDFSRESL______________________________
  IMS Version 9_  Reslib DSN IMS910G.SDFSRESL______________________________
  IMS Version __  Reslib DSN ______________________________________________
  IMS Version __  Reslib DSN ______________________________________________
  IMS Version __  Reslib DSN ______________________________________________
  COPY1 DSN. . . . . . . . . ______________________________________________
  COPY2 DSN. . . . . . . . . ______________________________________________

  Verify LOG DSN Exists  . . Y  (Y/N)  Default: Y
  Default Output Space Parms: Type CYL   Primary 100  Secondary 50_
  Default SLDS Unit. . . . . ________

  Retain output reports in dataset _  (Y/N)  Default: Y
  JOB JCL statement . . . . . . .  Y  (Y/N)  Default: N
  Route output to alternate system N  (Y/N)  Default: N
  Alternate system userid/node. .  ________ / ________
```

Figure 11-4 Define KBLA Environment panel

Pressing the enter key with Y selected for the JOB JCL statement will show you the generated JOB statement which you may then edit as appropriate for your installation. After editing and saving your JOB statement you can PF3 back to primary KBLA menu.

11.2.1 KBLA task selection 1 - IMS Log Utilities

If you then select option 1 from the primary KBLA menu previously shown in Figure 11-2 on page 181 you will see the list of utilities shown in Figure 11-5 on page 183. The first six utilities have been in IMS for many years and selecting any of them will take you to another panel where you can enter information which will be filled into the various JCL parameters or control statements as appropriate. Help is available for any of the fields by pressing PF1.

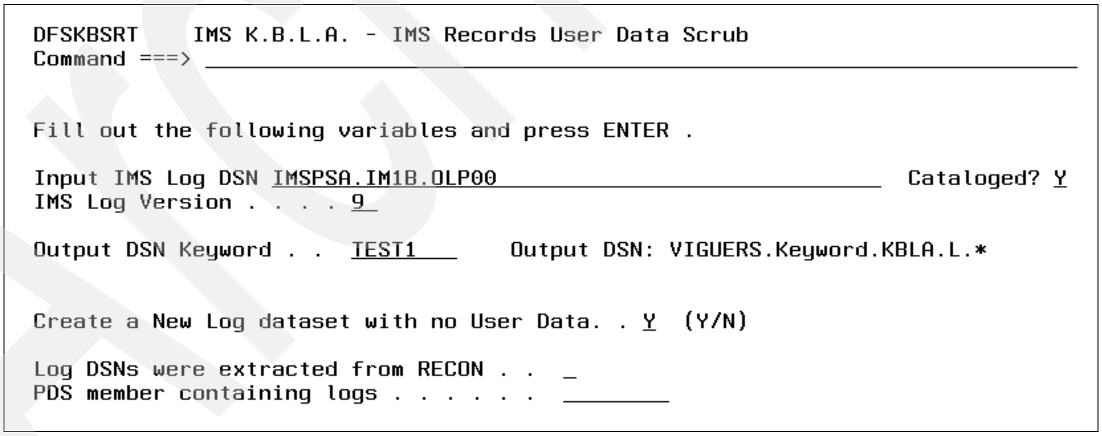


```
             IMS K.B.L.A. - IMS Log Utilities
Command ===> ____________________________________________________________________

                                                           TIME....12:54:00
                                                           DATE....2004/07/23
                                                           JULIAN..2004.205
Select any of the following subtasks and press ENTER .


   Subtasks . . _  1. IMS Log Transaction Analysis   - (DFSILTA0)
                   2. IMS Fast Path Log Analysis     - (DBFULTA0)
                   3. IMS Statistical Analysis       - (DFSISTS0)
                   4. IMS Log Merge Utility          - (DFSLTMG0)
                   5. IMS Log Recovery Utility       - (DFSULTR0)
                   6. Program Isolation Trace Report - (DFSPIRP0)
                   7. IMS Records User Data Scrub    - (DFSKSCR0)


To Exit the KBLA menu, press END .
For Help place cursor on any field and press PF1 .
```

Figure 11-5 IMS Log Utilities panel

KBLA IMS Records User Data Scrub

There is one new utility which may be invoked from the IMS Log Utilities panel for IMS Version 9 which is the data scrubbing utility. IMS Records User Data Scrub is a routine that scans all the IMS log and deletes those record parts that may contain sensitive or customer business transaction information without compromising the integrity nor the content of the vital IMS system data. Selecting Subtask 7 will take you to the panel shown in Figure 11-6. Once you have filled in the Input log data set name and the other fields you press enter and it should take you to an edit panel with the JCL to run the job.



```
DFSKBSRT     IMS K.B.L.A. - IMS Records User Data Scrub
Command ===> ____________________________________________________________________


Fill out the following variables and press ENTER .

Input IMS Log DSN IMSPSA.IM1B.OLP00____________________________  Cataloged? Y
IMS Log Version . . . . 9_

Output DSN Keyword . .  TEST1___    Output DSN: VIGUERS.Keyword.KBLA.L.*


Create a New Log dataset with no User Data. . Y  (Y/N)

Log DSNs were extracted from RECON . .  _
PDS member containing logs . . . . . .  ________
```

Figure 11-6 IMS Records User Data Scrub panel

In the previous example we selected Y to create an new log data set with the user data removed from specific records. The output of the job will show the records which have had the user data removed. An sample of the summary output is shown in Example 11-1.

Example 11-1 User Data Scrub summary output

```
CONTROL  CNTL  STOPAFT=EOF
*************************************************
* IMS-V9 INPUT LOG DATA SET NAME(S):            *
* IMSPSA.IM1B.SAVEOLDS
**************************************************
*                                                *
* RECORDS USER DATA SCRUB                        *
*                                                *
**************************************************
OPTION   COPY E=DFSKSCRO
*
DFS707I END OF FILE ON INPUT
#01   RECORDS PROCESSED:          384
#03   RECORDS PROCESSED:        2,929
#50   RECORDS PROCESSED:       12,120
#5901 RECORDS PROCESSED:        1,967
#5903 RECORDS PROCESSED:        1,967
#5950 RECORDS PROCESSED:       26,917
#67   RECORDS PROCESSED:            0
DFS708I OPTION COMPLETE
DFS703I END OF JOB
```

KBLA option 2.3 was used with the basic formatting option to print a sample log record showing the user data before running the scrub utility in Example 11-2and after running the KBLA user data scrub utility in Example 11-3.

Example 11-2   Input message record before running scrub

```
01 RECORD    INPUT MESSAGE QUEUED          DATE/TIME: 2004-07-28 02:12:17.903302 GMT
MSG PREFIX HEADER
 00000000 000000   018C0000 01C18194  04000053 04000053   01669001 C9D4F1C2  40404040 BB94E081   *.....AAM............IM1B    .M
 00000020 000020   38A8F2A1 C9D4F1C2  40404040 BB94E081   38A8F2A1 80000100  00000000 00000000   *.Y2.IM1B    .M.A.Y2..........

SYSTEM PREFIX
 00000000 000000   00408100 88000000  E3D7F0F0 F0F1F9F5   00260000 00000000  00000000 00000001   *. A.H...TP000195.............
 00000020 000020   E6C8F0F0 F2F0F0F5  E3D7C3C3 C4404040   00000000 00000000  E3D7C3C3 C4D6F240   *WH002005TPCCD   ........TPCCDO

EXTENDED PREFIX HEADER
 00000000 000000   00108600 00E67400  00000000 00000000                                          *..F..W.........

SECURITY PREFIX
 00000000 000000   00168800 E6C8F0F0  F2F0F0F5 40404040   40404040 E400                          *..H.WH002005       U.

WORK LOAD MANAGER PREFIX
 00000000 000000   00188900 0C918004  BB94E081 38ABF981   00000000 00000000                      *..I..J...M.A..9A........

SYSTEM EXTENSION PREFIX
 00000000 000000   00188A00 2004210F  02121790 3294016D   00008000 00000000                      *............M._........

TRANSACTION MANAGER PREFIX
 00000000 000000   00908C00 00000000  00000000 00000000   E3D7C3C3 C4404040  E6C8F0F0 F2F0F0F5   *................TPCCD   WH0020
 00000020 000020   00000148 40000000  00000000 00000000   00000000 00000000  00000000 00000000   *.... ........................
 00000040 000040   00000000 09100000  00000000 00000000   00000000 000001E3  D7C3C3C4 40404040   *.......................TPCCD
 00000060 000060   40404040 40400000  00000000 00000000   00000910 00000000  00000000 00000000   *      ........................
 00000080 000080   00000000 00000000  00000000 00000000                                          *................

USER DATA
 00000000 000000   00160301 E3D7C3C3  C4404040 E6C8F0F0   F2F04040 F0F3                          *....TPCCD   WH0020  03

RECORD SUFFIX
 00000000 000000   BB94E081 38AC6801  00000000 001FB780                                          *.M.A...........
```

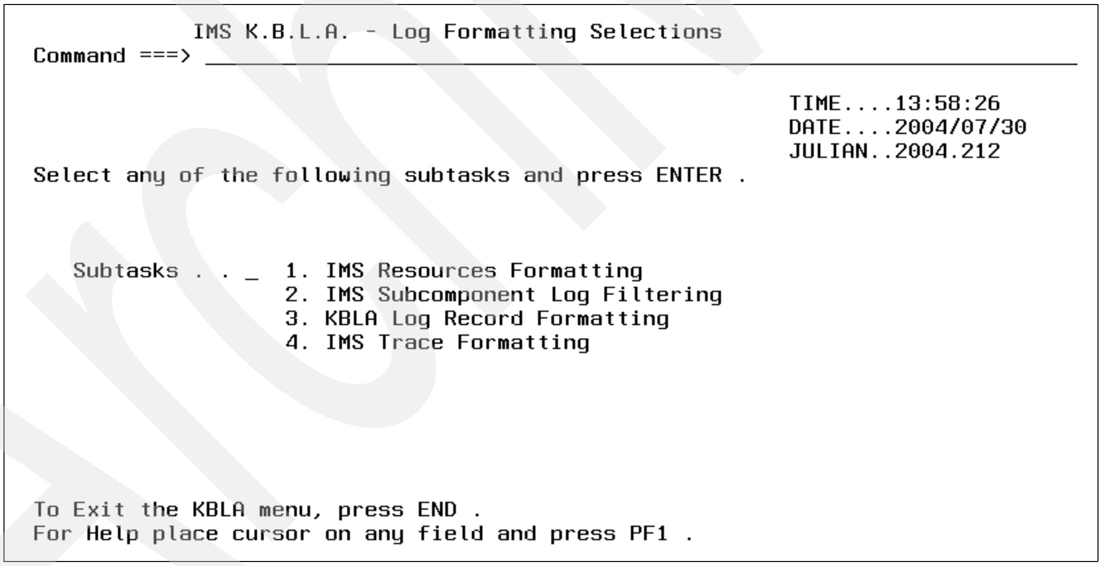
Example 11-3   Input message record after running scrub

```
01 RECORD    INPUT MESSAGE QUEUED          DATE/TIME: 2004-07-28 02:12:17.903302 GMT
MSG PREFIX HEADER
 00000000 000000   018C0000 01C18194  04000053 04000053   01669001 C9D4F1C2  40404040 BB94E081   *.....AAM............IM1B    .M
 00000020 000020   38A8F2A1 C9D4F1C2  40404040 BB94E081   38A8F2A1 80000100  00000000 00000000   *.Y2.IM1B    .M.A.Y2..........
```

```
SYSTEM PREFIX
 00000000 000000   00408100 88000000  E3D7F0F0 F0F1F9F5   00260000 00000000  00000000 00000001   *. A.H...TP000195.............
 00000020 000020   E6C8F0F0 F2F0F0F5  E3D7C3C3 C4404040   00000000 00000000  E3D7C3C3 C4D6F240   *WH002005TPCCD   ........TPCCDO

EXTENDED PREFIX HEADER
 00000000 000000   00108600 00E67400  00000000 00000000                                          *..F..W.........

SECURITY PREFIX
 00000000 000000   00168800 E6C8F0F0  F2F0F0F5 40404040   40404040 E400                          *..H.WH002005       U.

WORK LOAD MANAGER PREFIX
 00000000 000000   00188900 0C918004  BB94E081 38ABF981   00000000 00000000                      *..I..J...M.A..9A........

SYSTEM EXTENSION PREFIX
 00000000 000000   00188A00 2004210F  02121790 3294016D   00008000 00000000                      *............M._........

TRANSACTION MANAGER PREFIX
 00000000 000000   00908C00 00000000  00000000 00000000   E3D7C3C3 C4404040  E6C8F0F0 F2F0F0F5   *...............TPCCD   WH0020
 00000020 000020   00000148 40000000  00000000 00000000   00000000 00000000  00000000 00000000   *.... ........................
 00000040 000040   00000000 09100000  00000000 00000000   00000000 000001E3  D7C3C3C4 40404040   *.......................TPCCD
 00000060 000060   40404040 40400000  00000000 00000000   00000910 00000000  00000000 00000000   *     .......................
 00000080 000080   00000000 00000000  00000000 00000000                                          *................

USER DATA
 00000000 000000   60C3D3C5 C1D9C5C4  606E0000 00000000   00000000 0000                          *-CLEARED->...........

RECORD SUFFIX
 00000000 000000   BB94E081 38AC6801  00000000 001FB780                                          *.M.A...........
```

11.2.2 KBLA task selection 2 - IMS Log Formatting

Selecting task 2 from the KBLA primary menu will take you to the panel shown in Figure 11-7.



```
                IMS K.B.L.A. - Log Formatting Selections
Command ===> ________________________________________________________________

                                                         TIME....13:58:26
                                                         DATE....2004/07/30
                                                         JULIAN..2004.212
Select any of the following subtasks and press ENTER .


   Subtasks . . _  1. IMS Resources Formatting
                   2. IMS Subcomponent Log Filtering
                   3. KBLA Log Record Formatting
                   4. IMS Trace Formatting







To Exit the KBLA menu, press END .
For Help place cursor on any field and press PF1 .
```

Figure 11-7   Log Formatting Selections

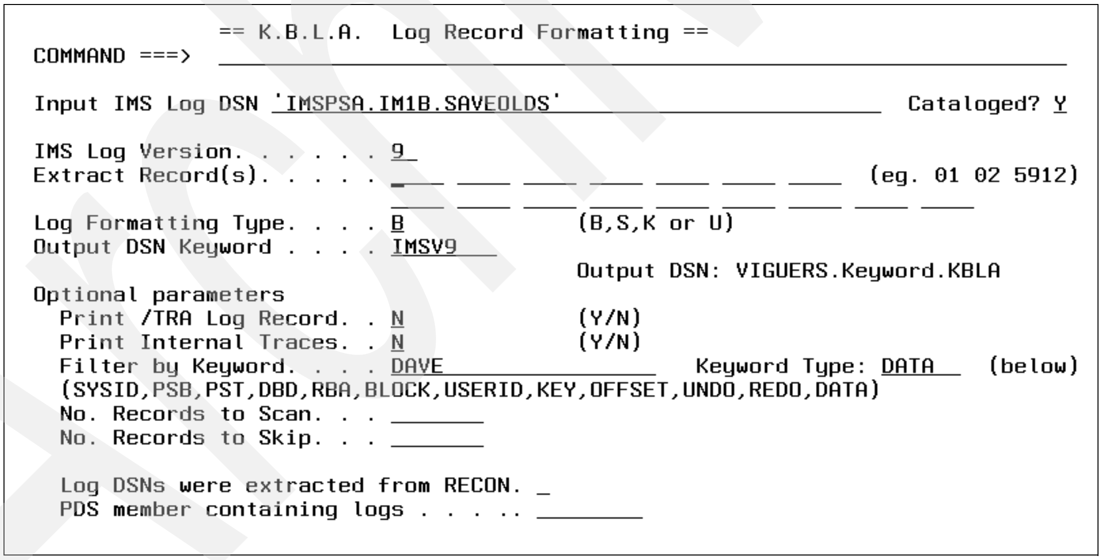
The following functions may be selected from the Log Formatting Menu shown above.

- 1. IMS Resources Formatting - this option allows you to extract information from the log such as input/output messages, transactions scheduled, security violations, and so on based on a specific resource name. There is no need to know the log record type since KBLA formatting routines will determine those automatically.
- 2. IMS Subcomponent Log Filtering - allows you to select log records related to a particular component of IMS such as DB, TM, FP, and so on. You may select any or all of the subcomponents and also filter by a specific resource name if desired.

- 3. KBLA Log Records Formatting - provides many options for selection and formatting of the log records. A sample of this selection screen is shown in Figure 11-8.
- 4. IMS Traces Formatting - allows you to select specific trace entries to be formatted using the standard DFSERA30 and DFSERA60 exits but also provides enhanced formatting for NODE trace entries.
- 5. Program Isolation Trace Report - generates the JCL and control cards to invoke DFSPIRP0 to create the PI trace report.

The subtask selections are well documented with help panels and therefore we will not go into much further detail in this book. However, there is a common field on the first 3 selections which is the Log Formatting type. An example of this field is shown in Figure 11-8. As you can see this field may be B, S, K, or U. The meaning of these options are as follows:

- B - basic formatting similar to DFSERA30 for most records but enhanced formatting where the log record is actually broken into multiple sections such as with the 01 input message and the 03 output message record. An example of this is shown in Example 11-4.
- S - summary formatting shows only the log record type followed by a short description and then the log sequence number and the date/time. This is shown in Example 11-5 on page 187.
- K - this option provides a much more interpretative description of the various log records and makes it easy to identify the important field contents. An example of the K formatting type is shown in Example 11-6 on page 187.
- U - the unformatted selection is used to copy all or selected records from the input log file to a separate data set. There is no formatting or printing from this option.



```
              == K.B.L.A.  Log Record Formatting ==
COMMAND ===> ________________________________________________________________

Input IMS Log DSN 'IMSPSA.IM1B.SAVEOLDS'                         Cataloged? Y

IMS Log Version. . . . . . 9_
Extract Record(s). . . . . ____ ____ ____ ____ ____ ____ ____  (eg. 01 02 5912)
                           ____ ____ ____ ____ ____ ____ ____ ____ ____
Log Formatting Type. . . . B           (B,S,K or U)
Output DSN Keyword . . . . IMSV9___
                                       Output DSN: VIGUERS.Keyword.KBLA
Optional parameters
  Print /TRA Log Record. . N           (Y/N)
  Print Internal Traces. . N           (Y/N)
  Filter by Keyword. . . . DAVE________________   Keyword Type: DATA__   (below)
  (SYSID,PSB,PST,DBD,RBA,BLOCK,USERID,KEY,OFFSET,UNDO,REDO,DATA)
  No. Records to Scan. . . _______
  No. Records to Skip. . . _______

  Log DSNs were extracted from RECON. _
  PDS member containing logs . . . .. ________
```

Figure 11-8 KBLA Log Record Formatting menu

Example 11-4 Basic log record formatting

```
37 RECORD    SYNCPOINT PROCESSOR LOG RECORD DATE/TIME: 2004-07-28 02:12:17.392038 GMT
 00000000 000000   00680000 37380000  00000000 1CBE8118   C9D4F1C2 40404040  00000070 00000734   *..............A.IM1B    ......
 00000020 000020   BB94E080 BBD39505  FFFFCA5B 00000000   00000000 00000000  00000000 00000000   *.M...LN....$..................
 00000040 000040   00000000 00000000  00000000 400000C0   00010000 00000005  BB94E080 BBDA6D85   *........... ............M....
 00000060 000060   00000000 001FB6B2                                                             *........
59 RECORD    FP SUCCESSFUL SYNCPOINT RECORD DATE/TIME: 2004-07-28 02:12:17.392039 GMT
 00000000 000000   00C40000 59370100  00010000 E3D7C3C3   D7404040 2004210F  02121739 2036016D   *.D..........TPCCP   ..........
 00000020 000020   4000003A 00010000  00000000 00000004   00010001 00000000  00000000 C9D4F1C2   * ............................IM
 00000040 000040   40404040 00000070  00000734 00000000   00000001 00000002  000A0002 00000000   *    ........................
 00000060 000060   00000002 00000000  00010000 00020000   00000000 00000000  00000000 0000BB94   *..............................
 00000080 000080   E080BBDA 58450000  00050000 C9D4F1C2   40404040 BB94E080  A86EE986 C9D4F1C2   *............IM1B    .M..Y>ZFIM
 000000A0 0000A0   40404040 BB94E080  AD54AA21 00000002   00010000 BB94E080  BBDA7605 00000000   *    .M..............M........
 000000C0 0000C0   001FB6B3                                                                     *....
42 RECORD    OLDS SWITCH/CHKPT WAS TAKEN    DATE/TIME: 2004-07-28 02:12:17.392048 GMT
 00000000 000000   02580000 42003800  00000800 00000224   00000000 C9D9F0C2  C9D4F1C2 40404040   *....................IROBIM1B
 00000020 000020   000801C0 00922000  00000230 00000000   00000000 0000023C  00000000 00000000   *.....K........................
 00000040 000040   001D5F10 40404040  40404040 40404040   40404040 C9D4F1C2  40404040 C9D4F1C2   *..¬.                IM1B    IM
03 RECORD    OUTPUT MESSAGE QUEUED          DATE/TIME: 2004-07-28 02:12:17.392077 GMT
MSG PREFIX HEADER
 00000000 000000   01B70000 03C38210  04000051 04000051   01661801 C9D4F1C2  40404040 BB94E080   *.....CB.............IM1B    .M
 00000020 000020   BBC5CE60 C9D4F1C2  40404040 BB94E080   BBC5CE60 80000500  00000000 00000000   *.E.-IM1B    .M...E.-..........

SYSTEM PREFIX
 00000000 000000   00408100 88000000  00000000 00000000   00000000 00000000  00000000 00000001   *. A.H.........................
 00000020 000020   FFFFFFFF FFFFFFFF  E2D4C1E2 E3C5D940   00000000 00000000  40404040 40404040   *........SMASTER ........

EXTENDED PREFIX HEADER
 00000000 000000   00108600 00E67400  00000000 00000000                                         *..F..W.........

SECURITY PREFIX
 00000000 000000   00168800 C4C6E2D4  E3C3D5E3 40404040   40404040 D300                         *..H.DFSMTCNT       L.

WORK LOAD MANAGER PREFIX
 00000000 000000   00188900 00000000  BB94E080 BBC77240   00000000 00000000                     *..I......M...G. ........

SYSTEM EXTENSION PREFIX
 00000000 000000   00188A00 2004210F  02121739 1734016D   00000000 00000000                     *..............._........

TRANSACTION MANAGER PREFIX
 00000000 000000   00908C00 00000000  00000000 00000000   E2D4C1E2 E3C5D940  C4C6E2D4 E3C3D5E3   *................SMASTER DFSMTC
 00000020 000020   00000108 00000000  00000000 00000000   00000000 00000000  00000000 00000000   *..............................
 00000040 000040   00000000 09100000  00000000 00000000   00000000 000005E2  D4C1E2E3 C5D94040   *.......................SMASTER
 00000060 000060   40404040 40400000  00000000 00000000   00000910 00000000  00000000 00000000   *      ........................
 00000080 000080   00000000 00000000  00000000 00000000                                         *................

USER DATA
 00000000 000000   00410B00 C4C6E2F3  F2F5F7C9 40D6D5D3   C9D5C540 D3D6C740  D5D6E640 E2E6C9E3   *....DFS3257I ONLINE LOG NOW SW
 00000020 000020   C3C8C5C4 406040C6  D9D6D440 C4C6E2D6   D3D7F0F0 40E3D640  C4C6E2D6 D3D7F0F3   *CHED - FROM DFSOLP00 TO DFSOLP
 00000040 000040   15                                                                           *.

RECORD SUFFIX
 00000000 000000   BB94E080 BBDCD5A5  00000000 001FB6B5                                         *.M....NV........
```

Example 11-5 Summary log record formatting

```
OPTION   PRINT E=DFSKBLAS
43 RECORD  STATUS OF CURRENT OLDS D/S                 LSN: 001FB6AB     DATE/TIME: 2004-07-28   02:12:17.391904 UTC
59 RECORD  FP INPUT MESSAGE RECEIVED                  LSN: 001FB6AC     DATE/TIME: 2004-07-28   02:12:17.391905 UTC
59 RECORD  FP OUTPUT MESSAGE RECEIVED                 LSN: 001FB6AD     DATE/TIME: 2004-07-28   02:12:17.391908 UTC
59 RECORD  FP DEDB UPDATED                            LSN: 001FB6AE     DATE/TIME: 2004-07-28   02:12:17.391912 UTC
59 RECORD  FP DEDB UPDATED                            LSN: 001FB6AF     DATE/TIME: 2004-07-28   02:12:17.391914 UTC
59 RECORD  FP DEDB UPDATED                            LSN: 001FB6B0     DATE/TIME: 2004-07-28   02:12:17.391917 UTC
59 RECORD  FP DEDB UPDATED                            LSN: 001FB6B1     DATE/TIME: 2004-07-28   02:12:17.391920 UTC
37 RECORD  SYNCPOINT PROCESSOR LOG RECORD             LSN: 001FB6B2     DATE/TIME: 2004-07-28   02:12:17.392038 UTC
59 RECORD  FP SUCCESSFUL SYNCPOINT RECORD             LSN: 001FB6B3     DATE/TIME: 2004-07-28   02:12:17.392039 UTC
42 RECORD  OLDS SWITCH/CHKPT WAS TAKEN                LSN: 001FB6B4     DATE/TIME: 2004-07-28   02:12:17.392048 UTC
03 RECORD  OUTPUT MESSAGE QUEUED                      LSN: 001FB6B5     DATE/TIME: 2004-07-28   02:12:17.392077 UTC
35 RECORD  MSG WAS ENQUEUED/RE-ENQUEUED               LSN: 001FB6B6     DATE/TIME: 2004-07-28   02:12:17.392958 UTC
03 RECORD  OUTPUT MESSAGE QUEUED                      LSN: 001FB6B7     DATE/TIME: 2004-07-28   02:12:17.393000 UTC
35 RECORD  MSG WAS ENQUEUED/RE-ENQUEUED               LSN: 001FB6B8     DATE/TIME: 2004-07-28   02:12:17.393006 UTC
31 RECORD  GET UNIQUE (GU) ISSUED FOR MSG             LSN: 001FB6B9     DATE/TIME: 2004-07-28   02:12:17.393332 UTC
59 RECORD  FP INPUT MESSAGE RECEIVED                  LSN: 001FB6BA     DATE/TIME: 2004-07-28   02:12:17.395421 UTC
59 RECORD  FP OUTPUT MESSAGE RECEIVED                 LSN: 001FB6BB     DATE/TIME: 2004-07-28   02:12:17.395423 UTC
59 RECORD  FP DEDB UPDATED                            LSN: 001FB6BC     DATE/TIME: 2004-07-28   02:12:17.395425 UTC
59 RECORD  FP DEDB UPDATED                            LSN: 001FB6BD     DATE/TIME: 2004-07-28   02:12:17.395426 UTC
59 RECORD  FP DEDB UPDATED                            LSN: 001FB6BE     DATE/TIME: 2004-07-28   02:12:17.395428 UTC
59 RECORD  FP DEDB UPDATED                            LSN: 001FB6BF     DATE/TIME: 2004-07-28   02:12:17.395429 UTC
```

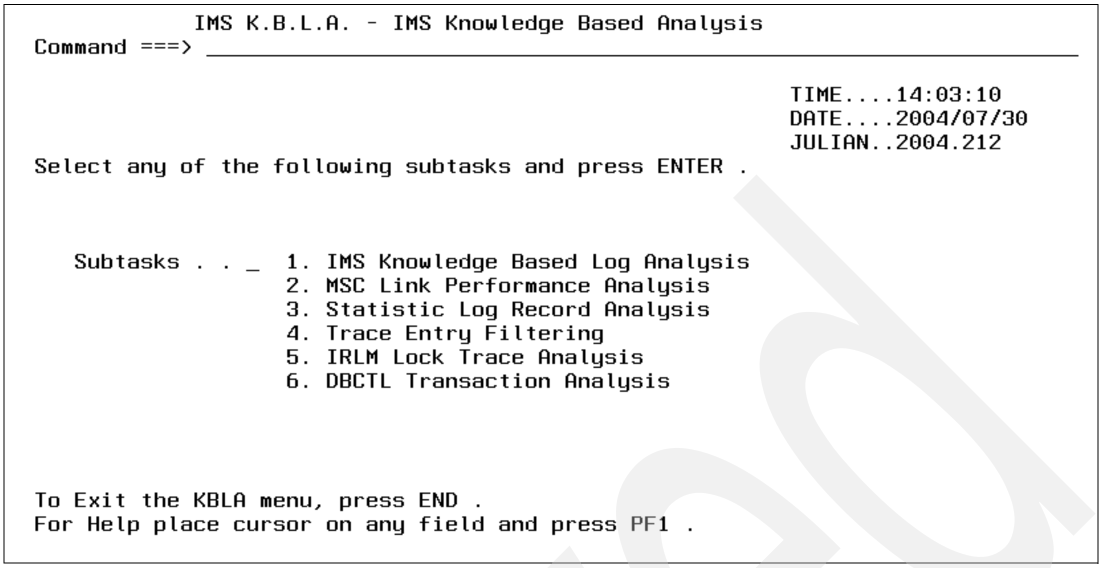
Example 11-6 KBLA interpretative formatting

```
5950 RECORD - FP DEDB UPDATED
     PROG  NAME: PSBPA          AREA NAME: AREAWH01      RBA FIRST CI: 00001000
         VSO/SVSO: ON        DMHR BUFF #: 0011       DB CHANGE OFFSET: 006B
     RCVY TOKN: C9D4F1C24040404040000000007000000734           DBD NAME: WAREDB
     DATE/TIME: 2004-07-28 02:12:17.391920 UTC                LOG SEQ NO: 001FB6B1
```

```
37 RECORD - IMS SYNCPOINT TRANSFER
     DESTINATION IS: SMB           PST #: 0001             PCB ADDR: 1CBE8118
     NO INPUT/OUTPUT         NO APPC/OTMA MSG
     RCVY TOKEN: C9D4F1C240404040000000007000000734
     DATE/TIME: 2004-07-28 02:12:17.392038 UTC             LOG SEQ NO: 001FB6B2

5937 RECORD - FP SUCCESSFUL SYNCPNT RECORD
     BATCH APPLICATION
     RCVY TOKN: C9D4F1C240404040000000007000000734
     UOW: C9D4F1C240404040BB94E080A86EE986C9D4F1C240404040BB94E080AD54AA210000
     DATE/TIME: 2004-07-28 02:12:17.392039 UTC             LOG SEQ NO: 001FB6B3

42 RECORD    OLDS SWITCH/CHKPT WAS TAKEN
     RECORDED BY: ONLINE SYSTEM IM1B    (ID=0008)        IRLM NAME/ID: IR0BIM1B/01C0
     PRILOG 1ST LSN: 00000000001D5F10    PRILOG START TIME: 2004210F01473170
     LOG TIMESTAMP:  2004210F01462576    LOG SWITCH TIME:   2004210F02121733
                     *** CHECKPOINT INFORMATION SECTION ***
     CURR. CHKPT ID: 2004210/01560987    CHKPT ZERO ID:     2004208/08052741
     OLDEST LCRE ID: 2004210/01473201    COLDBASE/COLDCOMM ID: 0000000/00000000
     OLDEST BMP ID:  2004210/01560987    FP CHKPT ID:       2004210/01560987
     SNAPQ CHKPT ID: 0000000/00000000    DC CHKPT ID:       2004210/01560987
     DUMPQ CHKPT ID: 2004208/08052741
     DATE/TIME: 2004-07-28 02:12:17.392048 UTC             LOG SEQ NO: 001FB6B4

03 RECORD    OUTPUT MESSAGE QUEUED
     NODE NAME: N/A          UNSOLICITED MSG          DEST LTERM/TRAN: SMASTER
    ORIGIN IMS: IM1B       PROCESSING IMS: IM1B         MSG TIMESTAMP: 2004210 02121662
     1ST/CURRENT DRRN: 04000051/04000051
     UOW: C9D4F1C240404040BB94E080BBC5CE60C9D4F1C240404040BB94E080BBC5CE608000
                             ** SQ MESSAGE - PUT PROCESS **D
     DATE/TIME: 2004-07-28 02:12:17.392077 UTC             LOG SEQ NO: 001FB6B5
```

11.2.3 KBLA task selection 4 - IMS Knowledge Based Analysis

Selecting task 4 from the primary KBLA menu will take you to the menu shown in Figure 11-9.



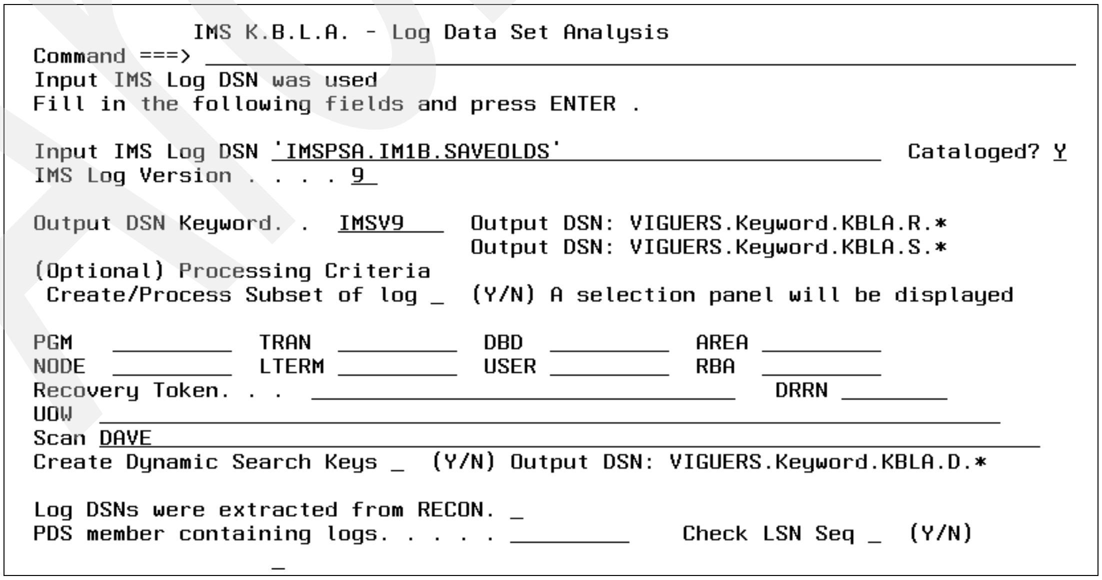
```
              IMS K.B.L.A. - IMS Knowledge Based Analysis
Command ===> __________________________________________________________________

                                                      TIME....14:03:10
                                                      DATE....2004/07/30
                                                      JULIAN..2004.212
Select any of the following subtasks and press ENTER .



   Subtasks . . _  1. IMS Knowledge Based Log Analysis
                   2. MSC Link Performance Analysis
                   3. Statistic Log Record Analysis
                   4. Trace Entry Filtering
                   5. IRLM Lock Trace Analysis
                   6. DBCTL Transaction Analysis




To Exit the KBLA menu, press END .
For Help place cursor on any field and press PF1 .
```

Figure 11-9 Knowledge Based Analysis menu

11.2.4 IMS KBLA - Log Data Set Analysis

Subtask 1 of KBLA allows you search the log based on specific criteria and also optionally create an output file which is a subset of the input log with only those records matching the processing criteria. The selection panel is shown in Figure 11-10. A very interesting function on this panel is the Create Dynamic Search Keys option. This function will use the selection criteria specified and in addition it will search for related records.

An example would be if you chose to process based on a transaction name and specified the dynamic keys options. In this case a log record would be selected even though the transaction name may not be carried in a particular record but, because the UOW was the same as a prior record, it would be included. The Processing Criteria and Dynamic Search Key option provide some very powerful tools in processing the log.



```
              IMS K.B.L.A. - Log Data Set Analysis
Command ===> __________________________________________________________________
Input IMS Log DSN was used
Fill in the following fields and press ENTER .

Input IMS Log DSN 'IMSPSA.IM1B.SAVEOLDS'_______________________  Cataloged? Y
IMS Log Version . . . . 9_

Output DSN Keyword. .  IMSV9___   Output DSN: VIGUERS.Keyword.KBLA.R.*
                                  Output DSN: VIGUERS.Keyword.KBLA.S.*
(Optional) Processing Criteria
 Create/Process Subset of log _  (Y/N) A selection panel will be displayed

PGM    _________  TRAN  _________  DBD  _________  AREA _________
NODE   _________  LTERM _________  USER _________  RBA  _________
Recovery Token. . .  ________________________________   DRRN ________
UOW  _____________________________________________________________________
Scan DAVE_________________________________________________________________
Create Dynamic Search Keys _  (Y/N) Output DSN: VIGUERS.Keyword.KBLA.D.*

Log DSNs were extracted from RECON. _
PDS member containing logs. . . . . _________    Check LSN Seq _  (Y/N)
                    _
```

Figure 11-10 KBLA Log Data Set Analysis

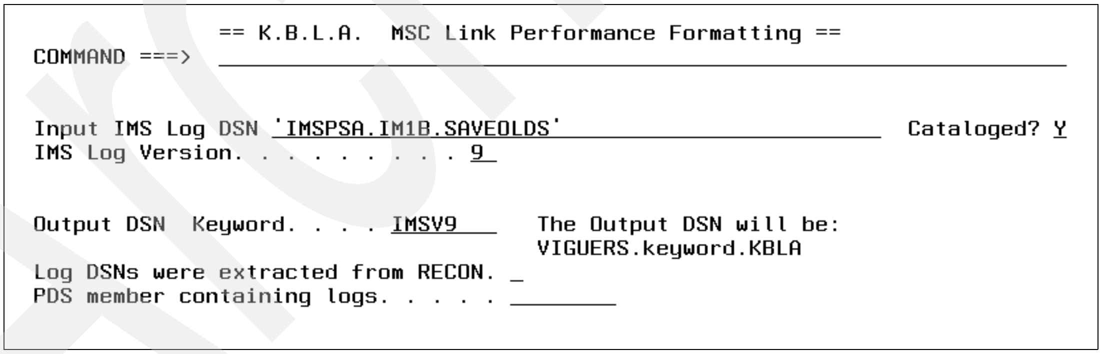
11.2.5 KBLA MSC Link Performance Formatting

Subtask 2 of KBLA is the MSC Link Performance Formatting function. This program uses the MSC link trace written to the log by the /TRA SET ON LINK n command. Multiple links may be traced and input to the program. Figure 11-11 on page 190 shows the panel used to generate the job to analyze the trace. The output of the formatting program shows a line for each link send or receive containing three different delta times and the actual time that operation completed so that send or receive can be matched against the full trace data for further analysis if necessary.

The delta times are:

- RECV DATA TO ACK - This is the number of milliseconds from a receive data trace record until the acknowledgement of that message is sent back to the other system. This time includes queuing of the message to the IMS message queue and a logger CHECKWRITE.
- SEND DATA TO ACK - This is the number of milliseconds from when IMS sends a message to another system and receives the acknowledgement that the message has been received and queued on that system. This time includes the time on the physical link.
- SEND CHECK WRITE - This is the time from when this link task issued the CHECK WRITE function until it was re-dispatched to continue processing. This time is included in the SEND DATA TO ACK time above. The time here included the actual I/O time (typically to the WADS) and the time until the IMS control task can re-dispatch this unit of work to continue.

If it is possible to run the link trace on the remote systems as well then the two can be compared to determine where any delays may be taking place. At the end of the report is a summary line for each of the links that were traced. If you look at the summary you can see that overall the largest delays were in the SEND DATA TO ACK but that the SEND CHECK WRITE component of that was relatively small. This would tend to indicate the delay was in the physical link processing which after some investigation was found to be the case in this example.



```
               == K.B.L.A.  MSC Link Performance Formatting ==
COMMAND ===>  ________________________________________________________________

Input IMS Log DSN 'IMSPSA.IM1B.SAVEOLDS'_______________________  Cataloged? Y
IMS Log Version. . . . . . . . . 9_

Output DSN  Keyword. . . . IMSV9___    The Output DSN will be:
                                       VIGUERS.keyword.KBLA
Log DSNs were extracted from RECON. _
PDS member containing logs. . . . . ________
```

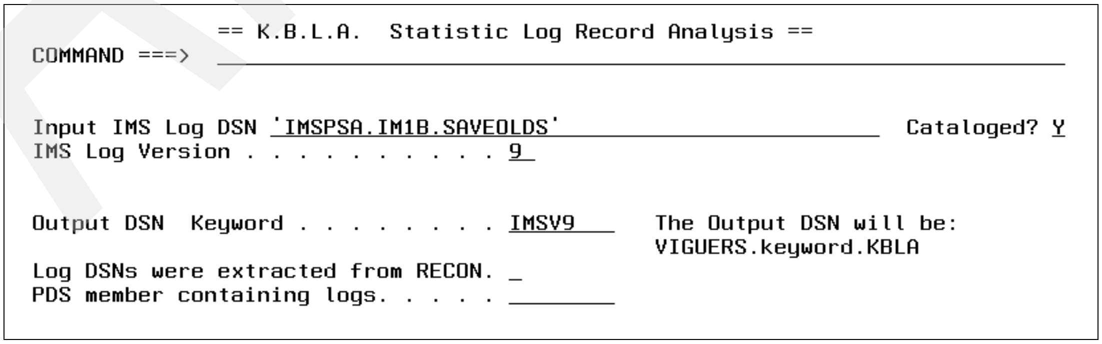
Figure 11-11   KBLA MSC Link Performance Formatting

Example 11-7   MSC Link Trace Formatting output

```
-                    RECV DATA TO ACK (MS)              SEND DATA TO ACK (MS) SEND CHECK WRITE (MS)          TIME
SEND FOR ID = EA                                                           69                    51   18:44:02.149
RECV FOR ID = EA                         9                                                            18:44:02.563
SEND FOR ID = EA                                                           63                     7   18:44:02.675
RECV FOR ID = CA                        18                                                            18:44:02.695
RECV FOR ID = EA                         8                                                            18:44:02.949
RECV FOR ID = AE                         6                                                            18:44:03.086
SEND FOR ID = EA                                                           57                    10   18:44:03.122
SEND FOR ID = AE                                                             59                     9   18:44:03.179
RECV FOR ID = EA                         9                                                              18:44:03.282
RECV FOR ID = AE                        14                                                              18:44:03.322
SEND FOR ID = EA                                                            122                    11   18:44:03.478
SEND FOR ID = AE                                                            181                    12   18:44:03.546
RECV FOR ID = EA                        12                                                              18:44:03.557
RECV FOR ID = AE                         9                                                              18:44:03.634
RECV FOR ID = AB                        10                                                              18:44:03.728
SEND FOR ID = AE                                                            120                     9   18:44:03.789
SEND FOR ID = EA                                                             90                    22   18:44:03.800
RECV FOR ID = EA                         7                                                              18:44:03.895
SEND FOR ID = AE                                                            110                     8   18:44:03.958
.....
.....
SEND FOR ID = EA                                                            180                    13   18:44:04.153
SEND FOR ID = AE                                                            179                     7   18:44:32.180
RECV FOR ID = EA                        38                                                              18:44:32.266
SEND FOR ID = AD                                                          1,556                    45   18:44:32.267
RECV FOR ID = AE                        14                                                              18:44:32.271
SEND FOR ID = AB                                                             15                    11   18:44:32.286
RECV FOR ID = DA                        36                                                              18:44:32.302
SEND FOR ID = DA                                                          1,552                    37   18:44:32.305
SEND FOR ID = AE                                                            152                    46   18:44:32.379
DFS707I END OF FILE ON INPUT
LINK PARTNER NAME    RECV DATA TO ACK (MS) # RECV SAMPLES SEND DATA TO ACK (MS) SEND CHECK WRITE (MS) # SEND SAMPLES
               AE                       14             80                   121                    15             79
               EA                       12             72                   115                    15             73
               CA                       12             13                    19                    14              5
               AB                       15             35                    27                    18             22
               BA                       13              7                    20                    12              7
               DA                       20             36                   725                    23             37
               AD                       21             36                   715                    27             37
DFS708I OPTION COMPLETE
DFS703I END OF JOB
```

11.2.6 KBLA Statistic Log Record Analysis

This function of KBLA will process the x'45' statistics log records written at each IMS checkpoint. Every checkpoint will create a complete set of statistics records with the exception of the **/CHE STATISTICS** command which will not create the IRLM statistics. The Statistic Log Record Analysis will create a report for each checkpoint interval so there must be at least two checkpoints on the log(s) being input to the program. The panel to create the JCL for this job is very simple and basically just requires the input log data set names as shown in Figure 11-12. An example of the output showing the general information about the checkpoint interval and a sample of the logger statistics is shown in Example 11-8.



```
              == K.B.L.A.  Statistic Log Record Analysis ==
COMMAND ===> ________________________________________________________________

Input IMS Log DSN 'IMSPSA.IM1B.SAVEOLDS'________________________  Cataloged? Y
IMS Log Version . . . . . . . . . . 9_

Output DSN  Keyword . . . . . . . . IMSV9___    The Output DSN will be:
                                                VIGUERS.keyword.KBLA
Log DSNs were extracted from RECON. _
PDS member containing logs. . . . . ________
```

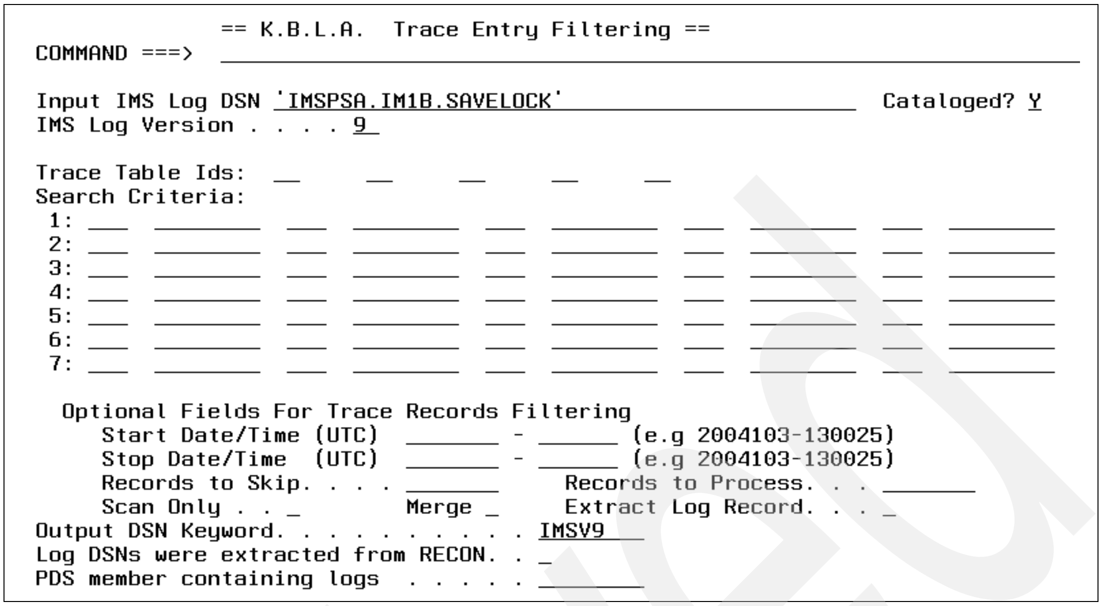
Figure 11-12 KBLA Statistic Log Record Analysis

Example 11-8 Partial output of Statistic Log Record Analysis

```
STATISTICS INFORMATION:
           BEGINNING YEAR/DAY           2004/210
           BEGINNING TIME (UTC)          2:12:46
           ENDING YEAR/DAY              2004/210
           ENDING TIME (UTC)             2:14:43
           OFFSET TO LOCAL (X 15M)          - 16
           ELAPSED TIME IN SECONDS        117.06
           COMPUTE TIME IN SECONDS           117
           IMS RELEASE             910
           REGION TYPE             ONLINE DB/DC
           IMSID                   IM1B
           OPERATING SYSTEM NAME   SC42
           OS PRODUCT              z/OS
           OS VERSION              010500
           FEATURE FLAGS           APPC=Y
                                   ETO=Y
                                   NOT XRF CAP.
                                   HSB=N
                                   NOT XRF SYNC
                                   NOT RSR CAP.
                                   SHARED QUEUES
                                   LSO=S
.....
.....
LOGGER STATISTICS                #         #/SEC
LOG BLOCK SIZE              26,624
NUMBER OF BUFFERS               10
WADS TRACK GROUPS                4
NUMBER OF RECORDS           54,447        465.35
CHECK WRITE REQUESTS         5,357         45.78
WAIT WRITE REQUESTS            497          4.24
WAIT 4 BUFF CKPT                 0           .00
WAIT 4 BUFF NON-CKPT             0           .00
AWE SUBMITTED ON WRT         1,632         13.94
WADS EXCPVRS                 3,034         25.93
2K SEGMENT WRITES            7,735         66.11
OLDS WRITES                    400          3.41
OLDS READS                       0           .00
INTERNAL CHKW REQ               13           .11
ACCUM WAIT TIME             30,782        263.09
```

11.2.7 KBLA trace entry filtering

Trace entry filtering allows you to extract trace records, which can be on the DFSTRAxx data sets or on the IMS log. In either case you can specify specific search criteria as a filter to locate only those entries related to a specific set of data. Using the panel shown in Figure 11-13, you can set the search criteria to W* meaning any word in the trace table entry and then a specific RBA that you may be concerned with, such as 0075BDE4, which happens to be one of the RBAs in the following lock trace analysis example. This allows you to search for only those entries in the trace which contain that RBA value. An example of the output using this search criteria is shown in Example 11-9 on page 193.



```
                == K.B.L.A.  Trace Entry Filtering ==
COMMAND ===> _________________________________________________________________

Input IMS Log DSN 'IMSPSA.IM1B.SAVELOCK'______________________  Cataloged? Y
IMS Log Version . . . . 9_

Trace Table Ids:  __     __     __     __     __
Search Criteria:
 1: ___  ________  ___  ________  ___  ________  ___  ________  ___  ________
 2: ___  ________  ___  ________  ___  ________  ___  ________  ___  ________
 3: ___  ________  ___  ________  ___  ________  ___  ________  ___  ________
 4: ___  ________  ___  ________  ___  ________  ___  ________  ___  ________
 5: ___  ________  ___  ________  ___  ________  ___  ________  ___  ________
 6: ___  ________  ___  ________  ___  ________  ___  ________  ___  ________
 7: ___  ________  ___  ________  ___  ________  ___  ________  ___  ________

  Optional Fields For Trace Records Filtering
     Start Date/Time (UTC)  _______ - ______ (e.g 2004103-130025)
     Stop Date/Time  (UTC)  _______ - ______ (e.g 2004103-130025)
     Records to Skip. . . . _______     Records to Process. . . _______
     Scan Only . . _        Merge _     Extract Log Record. . . _
Output DSN Keyword. . . . . . . . . . IMSV9___
Log DSNs were extracted from RECON. . _
PDS member containing logs  . . . . . ________
```

Figure 11-13 KBLA Trace Entry Filtering

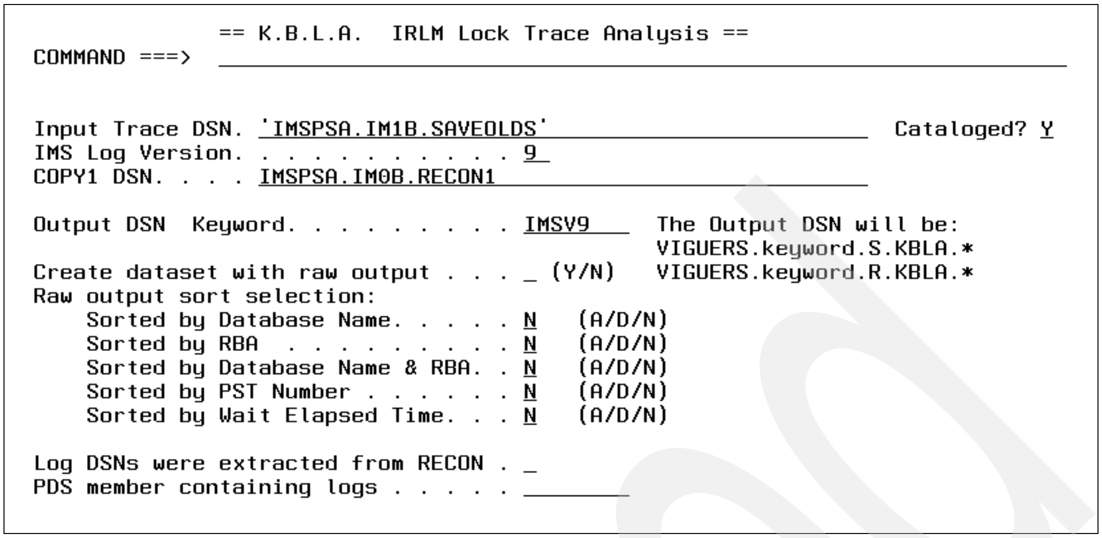
Example 11-9 Trace Entry Filtering output

```
FUNCTION        WORD 0    WORD 1    WORD 2    WORD 3    WORD 4    WORD 5    WORD 6    WORD 7                           PAGE 0001
* DLD TRACE TABLE - DATE 2004210 TIME 021343911336 OFFSET 016D SKIP 0000 TOTAL SKIP 00000000 RECORD NUMBER 8001B958
PSTGTSPC        31059EC2  001855E4  E2C90000  21B5F5E4  0075BDCC  0075BDE4  00110144  00000000  ...B...USI....5U.......U........
LOCK            C8059EC3  0206001E  1CB41060  00000000  09740580  0456CEC0  0075BDE4  800B01D7  H..C.......-...............U...P
GRIDU UNC U     CC059ECE  0075BDE4  7F740740  80400302  00000000  00000000  E8366FB4  8001B958  .......U".. . ..........Y.?.....
```

11.2.8 KBLA IRLM Lock Trace Analysis

The IRLM Lock Trace Analysis panel is shown in Figure 11-14. It uses as input the trace data created by using the command /TRA SET ON TABLE LOCK OPTION LOG. Normally you would allocated separate trace data sets to avoid this trace going to the IMS log since it creates a large amount of data. In this particular example the trace data sets were not available and so the trace data inadvertently went to the OLDS. There may be multiple output reports selected. By default you will get a summary report of the lock contention sorted by both database name and by total wait time for a database. A default detail report will also be generated showing lock contention by the time the request was granted.

The lock trace analysis input panel also allows you to specify that you want to create a separate output data set (also in request granted order) but without the heading lines. This is the create data set with raw output option. You could then use this output data set to process the data in any way you choose. In addition you may also request a report sorted by any of the criteria shown on the panel such as RBA or lock wait time depending on your needs. An sample of the summary report is shown in Example 11-10. A sample of the detail report in the default lock request completion time is shown in Example 11-11 on page 195.



```
                == K.B.L.A.  IRLM Lock Trace Analysis ==
COMMAND ===>   ________________________________________________________________

Input Trace DSN. 'IMSPSA.IM1B.SAVEOLDS'________________________  Cataloged? Y
IMS Log Version. . . . . . . . . . . 9_
COPY1 DSN. . . . IMSPSA.IM0B.RECON1____________________________

Output DSN  Keyword. . . . . . . . . IMSV9___  The Output DSN will be:
                                               VIGUERS.keyword.S.KBLA.*
Create dataset with raw output . . . _ (Y/N)   VIGUERS.keyword.R.KBLA.*
Raw output sort selection:
    Sorted by Database Name. . . . . N   (A/D/N)
    Sorted by RBA  . . . . . . . . . N   (A/D/N)
    Sorted by Database Name & RBA. . N   (A/D/N)
    Sorted by PST Number . . . . . . N   (A/D/N)
    Sorted by Wait Elapsed Time. . . N   (A/D/N)

Log DSNs were extracted from RECON . _
PDS member containing logs . . . . . ________
```

Figure 11-14 KBLA IRLM Lock Trace Analysis

Example 11-10 IRLM Lock Trace Analysis summary report

```
-         Suspended IRLM Lock Requests Summary Report - Wait Time Order Page 001
Trace Date = 07/27/2004  Trace Start Time = 22:13:43  Trace End Time = 22:14:15
Trace Elapsed Time (secs) =      31
Trace Input DSN = IMSPSA.IM1B.SAVELOCK

   Database  DS   Lock Req    Wait  Not Int      Total    Average   Maximum
       Name  Id      Count   Count    Count       Time       Time      Time

   DISTDB    01        444      85       85      3.570      0.042      1.54
   WAREDB    01        424     167      167      0.991      0.005      0.19
   STCKDB    02       1493     755      755      0.895      0.001      0.15
   STCKDB    05       1966     811      811      0.766      0.000      0.16
   STCKDB    04       1205     622      622      0.602      0.000      0.23

   ITEMDB    01       2102    1184     1182      0.529      0.000      0.08
   STCKDB    03       1037     395      395      0.487      0.001      0.26
   STCKDB    01        501     186      186      0.148      0.000      0.06
   ORDRSI    01        965     336      336      0.071      0.000      0.05
   ORDLDB    02        292      85       85      0.046      0.000      0.03

   NORDDB    01        857     438      438      0.043      0.000      0.01
   NOIX      01        856     290      290      0.024      0.000      0.01
   ORDLDB    01        172      65       64      0.023      0.000      0.01
   CUSTDB    01       1126     475      466      0.016      0.000      0.00
   ORDRDB    01        493     242      242      0.016      0.000      0.00

   CUSTSI    01       5781    1719     1718      0.015      0.000      0.00
   HISTDB    01        209      32       32      0.002      0.000      0.00
   FP DMY    00        424       1        0      0.000      0.000      0.00

    Totals           20347    7888     7874      8.251      0.001      1.54
```

Example 11-11 IRLM Lock Trace Analysis detail report

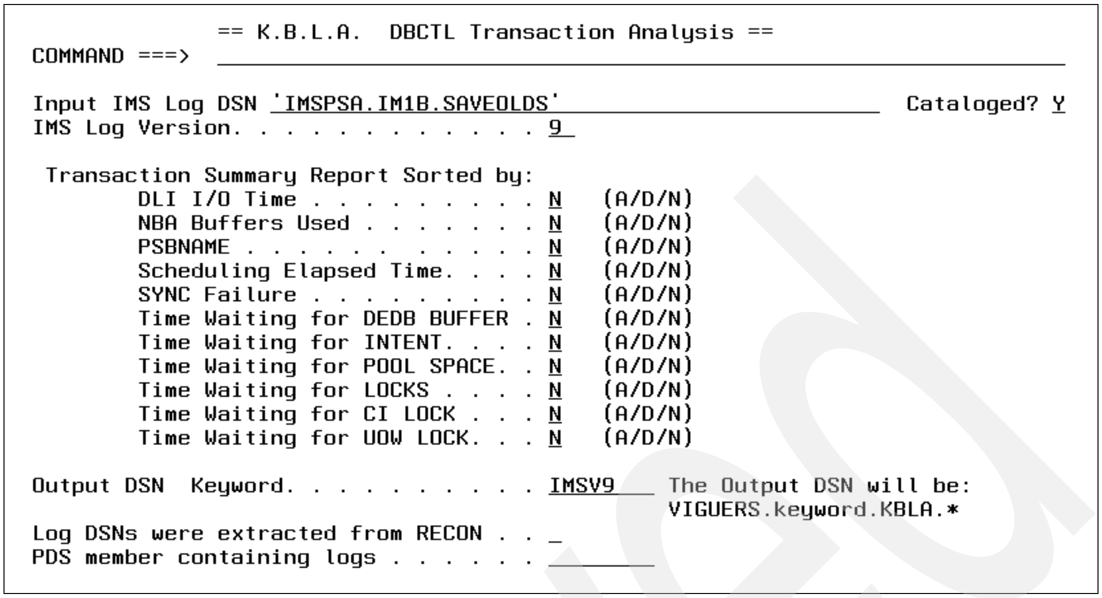
```
Trace Date = 07/27/2004 DSN = IMSPSA.IM1B.SAVELOCK
Lock Request Lock Request ----Wait----- PST --Lock-- -------Resource------- Flag --IRLM--- --------Call--------- Trace
 Start Time    End Time    Elapsed Type Num Type Lvl    DB    DS RBA/HASH S      RCFB TRAC Type Num      Time     Seq#

22:13:43.766 22:13:43.766     0.004  F  001 BIDP  4  CUSTSI   01 03540001 P CPKF 0000 08C0 GN   082 22:13:43.766  9509
22:13:43.774 22:13:43.774     0.004  F  001 BIDP  6  CUSTDB   01 0FD04014 P PKF  0000 08C0 GHU  369 22:13:43.774  97E6
22:13:43.783 22:13:43.783     0.004  F  001 BIDP  4  CUSTDB   01 15796000 P CPKF 0000 08C0 REPL 369 22:13:43.783  9865
22:13:43.811 22:13:43.811     0.004  F  001 BIDP  4  CUSTSI   01 035E9001 P CPKF 0000 08C0 GN   105 22:13:43.811  99DA
22:13:43.887 22:13:43.887     0.004  F  005 BIDP  4  CUSTDB   01 12F4A000 P CPKF 0000 08C0 GU   735 22:13:43.887  9E6B

22:13:43.895 22:13:43.895     0.004  F  005 BIDP  6  NORDDB   01 007582D4 P CPKF 0000 08C0 ISRT 469 22:13:43.895  9E90
22:13:43.895 22:13:43.895     0.004  F  005 BIDP  6  NORDDB   01 0075BDCC P CPKF 0000 08C0 ISRT 469 22:13:43.895  9EA6
22:13:43.895 22:13:43.895     0.004  F  005 BIDP  4  NORDDB   01 00756800 P CPKF 0000 08C0 ISRT 469 22:13:43.895  9EB4
22:13:43.895 22:13:43.895     0.004  F  005 BIDP  6  NORDDB   01 0075BDE4 P CPKF 0000 08C0 ISRT 469 22:13:43.895  9EC3
22:13:43.895 22:13:43.895     0.004  F  005 BIDP  6  NOIX     01 AC308505 P PKF  0000 08C0 ISRT 469 22:13:43.895  9ED1

22:13:43.895 22:13:43.895     0.004  F  005 BIDP  4  NOIX     01 00546001 P CPKF 0000 08C0 ISRT 469 22:13:43.895  9EEA
22:13:43.911 22:13:43.911     0.004  F  005 BIDP  6  ORDRDB   01 0065D0E0 P PKF  0000 08C0 ISRT 470 22:13:43.911  9F18
22:13:43.911 22:13:43.911     0.004  F  005 BIDP  4  ORDRDB   01 0065D000 P CPKF 0000 08C0 ISRT 470 22:13:43.911  9F2E
22:13:43.911 22:13:43.911     0.004  F  005 BIDP  6  ORDRSI   01 638BE1CF P PKF  0000 08C0 ISRT 470 22:13:43.911  9F3F
22:13:43.911 22:13:43.911     0.004  F  005 BIDP  4  ORDRSI   01 00C90001 P CPKF 0000 08C0 ISRT 470 22:13:43.911  9F61
```

11.2.9 KBLA DBCTL Transaction Analysis

The DBCTL Transaction Analysis uses information from the IMS log similar to what is reported by the utilities DFSILTA0 and DBFULTA0. It reports on a single line for each transaction both the Full Function information from the x’07’ log record and any Fast Path information from the x’5937’ log record.

While this program may be run with input from any IMS log it is most meaningful in the case where only one transaction is processed per schedule thus there is one x’07’ record for each transaction. This is the case in a DBCTL environment but is usually not so where IMS is the transaction manager.

As you can see from the selection panel shown in Figure 11-15 you may also select the order in which the report is sorted. The default is by program termination time but you may request the order to be any of the fields shown. For instance if you were having a locking problem you might choose to sort the report by Time Waiting for LOCKS. A sample of the output in the default order of program termination is shown in Example 11-12. You may also notice that where some values exceed the length of the report field a > sign will precede the value.



```
               == K.B.L.A.  DBCTL Transaction Analysis ==
COMMAND ===> ________________________________________________________________

Input IMS Log DSN 'IMSPSA.IM1B.SAVEOLDS'________________________  Cataloged? Y
IMS Log Version. . . . . . . . . . . . 9_

 Transaction Summary Report Sorted by:
        DLI I/O Time . . . . . . . . . N   (A/D/N)
        NBA Buffers Used . . . . . . . N   (A/D/N)
        PSBNAME . . . . . . . . . . .  N   (A/D/N)
        Scheduling Elapsed Time. . . . N   (A/D/N)
        SYNC Failure . . . . . . . . . N   (A/D/N)
        Time Waiting for DEDB BUFFER . N   (A/D/N)
        Time Waiting for INTENT. . . . N   (A/D/N)
        Time Waiting for POOL SPACE. . N   (A/D/N)
        Time Waiting for LOCKS . . . . N   (A/D/N)
        Time Waiting for CI LOCK . . . N   (A/D/N)
        Time Waiting for UOW LOCK. . . N   (A/D/N)

Output DSN  Keyword. . . . . . . . . . IMSV9___  The Output DSN will be:
                                                  VIGUERS.keyword.KBLA.*

Log DSNs were extracted from RECON . . _
PDS member containing logs . . . . . . ________
```

Figure 11-15   KBLA DBCTL Transaction Analysis

Example 11-12   DBCTL Transaction Analysis output

```
PSBNAME    SUBSYS R RGN ELAP SCHED_TIME SCHEL INT PWT DLI IOC   IOT    LWT DEC DEG DEP OVF BWT NBA UPD SDP CLK ULK VRD VDR VWR S
               ID T NBR SS.T HH:MM:SS.T    MS  MS  MS   #   #    MS     MS   #   #   #   #   #   #   #   #   #   #   #   #   # F
DVPOSO   CICSAOR3 D 22    .0 08:02:15.2     1   0   0   2   3     2      0  14   4   0   0   0   9   0   0   0   0   6   0   0
DVPOSO   CICSAOR3 D 22    .0 08:02:15.2     1   0   0   0   0     0      0  12   2   0   0   0   7   1   0   0   0   5   0   0
DVPOSO   CICSAOR1 D 12    .0 08:02:15.2     1   0   0   0   0     0      0  24  12   0   0   0  21   7   0   0   0  11   0   1
DVPOSO   CICSAOR3 D 22    .1 08:02:15.2     0   0   0   0   0     0      0  12   2   0   0   0   7   0   0   0   0   5   0   0
DVLOSO   CICSAOR3 D 35    .0 08:02:15.3     0   0   0   1   0     0      0  11   3   0   0   0  10   0   0   0   0   8   0   0
DVPOSO   CICSAOR1 D 12    .1 08:02:15.2     1   0   0   2   1     1      0  30   4   0   0   0  11   2   0   0   0   8   0   0
DVROSO   CICSAOR2 D 63    .0 08:02:15.3     1   0   0  12  11     9      0  84  21   0   0   0  17  13   0   0   0  22   0   0
DVAOSO   CICSAOR1 D 12    .1 08:02:15.3     1   0   0   0   0     0      0  22   8   0   0   0  15   3   0   0   0   8   0   0
DVPOSO   CICSAOR3 D 22    .1 08:02:15.3     1   0   0   2   1     0      0  28   4   0   0   0   9   2   0   0   0   6   0   0
DVLOSO   CICSAOR2 D 63    .1 08:02:15.3     0   0   0   8   0     0      0  14   4   0   0   0  12   2   0   0   0   8   0   0
DVPOSO   CICSAOR2 D 63    .0 08:02:15.4     1   0   0   0   0     0      0  12   2   0   0   0   8   1   0   0   0   5   0   0
DVGOSO   CICSAOR1 D 21    .1 08:02:15.3     1   0   0  16  34    30      0  34  15   0   0   0  23   7   0   0   0  10   0   0
DVAOSO   CICSAOR3 D 22    .0 08:02:15.4     1   0   0   2   3     2      0  12   3   0   0   0  12   0   0   0   0  10   0   0 U
DVAOSO   CICSAOR3 D 22    .0 08:02:15.4     1   0   0   0   0     0      0   4   1   0   0   0   1   1   0   0   0   0   0   0
DVEOSO   CICSAOR3 D 35    .2 08:02:15.3     1   0   0  23  66    64      0  61  26   0   0   0  22  11   0   0   0  18   0   1
DVPOSO   CICSAOR2 D 63    .0 08:02:15.5     1   0   0   0   0     0      0  12   2   0   0   0   8   0   0   0   0   6   0   0
DVLOSO   CICSAOR1 D 12    .0 08:02:15.5     0   0   0   8   0     0      0  14   4   0   0   0  12   2   0   0   0   8   0   0
DVAOSO   CICSAOR1 D 21    .1 08:02:15.4     1   0   0   4   3     3      0  47  14   0   0   0  29   8   0   0   0  17   0   0
DVLOSO   CICSAOR3 D 35    .1 08:02:15.5     0   0   0   8   0     0      0  14   4   0   0   0  12   2   0   0   0   8   0   0
DVLOSO   CICSAOR2 D 75   1.8 08:02:13.8     1   0   0 935>999  1376      0  15   3   0   0   0   8   1   0   0   0   6   0   0
DVROSO   CICSAOR1 D 21    .0 08:02:15.6     1   0   0   0   0     0      0  28   6   0   0   0  17   0   0   0   0  11   0   0
DVAOSO   CICSAOR3 D 35    .0 08:02:15.6     1   0   0   0   0     0      0  22  10   0   0   0  18   3   0   0   0   8   0   0
......
......
DVPOSO   CICSAOR2 D 74    .1 08:02:27.0     1   0   0   0   0     0      0  12   2   0   0   0   7   0   0   0   0   5   0   0
DVLOSO   CICSAOR2 D 75    .0 08:02:27.1     0   0   0   6  12     8      0   0   0   0   0   0   0   0   0   0   0   0   0   0
DVPOSO   CICSAOR3 D 22    .0 08:02:27.1     1   0   0   2   2     1      0  32   6   0   0   0  12   2   0   0   0   6   0   0
DVLOSO   CICSAOR2 D 75    .0 08:02:27.1     1   0   0   2   0     0      0  15   2   0   0   0  10   0   0   0   0   8   0   0 U
DVGOSO   CICSAOR1 D 21    .7 08:02:26.4     1   0   0  20  32    27    659  34  12   0   0   0  20   8   0   0   0  10   0   0
DVPOSO   CICSAOR1 D 21    .0 08:02:27.2     1   0   0   0   0     0      0  12   2   0   0   0   8   1   0   0   0   6   0   0
DVEOSO   CICSAOR2 D 74    .0 08:02:27.2     1   0   0   0   0     0      0  14   4   0   0   0  13   0   0   0   0   9   0   0
......
......
DFS707I END OF FILE ON INPUT
NUMBER OF SYNC RECORDS  (5937)      132871
NUMBER OF SYNC FAILURES (5938)        9425
NUMBER OF SCHD RECORDS  (08)        134238
NUMBER OF TERM RECORDS  (07)        134242
DFS708I OPTION COMPLETE
DFS703I END OF JOB
```

12

Miscellaneous enhancements in IMS Version 9

IMS Version 9 contains over 50 enhancements, all of them responding to specific customer requirements. In this chapter we have gathered together the miscellaneous enhancements that haven't been described in their own chapters previously in this book:

- Database Manager enhancements:

- Transaction Manager enhancements:

- System manageability enhancements:

12.1 Miscellaneous Database Manager enhancements

In this section we describe the following Database Manager enhancements:

- 12.1.1, “Parallel full function database open option” on page 200
- 12.1.2, “Miscellaneous HALDB enhancements” on page 200
- 12.1.3, “Improved message with DB abends” on page 202
- 12.1.4, “Greater than 32K tape block size for image copies” on page 202

12.1.1 Parallel full function database open option

IMS Version 8 introduced the use of parallel DB open. This function is invoked by IMS restarts, /NRE and /ERE. Databases which were open when IMS was last terminated are opened during restart instead of waiting for the first use of the databases.

IMS Version 9 makes the use of this function optional. The NOPDBO control statement in the DFSVSMxx member is used to turn it off. When a NOPDBO statement is used, databases are not authorized, allocated and opened as part of restart processing.

12.1.2 Miscellaneous HALDB enhancements

IMS Version 9 provides also the following HALDB enhancements:

- HALDB specific partition initialization
- DBRC option enhancement for HALDB
- HALDB processing control statement enhancement
- HALDB partition selection exit DSECT enhancement
- HALDB partition selection exit customization enhancement
- HD Unload and Reload utilities (DFSURGU0 and DFSURGL0) are enhanced to generate statistics for each HALDB partition

HALDB specific partition initialization

Originally HALDB partition initialization utilities initialized all of the partitions in a database for which the *partition initialization needed flag* (PINIT) in RECON was set. These utilities are the HALDB partition data set initialization utility (DFSUPNT0) and the database pre reorganization utility (DFSURPR0).

There is one exception. DFSUPNT0 has a capability to initialize all of the partitions in a database without regard to the PINIT flag. This was added to IMS Version 8 by APAR PQ55002 and to IMS Version 7 by APAR PQ49638. It allows a user to initialize all of the partitions in a database without having to issue any DBRC CHANGE.DB commands to change the PINIT flags. This unconditional partition initialization is invoked by including a DFSOVRDS DD statement with an INITALL control statement. When a partition is initialized, the *partition initialization needed* flag is turned off.

IMS Version 9 adds a capability to initialize specific partitions. This is done by specifying the partition name in the control statement for the HALDB Partition Initialization utility (DFSUPNT0).

The control statements for DSFSUPNT0 may include both databases and partitions. For the databases on control statements, the utility initializes only the partitions which have the *partition initialization needed* flag on in the RECONs. For the partitions on control statements, the utility initializes them no matter what the setting is for the PINIT flag.

Example 12-1 is an example of the use of the HALDB partition data set initialization utility (DFSUPNT0) to invoke both specific partition initialization and the former function.

Example 12-1 HALDB partition data set initialization utility (DFSUPNT0)

```
//PINIT EXEC PGM=DFSUPNT0,REGION=2048K
//STEPLIB  DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//IMS      DD DSN=IMS.DBDLIB,DISP=SHR
//DFSVSAMP DD DSN=IMS.DFSVSAMP(PINIT),DISP=SHR
//SYSUDUMP DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
DBABC011
DBEFG022
PAR00AB
PAR78CD
/*
```

In this example, DBABC011 and DBEFG022 are databases. The utility will initialize only their partitions which have the *partition initialization needed* flag set. PAR00AB and PAR78CD are partitions. The utility will initialize these partitions without regard to their PINIT flags, on or off. In any case, all partitions which are initialized will have their PINIT flags turned off.

There are two ways to execute the HALDB partition data set initialization utility (DFSUPNT0). The example shows the execution as a stand-alone program. That is, the utility is executed with PGM=DFSUPNT0 on the EXEC statement. The utility also may be executed in a ULU utility region. In this case, the EXEC statement has PGM=DFSRRC00 and the PARM list begins with PARM=(ULU,DFSUPNT0,dbname,...) where 'dbname' is the name of a HALDB. This form of execution does not allow one to specify a partition name. It operates on only one database.

DBRC option enhancement for HALDB

DBRC option enhancement for HALDB allows a batch program that references a HALDB to run without DBRC so long as no database calls refer to the HALDB during execution. This provides more flexibility in running a single program in different environments, thus reducing application developer and operations efforts that might be needed to convert programs and ensure their running only in specific environments. This enhancement is also being provided through the IMS Version 7 and IMS Version 8 service process.

HALDB processing control statement enhancement

HALDB processing control statement enhancement increases the number of HALDB single partition processing control statements from 10 to 20 for usability. This allows a customer's application to specify up to 20 DB PCBs that are allowed to perform single partition processing in a HALDB. This enhancement is also being provided through the IMS Version 7 and IMS Version 8 service process.

HALDB partition selection exit DSECT enhancement

HALDB partition selection exit DSECT enhancement allows customers to use the DFSPSEIB macro to get the partition selection exit DSECTs. This makes HALDB with a user-written partition selection exit easier to use. Customers no longer have to create their own DSECTs for the partition selection exit parameters. This enhancement is also being provided through the IMS Version 7 and IMS Version 8 service process.

HALDB partition selection exit customization enhancement

HALDB partition selection exit customization enhancement makes HALDB with a user-written partition selection exit more customizable and easier to use. This allows a partition selection exit to run differently, depending on IMS and the application running. This enhancement is being provided through the IMS Version 9 service process.

HD Unload and Reload utilities statistics

HD Unload and Reload utilities (DFSURGU0 and DFSURGL0) are enhanced to generate statistics for each HALDB partition, previously they had statistics for the entire HALDB database. HD Reload has been enhanced to read partition statistics in the unload file and to support concatenated unload files. This enhancement is also available through the IMS Version 7 and Version 8 service process.

12.1.3 Improved message with DB abends

In previous releases message DFS0832I is issued for HALDB abends, U0880, U3303 and U3498. It is issued prior to abends to indicate which database or partition is affected.

IMS Version 9 also issues this message for some non-HALDB abends, such as U085x. The advantage is that it identifies the database in error. The other error messages associated with these abends, such as DFS554A, do not identify the database. Example 12-2 shows DFS0832I format.

Example 12-2 DFS0832I

```
DFS0832I ABEND Uxxxx REASON CODE yyyy <PARTITION|DATABASE> zzzzzzzz
```

The field xxxx shows abend code, yyyy reason code, PARTITION for HALDB, or DATABASE for non-HALDB, zzzzzzzz DBD name. If this message is issued for non-HALDB, reason code, which is shown as yyyy, is not applicable. If DBD name is not available, zzzzzzzz field is shown as NOTAPPLI which is default value of zzzzzzzz.

12.1.4 Greater than 32K tape block size for image copies

Large tape block sizes, a capability introduced in OS/390 V2R10 and known as the large block interface (LBI), is supported with IMS Version 9.

When a block size is not specified and a device supporting large tape block sizes is used, or when a block size larger than 32K is specified, the Image Copy utility (DFSUDMP0) will use the large (>32K) block size. This is also supported by the utilities that read the image copies produced by the Image Copy utility. These are the database recovery utility (DFSURDB0) and the Database Recovery Facility product (program number 5655-I44) with PQ79327. It is not supported by Online Image Copy utility or Image Copy 2.

When large tape block sizes are used, image copy issues the following message:

Example 12-3 DFS3185I

```
DFS3185I LARGE BLOCK INTERFACE DETECTED, IT WILL BE USED FOR DDNAME....
```

This message explains that the image copy utility has determined that the image copy output data set is on a device that supports the large block interface and no explicit block size was specified. 32K tape blocksize support requires the large block interface on 3480 magnetic tape, 3490 and 3490E magnetic tape subsystems, or 3590 devices.

12.2 Miscellaneous Transaction Manager enhancements

In this section we describe the following Transaction manager enhancements:

- 12.2.1, “OTMA and APPC /EXIT command support” on page 203
- 12.2.2, “Message level ACEE aging value for OTMA” on page 204
- 12.2.3, “OTMA trace table enhancement” on page 206
- 12.2.4, “DFSYPRX0 pre-routing exit” on page 206
- 12.2.5, “Input message ZZ protection for OTMA” on page 207
- 12.2.6, “APPCIOT= in DFSDCxxx for IMS/APPC application time-out” on page 208
- 12.2.7, “LU type 3 logon option as ETO SLUTYPE1 or 3270P” on page 208
- 12.2.8, “Notify CQS outage to terminal users” on page 209
- 12.2.9, “Greater than 255 transaction classes” on page 209

12.2.1 OTMA and APPC /EXIT command support

IMS Version 9 provides /EXIT command support to APPC and OTMA conversational processing.

Releases prior to IMS Version 9 have had an outstanding issue in APPC and OTMA environments for IMS conversational processing. There was no method to end an APPC or OTMA conversation externally. This affected, for example, online change related to that conversational transaction and its related storage. For online change, all conversations need to be terminated beforehand. For the storage, all the related control blocks remain in the storage if the client does not terminate the conversation. If a remote client does not cleanly terminate the IMS conversation such that IMS considers it still active, then all the related control blocks of this “hung” interaction remain in storage. Every time IMS takes a system checkpoint, the control blocks are written to the log. A large number of these types of conversations that hang around indefinitely could potentially impact system checkpoint performance.

The support relied on either the IMS application or the remote APPC or OTMA program to cleanly terminate the conversation. For IMS applications this meant placing a blank transaction code in the SPA. For APPC this meant issuing a DEALLOCATE verb which cleaned up both the APPC conversation and the IMS conversation. For OTMA this meant that the remote program sent a message with the commit-confirmation flag set in the message-control information section of the message prefix (indicating a deallocate flow). If none of the above were done, or if the remote program took a long time in sending the reply then IMS processes such as online change could not complete.

IMS Version 9 introduces new parameters in the /EXIT command that affect active IMS Conversational programming flows in the APPC and OTMA environments:

```
/EXIT CONVERSATION conv# LUNAME luname
/EXIT CONVERSATION conv# TMEMBER tmembername TPIPE tpipename
```

LUNAME, TMEMBER, and TPIPE are new parameters for the /EXIT command. /EXIT with LUNAME terminates both the APPC conversation and the IMS conversation on the luname specified. /EXIT with TMEMBER and TPIPE terminates the IMS conversation on the tmembername and tpipename combination specified.

Output from the **/DISPLAY CONVERSATION** command in prior releases did not display information about the OTMA. IMS Version 9 supports an enhancement to the command to display the OTMA information, TMEMBER and TPIPE. The APPC information was supported in the prior releases. This information is necessary for identifying the conversations when issuing the **/EXIT** command for this environment. Example 12-4 on page 204 shows /EXIT command output.

Example 12-4 /EXIT command output

```
IEE600I REPLY TO 597 IS;/DIS CONV
DFS000I     TERMINAL          USER     ID   STATUS   IMSG
DFS000I     HWS910G.7003      JOUK01   0001 ACTIVE   IMSG
DFS000I     *04211/140031*   IMSG
```

TMEMBER and TPIPE are displayed on TERMINAL column as HWS910G and 7003.

Note that the /EXIT command enhancements in IMS Version 9 apply to issuing the command from an IMS terminal. Remote programs are not allowed to pass in /EXIT as a target name. This has been true for all previous releases and continues to be true in IMS Version 9. The MQSeries® externalization of a /EXIT is actually trapped by MQ and converted into a commit-confirmation flag to indicate a deallocate flow in the accepted protocol - the /EXIT does not actually flow as a command into IMS.

In an IMSplex environment where there are multiple IMS systems, care should be taken to ensure that the /EXIT command is issued on the correct IMS system that owns the conversation. Additionally, note that for APPC and OTMA, a conversation id is only unique within an IMS system and not necessarily across all the systems in an IMSplex even with sysplex terminal management.

12.2.2 Message level ACEE aging value for OTMA

IMS Version 9 allows the OTMA client to specify an override for the ACEE aging value for any user ID.

When an input message from an OTMA client arrives in IMS, a call to RACF or equivalent security product is issued to build an ACEE control block for the user ID associated with the message. OTMA uses a hash table to maintain pointers to ACEEs so that if the same user ID enters another transaction from the OTMA client, the previously created ACEE is reused. Use of the existing ACEE eliminates a call to RACF. The ACEEs are kept in storage based on a defined aging value that is established at client-bid process in which OTMA client sends client-bid message toward OTMA to connect IMS. The ability to reuse an ACEE rather than create it for every request, is a performance benefit particularly if the same user ID sends in multiple transaction requests.

In prior releases, the aging value was established globally for all user IDs as part of the client-bid process. For more details of client-bid process, see *IMS Version 9: Open Transaction Manager Access Guide and Reference,* SC18-7829. This is still the way to specify the default value in IMS Version 9.

Sometimes, however, it may be necessary to refresh the cached ACEEs during the lifetime of the ACEE aging value, for example, the user ID has been revoked and the ACEE has not yet expired. There are two ways to do this:

- Stop OTMA by /STOP OTMA and restart OTMA by /START OTMA
- Invoke the enhanced format of the /SECURE OTMA command:

  `/SECURE OTMA REFRESH [TMEMBER tmembername]`

With APARs PQ68350 (IMS Version 7) and PQ66549 (IMS Version 8), the /SECURE OTMA command can request a REFRESH of the cached ACEEs. Issuing the /SEC OTMA REFRESH command without the TMEMBER results in a refresh of all the user IDs for all the OTMA clients. The addition of the TMEMBER parameter with an associated tmembername requests a refresh of the ACEEs associate with just that tmembername.

In both cases, however, the actual ACEE refresh occurs when the next OTMA message for a user ID is received. This is designed to prevent all the RACF ACEE refresh from happening at one time.

IMS Version 9 has enhanced the support in this area to allow for greater granularity on a user ID level. OTMA clients are now able to set an override security ACEE aging value in the OTMA message prefix. For more details of OTMA message prefix, see *IMS Version 9: Open Transaction Manager Access Guide and Reference,* SC18-7829.

There are two fields that support this capability:

- TMAMAGNG in the message-control information section of the OTMA message prefix
  - This is a 4 byte field in byte 20 of the message-control information. The value that can be specified in this field defines an override ACEE aging value for the user ID in seconds. The offset is the same as the existing TMAMCSNS field which stores the sense code and reason code if an OTMA message error is detected.
  - The minimum value for caching support is 300 seconds (5 minutes). If the aging value specified is less than the minimum, IMS always creates a non-cached ACEE.
- TMAMSAGN in the state data section of the OTMA message prefix
  - This is a bit in the TMAMHCFL in byte 5 of the state data, which signifies to OTMA that the aging value in TMAMAGNG is to be used to override the default value.

Both indicators, TMAMAGNG and TMAMSAGN, are reset before the message is enqueued.

Although the ACEE aging value is set, if OTMA does not have security turned on with either a value of CHECK or FULL then the ACEEs are not created or cached in IMS. A refresh of the ACEE is requested in one of the following circumstances:

- Message level aging value specified in the message expires
- New aging value is specified in the message
- Client-bid aging value expires
- /SECURE OTMA REFRESH command is issued
- /STOP and subsequent /START OTMA
- OTMA client disconnect followed by reconnect (client bid)

As documented earlier, the actual ACEE refresh occurs when the next OTMA message for the user ID is received.

When determining a value to specify for the ACEE aging value, a primary consideration involves understanding that it is meant to cache information about the user for the purpose of minimizing I/O to the RACF (or underlying security product) data sets. If a value of zero is used, then ACEE caching is turned off and every message that carries the user ID results in accessing RACF for the information to build the ACEE (this assumes that OTMA security is enabled).

Although OTMA provides the capability for a message level override of the ACEE aging value, it is actually up to the OTMA client to decide whether or not this should be implemented in their environments and, if so, how it should be implemented. For example, IMS Connect already provides access to the OTMA message prefix through its message exits. For an IMS Version 9 environment, a user message exit could easily be modified to set the appropriate flags and fields for the ACEE override. Additionally, the exits could enhance the client interface to optionally allow the client headers to also specify an override value.

12.2.3 OTMA trace table enhancement

New OTMA trace table entries for trace activity in and out of the exits as X’5A05’-X’5A0A’ are supported in IMS Version 9.

OTMA supports the creation of new OTMT trace table entries in the effort to standardize OTMA traces and conform to the tracing that is done for other OTMA modules. The existing exit entries do not follow the OTMA table entry standard that is documented in the Diagnosis Guide and Reference manual. This enhancement provides customers and IBM service support personnel with an easier way to check the return code set by the user exits.

Whenever an OTMA user exit is called, an OTMT table trace record is cut to trace the input parameters associated with the exit. After the call to the exit, another OTMT table trace entry is cut to trace the return code set by the exit. This enhancement applies to all the OTMA exit routines, DFSYIOE0, DFSYPRX0, and DFSYDRU0.

- DFSYIOE0 - OTMA input output edit exit routine
  - x'5A05' when entering the exit
  - x'5A06' when leaving the exit, word 3 contains the exit RC
- DFSYPRX0 - OTMA pre routing exit routine
  - x'5A07' when entering the exit
  - x'5A08' when leaving the exit, word 3 contains the exit RC
- DFSYDRU0 - OTMA destination resolution exit routine
  - x'5A09' when entering the exit
  - x'5A0A' when leaving the exit, word 3 contains the exit RC

This now conforms to the way other OTMA modules cut trace entries which are: OTMA module entry (x'5A01') and OTMA module exit (x'5A02').

12.2.4 DFSYPRX0 pre-routing exit

When ALTPCB output destination is created in an OTMA environment, the processing of the message is controlled by whether or not any OTMA exits exist. The OTMA pre-routing exit (DFSYPRX0) routing provides a mechanism to control whether OTMA or IMS TM is called to handle the message. See Figure 12-4 on page 204 for this process.

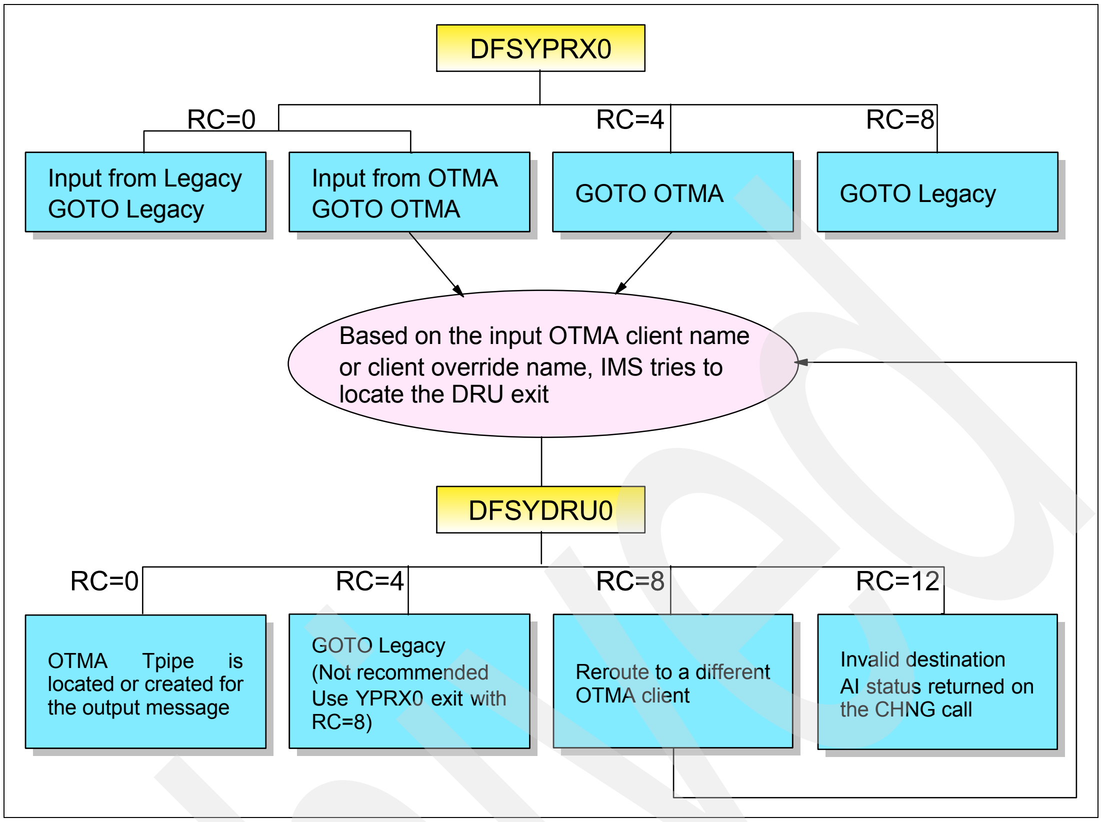

DFSYPRX0
RC=0
RC=4
RC=8
Input from Legacy GOTO Legacy
Input from OTMA GOTO OTMA
GOTO OTMA
GOTO Legacy
Based on the input OTMA client name or client override name, IMS tries to locate the DRU exit
DFSYDRU0
RC=0
RC=4
RC=8
RC=12
OTMA Tpipe is located or created for the output message
GOTO Legacy (Not recommended Use YPRX0 exit with RC=8)
Reroute to a different OTMA client
Invalid destination AI status returned on the CHNG call

Figure 12-1 DFSYPRX0 and DFSYDRU0

A second exit routine, the OTMA destination resolution exit (DFSYDRU0) can be further invoked when the destination is an OTMA client. DFSYDRU0 can be coded to specify the target OTMA TMEMBER name, the TPIPE name to be used, and whether a synchronized or non-synchronized TPIPE is to be created.

Over several releases, IMS has provided the capability to define the target TMEMBER and type of TPIPE outside DFSYDRU0. The TMEMBER can be specified as a client override name in DFSYPRX0. The TPIPE type can be specified in an IMS execution parameter OTMASP.

IMS Version 9 adds the ability to specify the target TPIPE name in DFSYPRX0. Prior to Version 9, DFSYPRX0 determines whether ALTPCB message are to be processed by OTMA or non-OTMA destinations, and for OTMA destinations, TMEMBER can be specified by this exit routine. In IMS Version 9, the specification for TPIPE override name in DFSYPRX0 has been added to the end of the input parameter list at offset 84. With this enhancement, the exit routine can specify both the TMEMBER and TPIPE names.

12.2.5 Input message ZZ protection for OTMA

In prior releases, OTMA always zeroed out the values in the ZZ field of the input LLZZ application data. In IMS Version 9, the bit settings are untouched and sent in to IMS. If necessary, OTMA exit DFSYIOE0 which has access to all messages inbound and outbound, can set or modify the value.

This could be important to IMS applications that were originally coded for sensitivity to MFS options. The ZZ field in (non-OTMA) MFS messages contains the MFS formatting option being used to format the messages to and from the IMS application program. Although OTMA does not invoke MFS, the existing IMS application program may be sensitive to this information and expect to see a value in the field. Remote clients, understanding that they are communicating with a program that looks for this field, can set the bit prior to transmission to IMS.

12.2.6 APPCIOT= in DFSDCxxx for IMS/APPC application time-out

IMS Version 9 adds a second time-out value for APPC to the existing APPCIOT parameter in DFSDCxxx. This new value limits the amount of time an IMS application is inactive and can be used to terminate conversations that appear to be hung. The parameter can be described as follows:

```
APPCIOT=(xxxx,yyyy)
```

The first parameter of APPCIOT specifies the APPC time-out value expressed in minutes. This refers to an IMS wait for a requested APPC/MVS service to complete. IMS takes advantage of the ATBSTO5 service to enforce the time-out. Valid values are between 0 and 1440. If APPCIOT is specified as 0, time-out detection is deactivated.

The second parameter is new. This is the APPC/IMS time-out. This refers to an APPC/MVS wait for an IMS process to complete. For example, the timer would begin when IMS receives a message from APPC/MVS and would be reset when another APPC verb is issued, for example, send of a reply, error message or deallocate. This would be similar to putting a time-out value on a queue to queue response. Like the first parameter, valid values are between 0 and 1440. A specification of 0 causes deactivation of the time-out detection capability.

12.2.7 LU type 3 logon option as ETO SLUTYPE1 or 3270P

ETO now allows devices that are defined to VTAM as SLU type 3 (SLU3) to logon to IMS. IMS is not adding a SLU3 device driver, rather, the connection to IMS will be established as either a SLU type 1 (SLU1) or a 3270 Printer (3270P). This action extends what has been used in a static environment for years. The capability eases migration towards the use of ETO for networks that have SLU3 devices.

Specification of how a SLU3 device should connect to IMS is done in the initialization exit routine (DFSINTXO). During IMS initialization, this exit provides a set of flags to define whether SLU3 devices should connect to IMS as either SLUTYPE1 or 3270P. Sample routine is located in IMS.ADFSSMPL. You can specify the flags as follows:

- Parmlist - input/output flag byte (+16)
  - X‘08'
    - 0: SLU3 not allowed for ETO SLU1 (default)
    - 1: SLU3 allowed to logon as SLU1
  - X’04’
    - 0: SLU3 not allowed for ETO 3270P (default)
    - 1: SLU3 allowed to logon as

The exit should specify one or the other. If, by mistake, both flags are set then this is treated as an error condition and SLU3 devices will not be allowed to logon.

DFSINTXP is a new mapping macro that maps the options for DFSINTX0. It is used by DFSINITX0 and DFSAINB0. IMS ensures that this is only done for ETO environments. Sample is located in IMS.ADFSMAC.

12.2.8 Notify CQS outage to terminal users

In previous releases, if the CQS address space abended, users who tried to connect to IMS would get disconnected immediately after the DFS3649A message was sent. The terminal user was not told what caused the problem.

IMS Version 9 has enhanced the DFS3649A sign-on message for two sets of users. For ETO terminals, the user is signed off and the DFS3649A message includes RC=436. For static non-STSN terminals, the session persists, the user is signed off if already signed on, and RC=436 is included in the DFS3649A message. Other terminal users (static STSN terminals) are not affected by this change. The documentation of RC=436 indicates that CQS is unavailable. You can modify their greeting messages exit routine (DFSGMSG0) to react to the RC=436.

12.2.9 Greater than 255 transaction classes

IMS Version 9 lifts the restriction on the number of transaction scheduling classes that can be specified, from 255 to 999. A maximum specification of 999 was chosen so that command output and output messages in general would not adversely affect existing automation capabilities. It could be affected by a value greater than 3 digits. The following areas are affected:

- System definition
- Dependent region initialization
- Transaction scheduling
- Command processing
- Log records
- IMS control blocks

System definition

In the system definition, the APPLCTN, TRANSACT and IMSCTRL macros have been changed to allow a maximum transaction scheduling class specification of 999. You can specify as follows:

- APPLCTN macro, PGMTYPE= (,,999)
- TRANSACT macro, MSGTYPE= (,,999)
- IMSCTRL macro, MAXCLAS= 999

Dependent region initialization

It is also true for the CL1 through CL4 parameters in the DFSMPR and DFSJMP procedures.

Transaction scheduling

Since CPI-C driven transactions are not defined to IMS, the class specification is provided in the associated APPC/MVS TP_Profile entry. The value can now also be specified as 999. An example of this specification in MVS System Data Manager Utility (ATBSDFMU) is shown in Example 12-5. For more information about TP_Profile and ATBSDFMU, see *IMS Version 9: Administration Guide: Transaction Manager,* SC18-7808.

Example 12-5 The specification of TP_Profile in MVS System Data Manager Utility

```
TPADD TPSCHED_EXIT(DFSTPPE0)
      TPNAME(INQUIRY_PART)
      SYSTEM
      ACTIVE(YES)
      TPSCHED_DELIMITER(##)
            TRANCODE=PART
            CLASS=999
```

```
MAXRGN=1
CPUTIME=0
##
```

Command processing

The commands that are affected by the new maximum value are:

- /ASSIGN CLASS
- /ASSIGN TRANSACTION
- /DISPLAY ACTIVE (REGION)
- /DISPLAY STATUS CLASS
- /DISPLAY Q (CLASS)
- /DISPLAY TRANSACTION
- /PSTOP TRANSACTION CLASS
- /PURGE TRANSACTION CLASS
- /START CLASS
- /STOP CLASS
- /START TRANSACTION CLASS
- /STOP TRANSACTION CLASS
- QUERY TRAN
- UPDATE TRAN

The /ASSIGN command can be used to specify a new transaction scheduling class value. The /DISPLAY commands display the current transaction scheduling class value. The /PSTOP, /PURGE, QUERY, /START, /STOP, and UPDATE commands can also specify a transaction scheduling class value.

Log records

Two of the log records associated with scheduling have changed. This may affect user programs that are looking at these specific records.

- X'08' log record (Scheduling)
  - LINTSY2 changed to contain new 8-byte PSTCLASS field
  - LINTCLAS (new field) used available storage, class value was previously stored at LINTSY2+4
  - LINTPRTY (new field) used available storage, priority value was previously stored at LINTSY2+5
- X'0A08' log record (CPIC program Scheduling) changed the size of L0ACLASS field to 8 bytes

IMS control blocks

Regarding 999 transaction classes support, 4 IMS control blocks are changed.

SMB (scheduler message block) - IAPS mapping macro

SMBCLASS is moved (offset changed) to reserved field after SMBGUCNT, and changed size to half word.

The size is increased by adding 2 new reserved words before suffix:

- Old length of SMB = X'80' (EQU SMBLGTH)
- Old length of SMB plus suffix = X'88' (EQU SMBTOTSZ)
- New length of SMB = X'88' (EQU SMBLGTH)
- New length of SMB plus suffix = X'90' (EQU SMBTOTSZ)

And, offset changed for field SMBHELM in the suffix.

PST (partition specification table) - IPST mapping macro

PSTCLASS is moved (offset changed) to reserved fields at end of PST, and changed size to double word. No size increases.

TAB (transaction anchor block) - DFSTAB mapping macro

TCTCLAS is moved (offset changed) to reserved field within TCT block, and changed size to half word.

TABPSTQE field is deleted, and made as a reserved field.

The size is increased:

- Old size = X'B7F0' (EQU TABLGTH)
- New size = X'2CEB8' (EQU TABLGTH)

TABCLASS_JAVA_EYECATCHER field is added. Offset of field TABCLASS_JAVA and all TCTs are changed. And TCTN EQU is changed to 999 from 255.

SSOB (subsystem options block) - DFSSSOB mapping macro

SSCTNBA and SSCTOBA are moved to prior location of SSCTCLAS at offset X’58’.

SSCTCLAS is moved (offset changed) to reserved word at offset X’74’. And increased size to 8-bytes, that other 4-bytes are used at prior location of SSCTNBA and SSCTOBA starting at offset X'78'. No size increases.

12.3 Miscellaneous system manageability enhancements

In this section we describe the following system manageability enhancements:

- 12.3.1, “Operational enhancement for FDBR message DFS4172” on page 211
- 12.3.2, “FDBR virtual storage constraint relief” on page 212
- 12.3.3, “Independent ARM restart for FDBR” on page 212
- 12.3.4, “External subsystem resolve-in-doubt notification” on page 213
- 12.3.5, “Virtual storage constraint relief (VSCR)” on page 215
- 12.3.6, “Command recognition character (CRC) enhancement” on page 215
- 12.3.7, “/DIAGNOSE command” on page 217
- 12.3.8, “IPCS enhancement” on page 219
- 12.3.9, “Online change copy utility enhancements” on page 220
- 12.3.10, “Disabling of z/OS DFSMS V1R5 PS EDI” on page 221

12.3.1 Operational enhancement for FDBR message DFS4172

An operational usability enhancement changes the DFS4172 WTOR message to a WTO message so that a reply is not necessary. This message notifies operations when the FDBR tracker is started before the active IMS

Message DFS4172 is issued in the event that FDBR is started but IMS has not yet been started or has not reached the point of joining the FDBR XCF group. Prior to IMS Version 9 this message was issued as a WTOR meaning that action must be taken to reply to this message once the active IMS subsystem has been initialized.

In IMS Version 9 this message is now just a WTO to inform the operator that there is no active IMS to track. Once the active IMS has been started it will join the XCF group already established by FDBR. This will cause FDBR to be notified that the active IMS is up and it may now start tracking. Operator intervention to allow FDBR to begin tracking is no longer required.

The FDBR system waits for the cross-system coupling facility (XCF) group exit routine to indicate a change in status of a member of the group. That is, the FDBR system waits for the active IMS system to rejoin the group. Use the IRLM MODIFY command (F irlmproc,STATUS) to display the status of the identified IMS system. To ensure that the IMS system is active, check that the IMSID specified in the IMSID parameter of the EXEC statement is identical to the IMSID of the active system. Correct any errors and restart, or let ARM restart, the IMS active system. If the IMS system is XRF capable, RSENAME is used for the identifier instead of IMSID, which causes this message and an abend.

12.3.2 FDBR virtual storage constraint relief

For IMS Version 9, IMS can obtain storage for DEDB buffer pools from the IMS private region rather than from ECSA. For the ACTIVE IMSID control statement of the DFSFDRxx member of IMS.PROCLIB, you can specify an optional parameter *FPBUFF=LOCAL* that specifies that the control blocks for Fast Path DEDB processing should be obtained from the FDBR control region private storage instead of from ECSA. This includes the DEDB common buffer pool as well as the SVSO buffer pools.

FDBR in previous releases of IMS built Fast Path module DBFCONT0 in ECSA. DBFCONT0 includes the Fast Path data base buffer pool, SVSO buffer pools, DMACs, DMHRs, OTHRs, and other control blocks. FPBUFF=LOCAL statement in the DFSFDRxx member causes Fast Path to place DBFCONT0 in extended private. Since FDBR typically runs in an LPAR with other IMS subsystems or FDBR instances, this option may allow users to build configurations that ECSA limitations would otherwise prevent. This can mean significant ECSA relief for systems with large Fast Path buffer pools since FDBR must have the same size pool as the active IMS it is tracking. It may also prove very valuable where multiple FDBR subsystems are started on the same LPAR.

If the FPBUFF=LOCAL statement is not included in the DFSFDRxx member, the Fast Path DBFCONT0 module is built in ECSA as with previous releases of IMS.

This enhancement is also available for IMS Version 8 with APAR PQ80264.

Other VSCR for FDBR

Included in IMS Version 9 but also made available in Version 8 (PQ66531) and Version 7 (PQ66507) is additional VSCR for other pools previously obtained in ECSA. The following pools have been moved to EPVT:

- PSBW - The PSB work pool
- CSAPSB - The CSA portion of the PSB pool
- DMBP - The data management block pool
- DBWP - The database work pool

Note that this is a change for FDBR only; it has been true for IMS online systems previously.

12.3.3 Independent ARM restart for FDBR

With IMS Version 9, FDBR allows the active IMS and FDBR to be totally independent with respect to the Automatic Restart Manager. Prior to this change either IMS or FDBR could use ARM restart. If ARMRST was specified for FDBR it would disable ARMRST for the active IMS by using an associate function to prevent IMS from possibly restarting before FDBR could finish its processing causing an integrity exposure. Unfortunately this did not eliminate this possibility since IMS could be restarted by automation.

The code to close this potential exposure also allowed for the independent ARM restart possibility. Now if you specify ARM restart for FDBR the associate will no longer be done and ARM restart will work for both FDBR and IMS if specified at startup.

This change is also available in IMS Version 8 with APAR PQ88094, and in IMS Version 7 with APAR PQ86024.

12.3.4 External subsystem resolve-in-doubt notification

A new external subsystem attach facility (ESAF) indoubt notification exit routine (DFSFIDN0), is provided in IMS Version 9 that allows customers to resolve in-doubt work without waiting for IMS restart to complete. In-doubt work is work that has completed sync point phase 1 (prepare) but has not completed sync point phase 2. This enhancement provides the identity of in-doubt IMS ESAF UOWs to a user notification exit during the recovery phase, thus enabling you to resolve the in-doubt UOW between IMS and external subsystems before IMS and the external subsystem have been reconnected.

The new exit is called for each unresolved unit of work (UOW) during `/ERE`, XRF takeover and FDBR processing. The action of the exit routine is determined by the user. IMS does not take any action to resolve unresolved units of work. This is the responsibility of the user. The user exit is not called during `/ERE COLDBASE` or if RRS is the sync point coordinator. IMS retains information necessary to do resolve-in-doubt processing. When IMS again connects to the subsystem, it will do resolve-in-doubt processing for all work that was unresolved when IMS last terminated, regardless of any action taken by the user exit

This capability is especially important for IMS and DB2 data sharing environments using FDBR. When an IMS system fails, FDBR backs out its uncommitted IMS database updates and releases its IMS locks. It does not resolve any in-doubt UOWs with DB2. Normally, this has been done after IMS is restarted and reconnects to DB2. The new exit routine capability allows users to resolve these in-doubts much more quickly.

The FDBR region is a separate address space that monitors an IMS, detects failure, and recovers any IMS-owned database resources that are locked by the failed IMS, making them available for other IMS systems without having to wait for the next full restart. As they scan their collections of IMS units of work for recovery, FDBR and IMS systems determine whether external-subsystem data was also changed as part of each unit of work. If so, they provide all of the following information to a user exit routine:

- Recovery token
- The name of the external subsystem that was connected to the failed IMS
- The final resolution of the data

The exit parameter list is mapped by DFSRNID macro in IMS.SDFSMAC. This exit is optional and no sample code is provided by IBM. The name of the exit routine must be DFSFIDN0. The FDBR or IMS system attempts to load the user exit routine from the concatenation of the IMS.SDFSRESL libraries. IMS calls the exit routine once for each in doubt ESAF unit of work. This is a synchronous call to the exit routine, so you should ensure that the exit routine does not impact the performance of the FDBR or the IMS system. For more details about ESAF indoubt notification exit routine (DFSFIDN0), refer to *IMS Version 9: Customization Guide,* SC18-7817.

DB2 has a facility which provides services somewhat similar to FDBR. However, it is implemented differently. The facility is DB2 Restart Light. Unlike FDBR, DB2 Restart Light does not track another subsystem. Instead, it is a special form of DB2 restart. As the name implies, it uses less resources, including less ECSA, than a normal DB2 restart. It backs out in-flight DB2 updates and releases the associated locks. In DB2 Version 7 it terminated, after this processing. It did not address in-doubt updates. An enhancement in DB2 Version 8 addresses this. DB2 Version 8 Restart Light remains active if there are any in-doubts.

These can be resolved by restarting the partner IMS on the same LPAR. Alternatively, the new IMS exit routine may be used with FDBR or IMS emergency restart. The exit routine may be used to identify whether the in-doubt UOWs should be committed or aborted. This information may be used to issue the -RECOVER INDOUBT command on the DB2 Restart Light instance.

Figure 12-2 is an illustration of a possible interaction of IMS FDBR and DB2 with Restart Light.

At the top of Figure 12-2, there are two systems in the Parallel Sysplex. Both systems have an active IMS and an active DB2. Both IMS systems have a tracking FDBR on the other system.

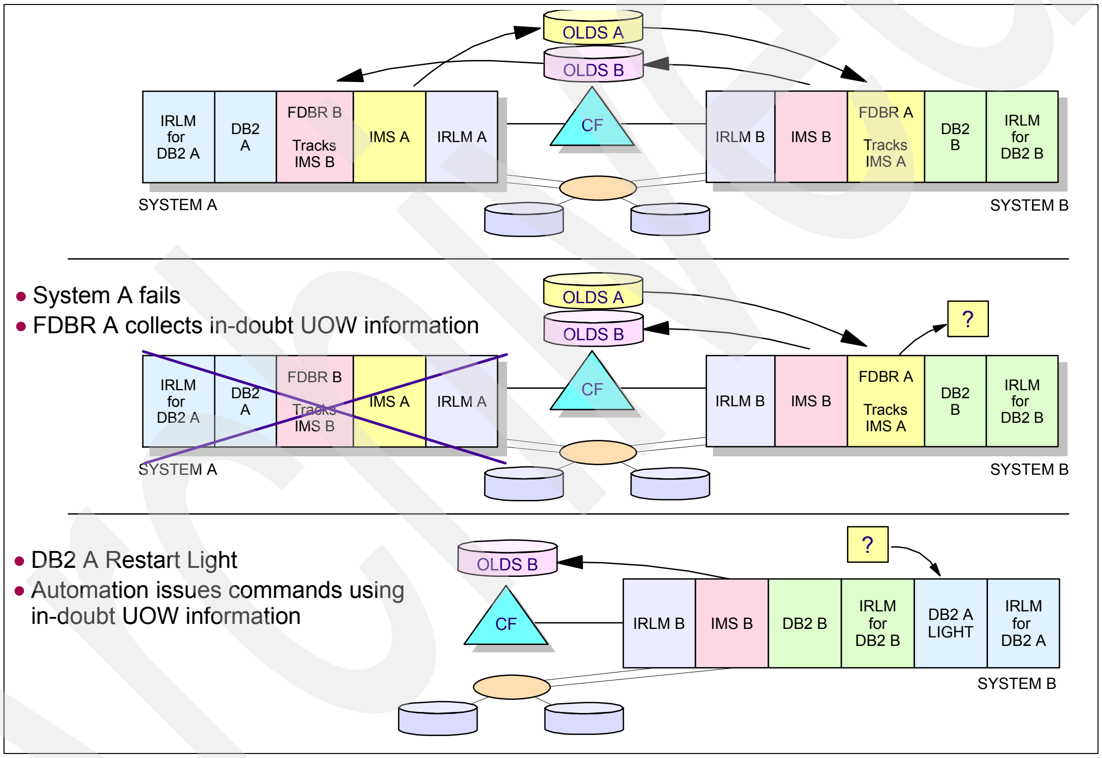


Figure 12-2 FDBR DB2 in-doubt UOW

In the middle of Figure 12-2, system A fails. This invokes FDBR A which was tracking IMS A. FDBR A releases IMS locks held by IMS A. This makes this data available to IMS A. FDBR A completes before DB2 Restart Light for DB2 A can be started. FDBR A externalizes information about the in-doubt DB2 UOWs. This could be writing records to a file, sending messages via MQ, or some other method. One should be careful that the release of IMS locks by FDBR is not delayed. The exit routine is called for each unresolved UOW before any back outs are done or any locks are released.

At the bottom of Figure 12-2, DB2 Restart Light is invoked for DB2 A. The -RECOVER INDOUBT commands are issued using the saved information from the FDBR exit routine. This releases the DB2 locks and makes this data available to DB2 B.

12.3.5 Virtual storage constraint relief (VSCR)

The VSCR enhancement ensures that the data set association blocks (DSABs) generated are allocated in above-the-line private storage, thereby providing enhanced virtual storage constraint relief. For example, if you have 20,000 full function and/or Fast Path data sets dynamically allocated, you could have over one megabyte of storage allocated above the line. Approximately 20 online change modules moved from below-the-line private to above-the-line private. Also the online change modules are removed from the nucleus and become standalone modules.

IMS systems moving from IMS Version 7 to IMS Version 9 will see a savings in CSA of 48K or 112K depending on whether the environment includes Fast Path. An additional amount of savings can be calculated based on the number of log buffers and whether single or dual logging is used. This is further documented in the redbook *IMS Version 8 Implementation Guide - A Technical Introduction of the New Features,* SG24-6594, which can be downloaded from the following URL:

```
http://www.ibm.com/redbooks/
```

IMS systems moving from IMS Version 8 to IMS Version 9 will see a decrease in CSA usage below the line. This is due primarily to several modules being moved from 24-bit to 31-bit addressing.

- CSA to ECSA

  Load modules moved from 24-bit to 31-bit addressing
- PVT to EPVT

  Control blocks associated with dynamically allocated data sets
- ECSA to EPVT

  DEDB storage in an FDBR environment

The size of the control blocks associated with dynamically allocated databases is 96 bytes per dataset. For full function datasets the storage is moved from DLISAS PVT to EPVT, and for Fast Path, from IMS control region PVT to EPVT. Databases allocated via JCL maintain their DSABs in the PVT storage of the appropriate address space.

12.3.6 Command recognition character (CRC) enhancement

If all subsystems executing in a sysplex have a unique command recognition character, then an operator can enter a command from any system in a sysplex, and route that command to the appropriate subsystem for execution. For this to be operational, all subsystems must register their characters. The z/OS group requested that all subsystems cooperate in this endeavor.

During control region initialization, after all input parameters have been processed, the z/OS command prefix facility (CPF) macro is issued to register the IMS command recognition character (or command prefix) within the system.

The intention is to have all the IMSs on a system register a unique CRC. This gives the console operator the capability to issue a command from any system in the sysplex and direct it to the appropriate system for execution. If the character is not unique, IMS issues the following message but continues on through its initialization process:

```
DFS1946W IMS CRC OF 'x' ALREADY EXISTS WITHIN SYSTEM OR SYSPLEX
```

However, note that IMS checks this only to see if it is unique within a system, not across the sysplex, in spite of what the text of the DFS1946W might imply and the explanation in the *IMS*

Version 9: Messages and Codes Volume 2, GC18-7828 explicitly states. This is done intentionally since you may want to have the same CRC on different LPARs in the same sysplex, because it allows a single command to be processed by all the subsystems having the same CRC.

A DISPLAY OPDATA -command on the z/OS console shows you all recognition commands used on your environment, refer to Example 12-6.

Example 12-6 z/OS DISPLAY OPDATA command and its' response

```
SDSF OPERLOG  DATE 08/02/2004    14 WTORS               66 RES
COMMAND INPUT ===> /D OPDATA                                 S
RESPONSE=SC42
 IEE603I 13.58.03 OPDATA DISPLAY 861
  PREFIX     OWNER      SYSTEM     SCOPE     REMOVE   FAILDSP
  $          JES2       SC47       SYSTEM    NO       SYSPURGE
  -MQFI      MQFI
-MQFN      MQFN
-MQJ1      MQJ1
-MQJ2      MQJ2
-MQ4B      MQ4B
-MQ4D      MQ4D
-DBD8      DBD8MSTR   SC54       SYSPLEX   NO       SYSPURGE
-DB4B      DB4BMSTR   SC48       SYSPLEX   NO       SYSPURGE
-DB4C      DB4CMSTR   SC50       SYSPLEX   NO       SYSPURGE
-DB4D      DB4DMSTR   SC52       SYSPLEX   NO       SYSPURGE
-DB7A      DB7AMSTR   SC53       SYSPLEX   NO       SYSPURGE
-DB7C      DB7CMSTR   SC47       SYSPLEX   NO       SYSPURGE
/          IMSG       SC53       SYSTEM    NO       PURGE
?          IM2B       SC47       SYSTEM    NO       PURGE
?          IM1B       SC42       SYSTEM    NO       PURGE
```

The CRC can be used on the SDFS screen for example /?DIS A which returns the status of the IM1B if you are on SC42 console without using the reply number. In our case, if we had used ROUTE *ALL with the command, we would have gotten the response also from IM2B system on SC47.

Example 12-7 /?DIS A example

```
SDSF OPERLOG  DATE 08/02/2004    14 WTORS              COMMAND ISSUED
COMMAND INPUT ===> /?DIS A                                   SCROLL ===> CSR
RESPONSE=SC42
 DFS4444I DISPLAY FROM ID=IM1B
      REGID JOBNAME   TYPE  TRAN/STEP PROGRAM  STATUS           CLASS
          2 MPP1BSL   TP    TPCCS     PSBSL    WAIT-INPUT         5
          4 MPP1BDE   TP    TPCCD     PSBDE    WAIT-INPUT         4
          3 MPP1BOS   TP                       WAITING            3
            JMPRGN    JMP   NONE
            JBPRGN    JBP   NONE
            BATCHREG  BMP   NONE
          1 IFP1BPA   FPM   NO MSG.   PSBPA
          5 IFP1BNO   FPM   NO MSG.   PSBNO
            DBTRGN    DBT   NONE
            IM1BDBRC  DBRC
            IM1BDLS   DLS
      VTAM ACB OPEN         -LOGONS ENABLED
```

12.3.7 /DIAGNOSE command

Customers want the ability to obtain IMS diagnostic information without impacting operations. Currently, a console dump is required to gather the basic information necessary to analyze many types of ordinary problems, such as hung terminals or transactions. Even the simple verification of configuration, system generation, or startup information in key control blocks may require a dump. Taking a console dump in an IMS production environment, especially during peak hours, is a disruptive process, and many customers are forced to schedule dumps during an off-peak period or are prevented from taking dumps at all in their production environment.

This enhancement answers a long-standing requirement in the problem determination area. For prior releases, when the IBM support center, needed information to analyze a problem, oftentimes a console dump was requested. Unfortunately, taking a console dump in an IMS production environment, especially during peak hours, was considered a disruptive process that could potentially have a negative impact on operations, service level agreements, and financial goals.

The new **/DIAGNOSE** command in IMS Version 9 provides customers with a built-in, nondisruptive alternative to the console dump for gathering the required information. Additionally, the information produced is in a much more compact form than a console dump which will allow for easier transmission to the IBM support center, when necessary, and will allow IBM service to respond more quickly to problems.

The new **/DIAGNOSE** command operates like a standard type-1 command. An operational consideration is that the OLDS needs to be switched to extract logged data. The command output can also be directed to the external trace data se by using the OPTION(TRACE) keyword with the command. OPTION(OLDS) is the default and it directs the output to the online log data set.

IMS Version 9 supports the following control blocks: ALL, CMDE, ESCD, LSCD, MWA, QSCD, SCD, SQM, TSCD. **/DIAG** is a new type-1 command, and it logs IMS resource data to the OLDS as type X'6701' records. It uses existing log record mapping and facilities, and allows users to take a snap shot of IMS resources. The command itself is also logged as a X'02' record

This command can be issued from an IMS terminal, a console WTOR, APPC and OTMA clients, an AOI program, MCS/EMCS consoles, and any OM command clients including SPOC.

The SNAP function of the command captures storage, both addresses and raw data, for the requested IMS control blocks, NODEs and transactions. The information in the blocks is copied to a copy storage area to avoid holding enqueues, locks, latches, etc. The environment is further protected by a separate ESTAE that both protects the copy process and prevents an IMS failure. The information is written to the OLDS. A successful completion of the command returns a DFS058I message.

When requesting the SNAP function against the BLOCK resource, at least one control block name is required. The Example 12-8 lists all the valid names that can be requested. Specifying “ALL” results in capturing the information associated with one control block of each type mentioned in the list.

When requesting a SNAP of a specific transaction, the Scheduler Message Block (SMB) is captured. The SMB is the control block that contains information relative to a transaction.

Example 12-8 /Diagnose keywords

```
/DIAGNOSE          SNAP                BLOCK(     ALL           )
```

```
              ALL     Capture all control blocks
              CMDE    IMS Commands SCD Extension
              ESCD    Extended System Contents Directory
              LSCD    LU 6.2(APPC) SCD Extension
              MWA     Modify Command Work Area (between
                           /MOD prepare and /MOD commit)
              QSCD    Queue Manager SCD Extension
              SCD     System Contents Directory
              SQM     Shared Queue Master Control Block
              TSCD    OTMA SCD Extension
NODE(nodename)
TRAN (tranname)
LTERM(ltermname)
USER(username)
ADDRESS(address)
LENGTH(length) Key(0|7)
```

Since this is a new command, the implementation has some potential considerations. If command security is desired, then changes may need to be made to both the RACF definitions as well as the DFSCCMD0 command authorization user exit.

> **Note:** The command is invalid in an XRF alternate environment. It cannot be entered on the alternate nor is it processed by the alternate if the command is entered on the active system.

Command security is achieved by using on or more of the following:

- RACF CIMS definitions
  - Add DIAGNOSE command to RDEFINE for CIMS
- DFSCCMD0 command authorization user exit
- OM security user exit
- RACF OPERCMDS class for OM

It is supported on a RSR environment on RLT and DLT remote trackers.

In an IMSplex environment that uses the Resource Manager, the status recovery mode for USER and NODE resources can be defined as local, global or none. This is specified in the SRMDEF parameter of DFSDCxxx.

If specified as GLOBAL, the significant status of a resource is saved globally in the Coupling Facility resource structure every time the significant status changes, along with all other recoverable status for that resource. The resource status is restored at the next logon or sign on and is available from any IMS system in the IMSplex. The resource status is copied to the local system when that resource becomes active, but is deleted from the local system when it becomes inactive.

The use of the **/DIAGNOSE** command in this environment can therefore result in capturing the terminal control blocks from another system.

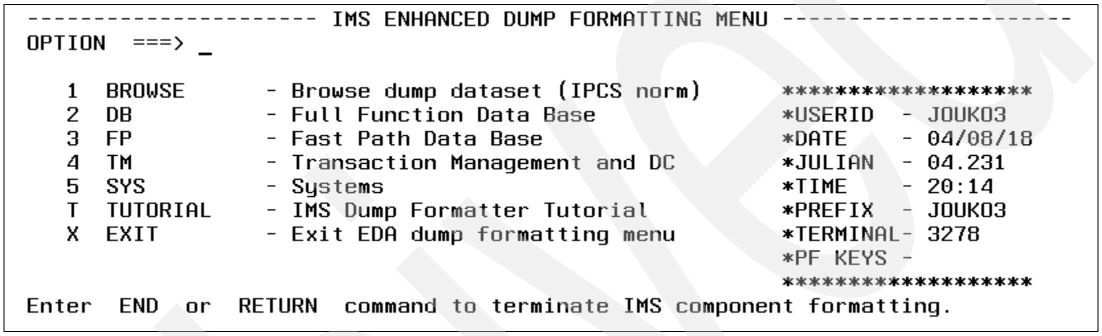
Use of the command is documented in the bullets below. Note that since the x'6701' log record is used with no change to the mapping of the record, the existing DFSERA10 batch job can be used to extract the information. The extracted information is what should be sent to the IBM support center when requested.

1. Issue command

2. Switch the OLDS
3. Extract the information using DFSERA10 (batch job)
4. Transmit to IBM support center:
   a. Cut and paste in PMR
   b. FAX
   c. E-mail

12.3.8 IPCS enhancement

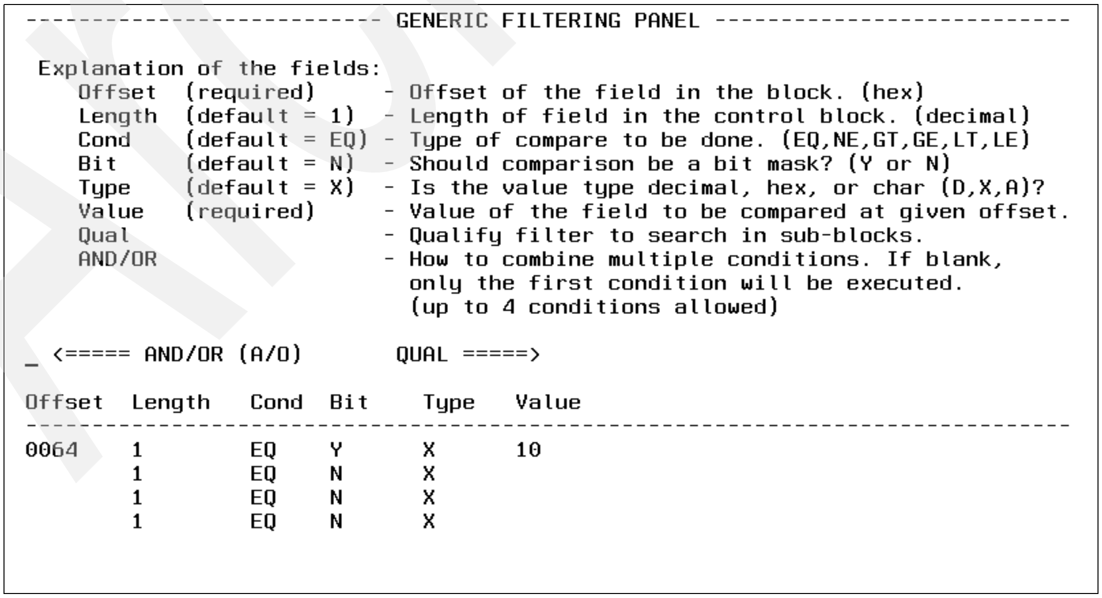
In IMS Version 9 the Interactive Problem Control System (IPCS) is enhanced with the Enhanced Dump Analysis (EDA). The EDA tool provides an alternative way to assist you in reducing the time spent on dump analysis.



```
---------------------- IMS ENHANCED DUMP FORMATTING MENU ----------------------
OPTION  ===> _

   1  BROWSE      - Browse dump dataset (IPCS norm)      ******************
   2  DB          - Full Function Data Base              *USERID  - JOUK03
   3  FP          - Fast Path Data Base                  *DATE    - 04/08/18
   4  TM          - Transaction Management and DC        *JULIAN  - 04.231
   5  SYS         - Systems                              *TIME    - 20:14
   T  TUTORIAL    - IMS Dump Formatter Tutorial          *PREFIX  - JOUK03
   X  EXIT        - Exit EDA dump formatting menu        *TERMINAL- 3278
                                                         *PF KEYS -
                                                         ******************
Enter  END  or  RETURN  command to terminate IMS component formatting.
```

Figure 12-3   IPCS EDA select panel

In the EDA primary menu you can select the different parts of IMS like DB, FP, TM, and SYS as shown in Figure 12-3. Each one shows you the possible sub selection list like PST for DB. This is not really new since we have been able to select PSTs from the previous IMS versions as well, but the filter option is new. If you select this option and press enter the panel as shown in Figure 12-4 appears.



```
-------------------------- GENERIC FILTERING PANEL --------------------------

 Explanation of the fields:
    Offset  (required)     - Offset of the field in the block. (hex)
    Length  (default = 1)  - Length of field in the control block. (decimal)
    Cond    (default = EQ) - Type of compare to be done. (EQ,NE,GT,GE,LT,LE)
    Bit     (default = N)  - Should comparison be a bit mask? (Y or N)
    Type    (default = X)  - Is the value type decimal, hex, or char (D,X,A)?
    Value   (required)     - Value of the field to be compared at given offset.
    Qual                   - Qualify filter to search in sub-blocks.
    AND/OR                 - How to combine multiple conditions. If blank,
                             only the first condition will be executed.
                             (up to 4 conditions allowed)

_ <===== AND/OR (A/O)       QUAL =====>

Offset  Length   Cond  Bit    Type   Value
-------------------------------------------------------------------------------
0064     1        EQ    Y      X      10
         1        EQ    N      X
         1        EQ    N      X
         1        EQ    N      X
```

Figure 12-4   Filter option

Place an S to the left of the desired low level formatting option(s) and fill in the associated argument. If you omit the argument for an option marked *P* you will be given a sub-list of selectable items for the option. Enter UP/DOWN to scroll or END to cancel the request.

Filtering provides a powerful method of finding only the blocks that you are interested in. Instead of manually searching through many blocks for the few blocks that you might be interested in, you can create a simple boolean expression (with the help of the filtering panel) to search for those blocks for you. Multiple boolean expressions are supported, as well as something called a qualifier. Qualifiers allow the user to filter with fields in a sub-block. For example, if block A has a sub-block B, you can use the qualifier to filter through fields in block B. The result will show all blocks A with the corresponding B blocks, that match the filter criteria for block B.

Let's start from the top and move our way down. The first two fields (AND/OR and QUAL) are global parameters. The AND/OR option is an optional parameter, but is required when defining more than one boolean statement. If this field is left blank, only the top boolean statement will be executed. Furthermore, in combining these statements, they are either all ANDed or all ORed. The QUAL field is the qualifier. For EPSTs, if the DMHR or XCRB was not selected from the previous panel, they may be typed in here.

Now to making the actual boolean statements. As shown above, the default values are already filled in for you. OFFSET is a required field that is the location in the block of the field you want to filter on. LENGTH is how long from that offset you want to compare values. COND is how you want to compare the value at the offset. BIT can be set to enable bit comparisons. TYPE is the type of the VALUE field. VALUE is a required field containing the value to compare to. Example: OFFSET=2F, LENGTH=1, COND=NE, BIT=N, TYPE=X, VALUE=00. This would mean you are searching for all blocks, that at offset 2F have a non-zero value.

12.3.9 Online change copy utility enhancements

The online change copy utility (DFSUOCU0) calls IEBCOPY to perform the copying of members between online change data sets. There have been times, especially when copying a large number of members, that the utility has failed because of inadequate default execution values.

The online change utility is enhanced to override IEBCOPY parameter values. This is a means to ensure that the utility runs to successful completion.

The IEBCOPY utility override parameters that can be specified in the online change utility include: WORK=, SIZE=, LIST=. These parameters are optional and positional and added as the 4th - 6th parameter in the OLCUTL procedure. If one or more are specified, the following combinations are valid:

- WORK=
- WORK=, SIZE=
- WORK=, SIZE=, LIST=

WORK=(n | nK | nM | 1M) specifies the number of bytes of virtual storage to request for a work area to hold directory entries, internal tables, and I/O buffers. The default request will be for 1M. The actual amount obtained will not exceed the space available in the REGION.

SIZE=(n | nK | nM) specifies the maximum number of bytes of virtual storage that IEBCOPY may use as a buffer. It is best to let IEBCOPY choose buffer sizes by not using this parameter. The minimum buffer size is approximately 4 times the largest track size of the devices being used, plus about 3%. There is no maximum for this value, but IEBCOPY cannot use more than the quantity available in the work area.

LIST=(NO|YES) is optional parameter that passes the list parameter to the IEBCOPY utility. LIST=NO suppresses IEBCOPY IEB154I messages that are issued for each member that is successfully copied. The default value is YES. This parameter can only be specified if the work and size parameters are also specified.

The IEBCOPY options list may not exceed 64 bytes. This count includes: the IEBCOPY keywords, equal signs, parameter values, and commas. If exceeded, the following error message is issued:

```
DFS3486 IEBCOPY PARAMETER ERROR
```

This error is occurred, for example, when length of the IEBCOPY options list exceeds 64 bytes, or one or more IEBCOPY parameters are specified on the PROC and EXEC statements but not specified as input parameters. The IEBCOPY parameters are passed to IEBCOPY when it is called.

12.3.10 Disabling of z/OS DFSMS V1R5 PS EDI

Physical Sequential Enhanced Data Integrity (PS EDI) is a data integrity function which is introduced newly in z/OS DFSMS V1R5. PS EDI prevents two programs from writing to the same physical sequential data set at the same time. It does this by preventing two programs from opening the data set concurrently if the data set is DSORG=PS, allocated with DISP=SHR, and opened for output or update. This check is for all programs in the same LPAR and in different LPARs in the same sysplex.

PS EDI is optional in z/OS DFSMS V1R5. PS EDI is invoked by the IFGPSEDI member of SYS1.PARMLIB. An installation may set it to a warning level (WARN mode). With this level, opens are not prevented but warning messages are issued. An installation may set it to an enforcement level (ENFORCE mode). With this level, opens are prevented.

IMS OSAM data sets are DSORG=PS, but IMS does not need this protection for database data sets, because DBRC authorization processing prevents concurrent up daters of these data sets unless data sharing is being used with IRLMs. If data sharing is used, IRLM lock processing provides the required integrity.

In addition to OSAM data sets, there are other data sets that must be opened for update by multiple IMS systems. When XRF is used both the active and the alternate must have the OLDS, WADS, RDS, and MSDB dump data sets open for update concurrently. When takeover occurs, the alternate must have write capabilities for these data sets. So, IMS also disables PS EDI for these data sets when XRF is used.

DFSMS allows authorized programs running in keys 0-7 to disable PS EDI. IMS disables PS EDI for OSAM database data sets and the mentioned system data sets, by setting a bit in the DCBE (DCBEEXPS) for the data set. The APARs supplying this function are PQ83940 for IMS Version 9, PQ83941 for IMS Version 8, and PQ83942 for IMS Version 7. There is no user specification. PS EDI is invoked and is an excellent facility for other IMS data sets, such as logs.

13

Migration and coexistence considerations

This chapter provides information about migration and coexistence steps you need to consider if migrating from a supported IMS release to IMS Version 9. Migration tasks such as applying the RECON coexistence code to the previous version is discussed.

- Supported migration paths
- Supported connections and required maintenance
- Migration tasks and coexistence considerations
- RECON upgrade

13.1 Supported migration paths

Minimum software level prerequisites for z/OS are z/OS V1R4 (5694-A01) with the IBM High-Level Assembler Toolkit (program number 5696-234).

If you use RACF for security, z/OS V1.6 is needed for new RACF security classes IIMS, JIMS, LIMS, MIMS as default classes. These classes can be defined manually with z/OS V1.4 and higher. See 6.2, "Resource access security (RAS)" on page 86 describing the new security classes.

When IMS recognizes it is running on a z/OS 1.5 or later system, it will ensure that DSAB blocks created as part of the allocations for FF or FP databases, will be above the line.

13.2 Supported connections and required maintenance

Here we provide the list of the minimum requirements for coexisting systems. Check the PSP bucket for the latest maintenance information.

- ISC is supported with
  - IMS Version 9, Version 8 and Version 7
  - CICS Transaction Server for z/OS Version 2 (program number 5697-E93)
  - CICS Transaction Server for OS/390 Version 1 (program number 5655-147)
  - CICS/ESA®(R) Version 4 (program number 5655-018)
  - User-written software
- DB2 connections are supported with
  - DB2 Universal Database for OS/390 Version 6 (program number 5645-DB2)
  - DB2 for z/OS and OS/390 Version 7 (program number 5675-DB2)
  - DB2 UDB for z/OS Version 8 (program number 5625-DB2)
- DBCTL connections are supported with
  - CICS Transaction Server for z/OS Version 2 (program number 5697-E93)
  - CICS Transaction Server for OS/390 Version 1 (program number 5655-147)
  - CICS/ESA Version 4 (program number 5655-018)
- MSC is supported with
  - IMS Version 8
  - IMS Version 7 (PQ32932, PQ58631)
- Shared queues is supported with
  - IMS Version 8
  - IMS Version 7 (PQ32932, PQ54585, PQ58631)

  Compatibility APARs are required for IMS Version 7 users during the migration to IMS Version 9. These are required for MSC support between the two versions, or the use of shared message queues, or the Queue Control Facility.
- Control Center support for IMS Version 9
  - DB2 Universal Database Version 8, with fix pack 5 or later
  - IMS Connect Version 1.2 or later, you must have APARs PQ62379 and PQ70216 installed.
  - IMS Version 9 provides an integrated IMS Connect function; make sure that the Connect function is installed and configured on your system.
- IMS Java RDS

  - Client
    - Distributed WebSphere Application Server V5.0.2
  - z/OS
    - IMS Version 9
    - z/OS V1.4 or later
    - WebSphere Application Server for z/OS Version 5.0.2.3 or later. If you have WebSphere Application Server for z/OS V5.0.2, you must install either V5.0.2.1 or apply APAR PQ81944.
    - WebSphere Application Server V5.0.2.2 with cumulative fixes that include PQ79485, or WebSphere Application Server V5.0.2.3 or later.
    - • IBM SDK for z/OS Java 2 Technology Edition V.3 IBM SDK for z/OS Java 2 Technology Edition, Version 1.3.1 or later
  - The following protocols and z/OS components are required:
    - RRS (resource recovery services) for z/OS
    - RACF or equivalent product
  - The following IMS components are required:
    - ODBA
    - DRA

  To use IMS Java remote database services to access IMS databases from applications that run on WebSphere Application Server on a non-z/OS platform, you must download IMS Java files from the IMS Java Web site. These files are required in addition to the files that are installed as part of the SMP/E installation of the IMS Java FMID.

  To download the required IMS Java files, go to the IMS Web site at:

  http://www.ibm.com/ims

  and link to the IMS Java Web page for more information. The download page contains the required files for remote database services as well as the installation verification program and two sample applications for remote database services.

- JDBC access to IMS DB
  - DB2 stored procedures
    - Minimum of DB2 for z/OS and OS/390 Version 7 (APARs PQ46673 and PQ50443)
    - DB2 SQLJ/JDBC driver with APAR PQ48383
  - CICS applications
    - Minimum of CICS Transaction Server for z/OS Version 2
  - WebSphere applications
    - WebSphere Application Server (WAS) z/OS V4.0.1, and additional WAS z/OS Connection Management support
- DB2 access from an IMS JMP or JBP region
  - Part of base IMS Version 9
  - IMS Version 8
    - PQ73326 (UQ80615)
    - Prerequisite PQ73897 (latest level of IMS Java)
    - Prerequisite PQ75284 (IMS code using DB2’s attachment facility)
  - DB2
    - DB2 Version 8 (PQ74629)

    - DB2 Version 7 (PQ69861)
  - Uses DB2 RRS attach facility
    - Requires SSM= and RRS=Y in the IMS startup definitions
- IMS Java and the DLIModel utility require Xalan-Java Version 2.6.0 or later from the Apache Software Foundation:

  http://www.apache.org

  XML Project, or equivalent code function. XSLT 4.3.1, which is equivalent to Xalan-Java Version 2.6.1, is included in SDK 1.4.2. If you are using SDK 1.4.2 or later, do not download and install the Apache XML files.

  The Apache XML Project is a collaborative software development project that licenses open source software at no charge. The following open source files (or equivalent code function) are required by IMS Java and the DLIModel utility: xercesImpl.jar, xalan.jar, xml-apis.jar.
- Integrated IMS Connect function

  The IMS Connector for Java 9.1.0.1 runtime component for z/OS is delivered with the base of IMS. The IMS Connector for Java 9.1.0.1 development component which ships with WebSphere Studio Application Developer Integration Edition (WSADIE) will be delivered as an update to WSADIE 5.1.1.
- IMS Version 9 adds the XPLINK= dependent region parameter for Java Message Processing (JMP) regions and Java Batch Processing (JBP) regions. This parameter allows IMS Java dependent regions to use the IBM Language Environment® Extra Performance Linkage (XPLINK) runtime option. If you use IBM SDK for z/OS, Java Technology Edition, Version 1.4.1 or later, you must specify XPLINK=Y for a JMP region or a JBP region. However, if you use Version 1.3.1 of the SDK, you should specify XPLINK=N or allow IMS to use XPLINK=N as the default.

Related reading

Specific information about product enhancements that were made for earlier releases of IMS might not be included in this book. When migrating to a more recent release of IMS, be sure to review the release planning guides that span your specific migration path. The release guides that are currently available are:

- *IMS Version 8: Release Planning Guide,* GC27-1305
- *IMS Version 7: Release Planning Guide,* GC26-9437
- *IMS/ESA Version 6: Release Planning Guide,* GC26-8744

13.3 Migration tasks and coexistence considerations

The following is a high level list of migration tasks. It is not supposed to be complete, rather vice versa, this list is an absolute minimum for all the installations. You should prepare your own migration and fallback plan, which should have a detailed list of all the tasks appropriate to your environment. The manual, *IMS Version 9: Release Planning Guide,* GC17-7831 provides a more detailed migration check list, and can be of help when compiling your own list. On very high level, you need to perform the following tasks:

- Check PSP bucket for latest maintenance information
- Apply coexistence maintenance to other IMS systems
- Install IMS Version 9
- Check the required software levels of IMS related products such as IMS Tools or OEM vendor products; upgrade to the appropriate levels and maintenance

- Check and test any changes needed to the user exits and usermods
- Upgrade RECONs
- Do system definition
- Do ACBGEN

13.3.1 CQS coexistence considerations

The following coexistence considerations exist for CQS:

- An IMS Version 9 CQS supports only IMS Version 9. Earlier versions of IMS cannot register with an IMS Version 9 CQS.
- User or vendor-written CQS clients that want to register with an IMS Version 9 CQS must be recompiled with the IMS Version 9 CQS macros.
- IMS Version 7 system can register only with IMS Version 7 CQS.
- IMS Version 7 CQS, IMS Version 8 CQS, and IMS Version 9 CQS can connect to the same Coupling Facility structure.
- IMS Version 7 CQS, IMS Version 8 CQS, and IMS Version 9 CQS can run on the same central processing complex (CPC).

13.3.2 Database utilities coexistence

The database recovery utilities take three types of input; log, image copy, and change accumulation data sets. The method of processing for coexistence in the utility programs is to translate the records as they are read into the format of the current release for that record type. For the pre-version 8 change accumulation records, there is not sufficient information to translate the records in IMS Version 8 or IMS Version 9. For this reason, it is necessary to apply the following coexistence APAR to the utilities in IMS Versions 7 to provide the necessary information to enable the translation:

- APAR PQ54585 for IMS Version 7

Only those CA records from IMS Version 7 that were created after installing the appropriate APAR are usable by IMS Version 8 or IMS Version 9. The installation of IMS Version 9 as an upgrade from IMS Version 7 requires a sequence of operations to achieve coexistence for the recovery and change accumulation utilities:

1. The coexistence APAR must be installed on the existing IMS Version 7 system.
2. The change accumulation files must be conditioned to allow them to be used by IMS Version 9. This is done in one of this ways:
   – Make an image copy of the database subsequent to the installation of the APAR. This will invalidate all of the old CA data sets.
   – Run the change accumulation utility on IMS Version 7 with the appropriate APAR installed. If the APAR is installed before IMS Version 9 is used, most or all of the CA data sets might be processed and made valid for coexistence with IMS Version 9.

The batch back out, log archive, and log recovery utilities read only one input. This is a log data set. The IMS Version 9 version of these utilities accepts only input produced by IMS Version 9. Logs produced by earlier releases must be processed by the utility from the same release.

The IMS Version 9 database recovery utility (DFSURDB0) accepts all types of image copies, including batch image copy, image copy with the CIC option, online image copy, and partial image copies produced by HSSP. A HISAM unload data set may be used in place of an image copy. These data sets, change accumulation data sets, and logs (RLDS and SLDS) may come from prior releases.

The IMS Version 9 change accumulation utility accepts logs and change accumulation input data sets produced by prior releases.

IMS Database Recovery Facility (DRF) V2R1 support requires the following APARs:

- PQ80052 to support recovery of HALDB database data sets which have undergone online reorganization
- PQ77559 to allow specification of sort parm overrides in SYSIN
- PQ79162 to enable secondary source support
- PQ80051 for IMS Version 9 coexistence and support for IMS Version 9 input
- PQ79326 to add the image copy cached function for the Virtual Tape Server (VTS)
- PQ79327 to support large block image copies

13.3.3 Remote Site Recovery (RSR) coexistence

For coordinated disaster recovery for IMS and DB2, an IMS Version 9 RSR tracking subsystem with XRC tracking enabled can track logs from supported pre-Version 9 active systems that have the usual and customary DBRC coexistence SPEs applied. For regular RSR, IMS Version 9 RSR coexistence supports configurations in which:

- The tracking site Transport Manager Subsystem (TMS) is at the IMS Version 9 level and the active site is at IMS Version 6, Version 7, or Version 8 level.
- The active site RECON data sets are at the IMS Version 6, Version 7, or Version 8 level (with the Version 9 DBRC coexistence SPE applied)
- The active TMS that is running ILS is at the IMS Version 6, Version 7, or Version 8 level.
- One or more active IMSs are at the IMS Version 6, Version 7, or Version 8 level.

Table 13-1 on page 229 contains all the supported configurations of a simple set of subsystems in an RSR Global Service Group (GSG). Some unsupported configurations are not included because a prerequisite configuration also is not supported. Some unsupported configurations are included even though a pre-requisite configuration is supported so that the reason for non-support can be explained.

The simple set of configurations in Table 7 on page 49 assumes:

- A single active IMS
- A single Transport Manager Subsystem (TMS) that is running the Isolated Log Transport
- A single set of active site RECON data sets
- A single tracking IMS v A single remote site TMS
- A single set of remote site RECON data sets

Table 13-1 on page 229 also describes valid and invalid RSR coexistence configurations. The following notation is used to denote the state of the subsystems and RECON data sets in this table:

- pre-V9 = IMS Version 7, or Version 8
- pre-V9 + SPE = IMS Version 7, or Version 8 with the DBRC coexistence SPE applied
- V9 = IMS Version 9

Table 13-1 Valid RSR coexistence subsystem configurations

Active Site State			**Remote Site State**			
IMS	**TMS ILS**	**RECON**	**IMS**	**TMS**	**RECON**	**NOTES**
pre-V9	pre-V9	pre-V9	pre-V9	pre-V9	pre-V9	Supported
pre-V9 + SPE	pre-V9	pre-V9	pre-V9	pre-V9	pre-V9	Supported
pre-V9 + SPE	V9	V9	pre-V9	pre-V9	pre-V9	Supported
pre-V9	V9	pre-V9	pre-V9	pre-V9	pre-V9	Not Supported. The local TMS/ILS connot be migrated to IMS Version 9 if the RECON data set have not been upgraded to IMS Version 9.
pre-V9 + SPE	pre-V9 + SPE	pre-V9	pre-V9	pre-V9	pre-V9	Supported
V9	pre-V9 + SPE	pre-V9	pre-V9	pre-V9	pre-V9	Not supported. The RECON data sets must be upgraded prior to running any IMS at the IMS Version 9 level. The active IMS cannot be at the IMS Version 9 level if the local site TMS that is running ILS is not at the same level. The local site IMS cannot be at a later level than the remote IMS.
pre-V9 + SPE	pre-V9 + SPE	V9	pre-V9	pre-V9	pre-V9	Supported
pre-V9	V9	V9	pre-V9	pre-V9	pre-V9	Not supported. The local site TMS that is running ILS cannot be at the IMS Version 9 level if the tracking IMS is at the pre-Version 9 level. The active IMS also cannot be at the IMS Version 9 level.
V9	V9	V9	pre-V9	pre-V9	pre-V9	Not supported. The local site TMS that is running ILS and the active IMS cannot be at the IMS Version 9 level if the tracking IMS is at the pre-Version 9 level. The active IMS also cannot be at the IMS Version 9 level.
pre-V9	pre-V9	pre-V9	pre-V9 + SPE	pre-V9	pre-V9	Supported
pre-V9 + SPE	pre-V9	pre-V9	pre-V9 + SPE	pre-V9	pre-V9	Supported

Active Site State			Remote Site State			
IMS	TMS ILS	RECON	IMS	TMS	RECON	NOTES
pre-V9 + SPE	pre-V9 + SPE	pre-V9	pre-V9 + SPE	pre-V9	pre-V9	Supported

13.3.4 Additional coexistence considerations

The OLCSTAT data set (used for global online change) is not compatible with Version 7 IMSs.

Extended checkpoint restriction

You cannot use extended checkpoint to restart applications across different releases of IMS.

13.3.5 Using IRLM 2.2

You can run IRLM 2.2 with the same procedure that you use for IRLM 2.1 by changing the procedure to use the IRLM 2.2 SDXRRESL.

The following execution parameters have changed for IRLM 2.2:

- PC=NO is ignored, IRLM 2.2 uses always PC=YES
- The MAXCSA value is ignored because it applies only to the PC=NO environment.
- The MVS EXEC parameter MEMLIMIT has been added to the IRLM 2.2 startup procedure. The value for MEMLIMIT is set to 2 GB by default. IRLM 2.2 only accepts values of 2 GB or higher.
- The default for DEADLOK is (1,1).

You can use IRLM 2.2 with IRLM 2.1 in an IMS Version 9 environment, including a data-sharing environment. IRLM 2.2 can use storage above the 2 GB bar for 64-bit processors, but IRLM 2.2 can also run on 32-bit processors. You can also use IRLM 2.2 with IMS Version 8 or IMS Version 7 using either 64-bit or 32-bit processors.

13.4 RECON upgrade

IMS Version 7 RECONs may be upgraded directly to IMS Version 9. Similarly, IMS Version 8 RECONs may be upgraded to IMS Version 9. There is no support to upgrade RECONs from previous IMS releases to IMS Version 9. The following APARs should be applied to IMS Version 7 or IMS Version 8 before its RECONs are upgraded to IMS Version 9:

- PQ72838 is an IMS Version 7 APAR which allows IMS Version 7 to use RECONs which have been upgraded to IMS Version 9.
- PQ72840 is an IMS Version 8 APAR which allows IMS Version 8 to use RECONs which have been upgraded to IMS Version 9.

RECONs are upgraded to IMS Version 9 by using the DBRC `CHANGE.RECON UPGRADE` command with the IMS Version 9 DBRC utility (DSPURX00).

The upgrade may be run while the RECONs are allocated to and being used by IMS Version 7 or IMS Version 8, Of course, these systems must be able to use IMS Version 9 RECONs. The upgrade checks the RECONs to ensure that any subsystems using the RECONs are capable of using IMS Version 9 RECONs. It does this by examining the SUBSYS records in the RECONs. Some IMS utilities do not create SUBSYS records. Thus, the upgrade cannot determine if they are running. Users must ensure that any IMS utility which is running at the time of the upgrade has the appropriate maintenance which allows it to read IMS Version 9 RECONs.

Upgrade from IMS Version 7 to IMS Version 9 reads SSYS records to check for DBRC SPE, updates every record from IMS Version 7 to IMS Version 8 format due to segmented record support which was added in IMS Version 8 and updates RECON header from IMS Version 7 to IMS Version 9.

Upgrade from IMS Version 8 to IMS Version 9 reads SSYS records to check for DBRC SPE, and updates the RECON header from IMS Version 8 to IMS Version 9. This should be very quick.

When RECONs are upgraded, any HALDB databases in the RECONs are marked as incapable of running integrated online reorganization. Users can issue CHANGE.DB commands to make them capable of using online reorganization.

When a HALDB master or partition database record from the RECONs is listed, IMS Version 9 includes a field which shows if online reorganization has been enabled for it. Existing HALDBs are upgraded to reflect that they are not capable of running integrated online reorganization and listing the RECON shows it as follows:

- New field for master database listing:

  ```
  ONLINE REORG CAPABLE =YES|NO
  ```

- New fields for partition listing:

  ```
  ONLINE REORG CAPABLE =YES|NO
  ```

- RSR fields:

  ```
  OLR ACTIVE HARD COUNT   =n
  OLR INACTIVE HARD COUNT =n
  ```

If a HALDB is made capable of online reorganization, it cannot be used by IMS Version 7. There is a small programming enhancement (SPE) for IMS Version 8 which allows it to tolerate online reorganization. It allows IMS Version 8 subsystems to use the database while an IMS Version 9 system is doing online reorganization for the database. Of course, IMS Version 8 cannot invoke online reorganization. If the SPE is not applied to the IMS Version 8 system, it has the same restrictions which IMS Version 7 has and which are explained above. The SPE is provided through four APARs. They are PQ78493, PQ78758, PQ78916, and PQ78917.

For more details on RECON and DBRC, see 5.5, “DBRC migration and coexistence” on page 82.

Note: The batch RECON upgrade utility, DSPURU00, is no longer supported

You must issue the **CHANGE.RECON UPGRADE** command using the DBRC utility (DSPURX00). You cannot issue the command from an online IMS.

The MINVERS keyword of the **INIT.RECON** and **CHANGE.RECON** commands specifies the minimum version of IMS that can sign on to DBRC. When a RECON data set is migrated to an IMS Version 9 level, DBRC sets a default of MINVERS(’7.1’). The value that is set for the minimum version is displayed in the **LIST.RECON** command output.

When all systems that access the RECON data set are at an IMS Version 9 level, you can reset the minimum version value by using the MINVERS keyword on the IMS Version 9

`CHANGE.RECON` command. After you set the MINVERS level for an IMS system, system sign on fails for earlier versions of IMS.

13.5 Fallback considerations

IMS does not generally support downward compatibility in any major function between releases. However, consider the following list when preparing your migration fallback plan. This information is intended as a guide to understanding fallback inhibitors, and should not be considered complete:

The ACB maintenance utility, ACBGEN, is required.

For each IMS that you are migrating to a pre-Version 9 level, perform the following steps:

- Shut down that IMS
- Install the version of IMS that you want
- Run INITMOD to initialize the MODSTAT data set
- Cold start the IMS

You can use the IMS Queue Control Facility (QCF) to convert IMS Version 9 messages to IMS Version 8 or IMS Version 7 messages.

For IMS Version 8 subsystems with the OLR coexistence SPE applied, full data sharing is allowed and no immediate action is necessary. For IMS Version 8 subsystems without the OLR coexistence SPE applied, and for IMS Version 7 subsystems, databases that are defined as not capable of being reorganized online are available and no action is required.

However, if any M-V (and Y) DBDSs are active or online reorganization is active for any partitions, then you must perform the following actions whenever processing those partitions before falling back to use the IMS Version 8 utilities:

- Reorganize all partitions with the M-V (and Y) data sets active or online reorganization active using the IMS Version 9 OLR or the offline reorganization utility.

  This leaves only the A-J (and X) data sets active.

- Change DBRC for the databases to reflect that they are not capable of being reorganized online using the IMS Version 9 command:

  ```
  CHANGE.DB DBD(HALDB master) OLRNOCAP
  ```

If you must fall back to an earlier version of IMS, you must set DBRC for all HALDBs to OLRNOCAP. Take a listing of the RECON and verify that there is no entry with an ACTIVE DBDS=M-V (and Y) data sets or an ACTIVE DBDS=A-J (and X) and M-V (and Y).

A

DBRC API control blocks

In this appendix, we show the layout of DBRC API control blocks:

Constants and equates needed by the DBRC API

Example: A-1 DSPAPI FUNC=DSECT - constants and equates needed by the API

```
+DSPAPI_Plist_Func         EQU   0         Function                      01-DSPAP
+DSPAPI_Plist_Type         EQU   1         Type                          01-DSPAP
+DSPAPI_Plist_Version      EQU   4         Version                       01-DSPAP
+DSPAPI_Plist_Token        EQU   8         Token                         01-DSPAP
+DSPAPI_Plist_Output       EQU   12        Output                        01-DSPAP
+DSPAPI_Plist_Retcode      EQU   16        Return Code                   01-DSPAP
+DSPAPI_Plist_Rsncode      EQU   20        Reason Code                   01-DSPAP
+DSPAPI_Plist_Sysprint     EQU   24        SYSPRINT                      01-DSPAP
+DSPAPI_Plist_Buffer       EQU   24        Buffer                        01-DSPAP
+DSPAPI_Plist_Name         EQU   28        DBname/DBlist/Name            01-DSPAP
+DSPAPI_Plist_DDN          EQU   32        DDN                           01-DSPAP
+DSPAPI_Plist_SSID         EQU   36        SSID                          01-DSPAP
+DSPAPI_Plist_Startime     EQU   40        Start time                    01-DSPAP
+DSPAPI_Plist_Flag1        EQU   44        Flags 1                       01-DSPAP
+DSPAPI_Plist_Flag2        EQU   45        Flags 2                       01-DSPAP
+DSPAPI_Plist_Flag3        EQU   46        Flags 3                       01-DSPAP
+DSPAPI_Plist_Flag4        EQU   47        Flags 4                       01-DSPAP
+DSPAPI_Plist_SaveR15      EQU   48        Reg 15 Save area              01-DSPAP
+DSPAPI_Plist_SaveR0       EQU   52        Reg 0 Save area               01-DSPAP
+DSPAPI_Plist_Reserve1     EQU   56        Reserved                      01-DSPAP
+DSPAPI_Plist_Reserve2     EQU   60        Reserved                      01-DSPAP
+DSPAPI_Plist_EOL          EQU   64        End of Plist                  01-DSPAP
+* These Equates define the various API Functions
+DSPAPI_Func_STARTDBRC     EQU   01        STARTDBRC                     01-DSPAP
+DSPAPI_Func_STOPDBRC      EQU   02        STOPDBRC                      01-DSPAP
+DSPAPI_Func_QUERY         EQU   03        QUERY                         01-DSPAP
+DSPAPI_Func_RELBUF        EQU   04        RELBUF                        01-DSPAP
+DSPAPI_Func_DSECT         EQU   05        DSECT                         01-DSPAP
+* These Equates define the various API Types
+DSPAPI_Type_DB            EQU   01        Type=DB                       01-DSPAP
+DSPAPI_Type_RECON         EQU   02        Type=RECON                    01-DSPAP
+DSPAPI_Type_DBDSGROUP     EQU   03        Type=DBDSGROUP                01-DSPAP
+DSPAPI_Type_DBGROUP       EQU   04        Type=DBGROUP                  01-DSPAP
+DSPAPI_Type_RECOVGROUP    EQU   05        Type=RECOVGROUP               01-DSPAP
+DSPAPI_Type_CAGROUP       EQU   06        Type=CAGROUP                  01-DSPAP
+DSPAPI_Type_GSGROUP       EQU   07        Type=GSGROUP                  01-DSPAP
+DSPAPI_Type_BACKOUT       EQU   08        Type=BACKOUT                  01-DSPAP
+DSPAPI_Type_LOG           EQU   09        Type=LOG                      01-DSPAP
+DSPAPI_Type_OLDS          EQU   10        Type=OLDS                     01-DSPAP
+DSPAPI_Type_SUBSYS        EQU   11        Type=SUBSY                    01-DSPAP
+* These Equates are for the various API options.
+DSPAPI_Namelist           EQU   X'80'     NAMELIST specified            01-DSPAP
+DSPAPI_CAINFO_YES         EQU   X'40'     CAINFO = YES                  01-DSPAP
+DSPAPI_TRACKER_YES        EQU   X'20'     TRACKER = YES   (Type=LOG)    01-DSPAP
+DSPAPI_SSTYPE_ONLINE      EQU   X'10'     SSTYPE=ONLINE   (TYPE=SUBSYS) 01-DSPAP
+DSPAPI_SSTYPE_BATCH       EQU   X'08'     SSTYPE=BATCH    (TYPE=SUBSYS) 01-DSPAP
+DSPAPI_Reserved_1         EQU   X'04'                                   01-DSPAP
+DSPAPI_DDN_All            EQU   X'02'     DDN=*                         01-DSPAP
+DSPAPI_Reserved_2         EQU   X'01'                                   01-DSPAP
+DSPAPI_First              EQU   X'80'     LOC FIRST                     01-DSPAP
+DSPAPI_Reserved_3         EQU   X'40'                                   01-DSPAP
+DSPAPI_Next               EQU   X'20'     LOC NEXT                      01-DSPAP
+DSPAPI_Prev               EQU   X'10'     LOC PREV                      01-DSPAP
+DSPAPI_Reserved_4         EQU   X'08'                                   01-DSPAP
+DSPAPI_Reserved_5         EQU   X'04'                                   01-DSPAP
+DSPAPI_Reserved_6         EQU   X'02'                                   01-DSPAP
```

```
+DSPAPI_Reserved_7         EQU   X'01'                                   01-DSPAP
+DSPAPI_DB_ALLOC           EQU   X'80'     LIST=ALLOC      (TYPE=DB)     01-DSPAP
+DSPAPI_DB_IC              EQU   X'40'     LIST=IC         (TYPE=DB)     01-DSPAP
+DSPAPI_DB_RECOV           EQU   X'20'     LIST=RECOV      (TYPE=DB)     01-DSPAP
+DSPAPI_DB_REORG           EQU   X'10'     LIST-REORG      (TYPE=DB)     01-DSPAP
+DSPAPI_Reserved_8         EQU   X'08'                                   01-DSPAP
+DSPAPI_Reserved_9         EQU   X'04'                                   01-DSPAP
+DSPAPI_Reserved_10        EQU   X'02'                                   01-DSPAP
+DSPAPI_Reserved_11        EQU   X'01'                                   01-DSPAP
+DSPAPI_Reserved_12        EQU   X'80'                                   01-DSPAP
+DSPAPI_Reserved_13        EQU   X'40'                                   01-DSPAP
+DSPAPI_Reserved_14        EQU   X'20'                                   01-DSPAP
+DSPAPI_Reserved_15        EQU   X'10'                                   01-DSPAP
+DSPAPI_Reserved_16        EQU   X'08'                                   01-DSPAP
+DSPAPI_Reserved_17        EQU   X'04'                                   01-DSPAP
+DSPAPI_Reserved_18        EQU   X'02'                                   01-DSPAP
+DSPAPI_Reserved_19        EQU   X'01'                                   01-DSPAP
+* These Equates define the Query Output Block Types
+DSPAPI_BlkType_RECON      EQU   X'0100'   RECON Status                  01-DSPAP
+DSPAPI_BlkType_PRILOG     EQU   X'0500'   PRIMARY LOG                   01-DSPAP
+DSPAPI_BlkType_LOGINFO    EQU   X'0501'   Log Information               01-DSPAP
+DSPAPI_BlkType_LOGALL     EQU   X'0700'   LOG ALLOCATION                01-DSPAP
+DSPAPI_BlkType_SECLOG     EQU   X'0900'   SECONDARY LOG                 01-DSPAP
+DSPAPI_BlkType_CAGROUP    EQU   X'0F00'   CA GROUP                      01-DSPAP
+DSPAPI_BlkType_CA         EQU   X'1100'   CA EXECUTION                  01-DSPAP
+DSPAPI_BlkType_DBDSGRP    EQU   X'1600'   DBDS GROUP                    01-DSPAP
+DSPAPI_BlkType_DBGRP      EQU   X'1601'   DB GROUP                      01-DSPAP
+DSPAPI_BlkType_RECOVGRP   EQU   X'1602'   RECOVERY GROUP                01-DSPAP
+DSPAPI_BlkType_IMSDB      EQU   X'1800'   IMS DB                        01-DSPAP
+DSPAPI_BlkType_HALDB      EQU   X'1801'   HALDB                         01-DSPAP
+DSPAPI_BlkType_FPDEDB     EQU   X'1802'   Fast Path DEDB                01-DSPAP
+DSPAPI_BlkType_DBNotFound EQU   X'18FF'   DB Not Found                  01-DSPAP
+DSPAPI_BlkType_PART       EQU   X'1900'   PARTition                     01-DSPAP
+DSPAPI_BlkType_DBDS       EQU   X'2000'   DBDS HEADER                   01-DSPAP
+DSPAPI_BlkType_AREA       EQU   X'2100'   Area                          01-DSPAP
+DSPAPI_BlkType_ALLOC      EQU   X'2800'   ALLOCATION                    01-DSPAP
+DSPAPI_BlkType_RCVINFO    EQU   X'2801'   Recovery Information          01-DSPAP
+DSPAPI_BlkType_IC         EQU   X'2D00'   IMAGE COPY                    01-DSPAP
+DSPAPI_BlkType_REORG      EQU   X'3200'   REORGANIZATION                01-DSPAP
+DSPAPI_BlkType_BACKOUT    EQU   X'3500'   BACKOUT                       01-DSPAP
+DSPAPI_BlkType_RECOV      EQU   X'3700'   RECOVERY                      01-DSPAP
+DSPAPI_BlkType_GSG        EQU   X'3A00'   GSG                           01-DSPAP
+DSPAPI_BlkType_SUBSYS     EQU   X'3F00'   SUBSYSTEM                     01-DSPAP
+DSPAPI_BlkType_PRISLDS    EQU   X'4300'   PRIM SYS LOG                  01-DSPAP
+DSPAPI_BlkType_PRITSLDS   EQU   X'4400'   TRACKING PSLDS                01-DSPAP
+DSPAPI_BlkType_SECSLDS    EQU   X'4700'   SEC SYS LOG                   01-DSPAP
+DSPAPI_BlkType_SECTSLDS   EQU   X'4800'   TRACKING SSLDS                01-DSPAP
+DSPAPI_BlkType_PRIOLDS    EQU   X'5300'   PRIMARY ONLINE                01-DSPAP
+DSPAPI_BlkType_SECOLDS    EQU   X'5700'   SEC ONLINE                    01-DSPAP
+* Block version constants                                    @PQ83837
+DSPAPI_BlkVer_1           EQU   X'00000100' Version=1.0                 01-DSPAP
+DSPAPI_BlkVer_101         EQU   X'00000101' Version=1.01                01-DSPAP
+         print off                                                      01-DSPAP
+         print on                                                       01-DSPAP
+**/
```

DSPAPQHD header

Example: A-2   DSPAPQHD - API query output header

```
***********************************************************************
+*                                                                     *
+*  Name:      DSPAPQHD - DBRC API Query output header                 *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   Describes the header of all output areas returned by the DBRC     *
+*   FUNC=QUERY calls.                                                 *
+*                                                                     *
+*   This macro can be included or invoked multiple times without      *
+*   problem - both in assembler and PL/X form.                        *
+*                                                                     *
+*   The header described by this macro is included in several DBRC    *
+*   API control blocks.  Changing the block's length will require     *
+*   finding both the direct and indirect users of the block.          *
+*                                                                     *
+*                                                                     *
+*   Arguments:  (assembler only)                                      *
+*      DSECT - YES|NO   Determines whether a DSECT statement line is  *
+*                       included in the expansion.  Use NO if the     *
+*                       block is to be imbedded in a larger block.    *
+*                                                                     *
+*      PREFIX-          Determines what prefix is to be added to the  *
+*                       assembler DSECT symbolic names for fields.    *
+*                       Default is APQHD.  Don't include the _ char.  *
+*                       All requested prefixes must be reflected in   *
+*                       the macro code to ensure that xref searches   *
+*                       can be done when needed.                      *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQHD       DSECT                OUTPUT HEADER                       01-DSPAP
+               DS  0F                                                  01-DSPAP
+APQHD_EYECATCHER DS CL8          Output area eyecatcher                01-DSPAP
+APQHD_LENGTH  DS  F              Block length, hdr + data              01-DSPAP
+APQHD_BLKTYPE DS  XL2            Block type                            01-DSPAP
+             DS  H               Reserved                              01-DSPAP
+APQHD_DEPPTR  DS  A              Ptr to block dependent                01-DSPAP
+APQHD_NEXTPTR DS  A              Ptr to next block of the same type    01-DSPAP
```

```
+APQHD_BLKOFFSET DS F              Offset to block data                  01-DSPAP
+APQHD_VERSION DS  F               Version of output block               01-DSPAP
+APQHD_APQHD_LNG EQU *-APQHD_EYECATCHER Length of header                 01-DSPAP
```

DSPAPQAL - allocation block

Example: A-3   DSPAPQAL - allocation block

```
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQAL - Allocation block                             *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the Allocation output block returned on the       *
+*   DSPAPI FUNC=QUERY TYPE=DB call.                                   *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQAL'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'2800'  (DSPAPI_BlkType_ALLOC)              *
+*     apqhd_depptr     = 0, no dependent blocks                       *
+*     apqhd_nextptr    = pointer to next ALLOC block for the DBDS     *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQAL             *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
```

```
+************************************************************************
+DSPAPQAL                DSECT        Allocation block                   01-DSPAP
+apqal_DBname            DS    CL8    Database name                      01-DSPAP
+apqal_DDname            DS    OCL8   DD name or                         01-DSPAP
+apqal_Areaname          DS    CL8    Area name                          01-DSPAP
+apqal_alloctm           DS    XL12   Allocation time                    01-DSPAP
+apqal_daltm             DS    XL12   Deallocation time                  01-DSPAP
+apqal_strtm             DS    XL12   Log start time                     01-DSPAP
+apqal_dssn              DS    F      Field for DSSN value               01-DSPAP
+apqal_usid              DS    F      Update set identifier              01-DSPAP
+apqal_alrid             DS    XL8    LRID of begin-upd rec              01-DSPAP
+apqal_dlrid             DS    XL8    LRID of end-upd rec                01-DSPAP
+apqal_slrid             DS    XL8    Last LRID applied if suspended     01-DSPAP
+apqal_flags             DS    XL1    Flags                              01-DSPAP
+apqal_tsusp             EQU   X'80'  Tracking is suspended              01-DSPAP
+apqal_nappl             EQU   X'40'  No records applied                 01-DSPAP
+apqal_cicpt             EQU   X'20'  Fuzzy ic purge time                01-DSPAP
+                        DS    CL3    Reserved                           01-DSPAP
+************************************************************************
+*  END OF DSPAPQAL                                                      *
+************************************************************************
```

DSPAPQAR - Fast Path area block

Example: A-4 DSPAPQAR - Fast Path area block

```
+*******************************************************@SCPYRT**@SCPYRT*
+*                                                                      *
+*  Licensed Materials - Property of IBM                                *
+*                                                                      *
+*  5655-J38                                                            *
+*                                                                      *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved               *
+*                                                                      *
+*  US Government Users Restricted Rights - Use, duplication or         *
+*  disclosure restricted by GSA ADP Schedule contract with             *
+*  IBM Corp.                                                           *
+*                                                                      *
+****************************************************************@ECPYRT*
+*                                                                      *
+*  Name:      DSPAPQAR - Fast Path Area block                          *
+*                                                                      *
+*  Product:   DBRC Application Programming Interface                   *
+*                                                                      *
+*  Function:                                                           *
+*                                                                      *
+*   This macro maps the Fast Path Area output block returned on        *
+*   the DSPAPI FUNC=QUERY TYPE=DB call.                                *
+*                                                                      *
+*   The header of this block is mapped by DSPAPQHD which contains      *
+*   the following information:                                         *
+*                                                                      *
+*     apqhd_eyecatcher = 'DSPAPQAR'                                    *
+*     apqhd_length     = block length                                  *
+*     apqhd_blktype    = X'2100'  (DSPAPI_BlkType_AREA)                *
+*     apqhd_depptr     = pointer to Recovery Information               *
+*                        block (DSPAPQRI)                              *
+*     apqhd_nextptr    = pointer to next Area block                    *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQAR             *
+*                                                                     *
+*   Within DSPAPQAR are the following fields which are used to        *
+*   access the Area Data Set, EEQE, and authorized subsystem          *
+*   information for the area:                                         *
+*                                                                     *
+*     apqar_ADSlist    = offset from DSPAPQAR to ADS data             *
+*                        (apqar_adslt)                                *
+*     apqar_ADSLength  = length of each ADS entry                     *
+*     apqar_NOADS      = number of ADSs                               *
+*                                                                     *
+*     apqar_EEQElist   = offset from DSPAPQAR to EEQE                 *
+*                        data (DSPAPQEL)                              *
+*     apqar_EEQELength = length of each EEQE entry                    *
+*     apqar_EEQECount  = number of EEQEs                              *
+*                                                                     *
+*     apqar_SSlist     = offset from DSPAPQAR to SS                   *
+*                        data (DSPAPQSL)                              *
+*     apqar_SSEntLen   = length of each SS entry                      *
+*     apqar_SSNUM      = number of authorized subsystems              *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*    2004-03-04 1.01  PQ83837  pas  Add user-recoverable flag         *
+*                                                                     *
+***********************************************************************
+DSPAPQAR                DSECT        Fast Path Area block               01-DSPAP
+apqar_DBname            DS    CL8    Database name                      01-DSPAP
+apqar_Areaname          DS    CL8    Area name                          01-DSPAP
+apqar_EEQElist          DS    F      Offset to EEQE list (DSPAQEL),     01-DSPAP
+*                                    zero if no EEQEs
+apqar_SSlist            DS    F      Offset to SS list (DSPAPQSL),      01-DSPAP
+*                                    zero if no SS auth'd
+apqar_ADSlist           DS    F      Offset to ADS list,                01-DSPAP
+*                                    zero if none registered
+                        DS    F      Reserved                           01-DSPAP
+apqar_SHRLVL            DS    FL1    Share level of DB                  01-DSPAP
+apqar_HELDAU            DS    FL1    Held auth state                    01-DSPAP
+apqar_HAUBIT            EQU   X'80'  High order bit flag                01-DSPAP
+apqar_DMBNUM            DS    H      Global DMB number                  01-DSPAP
+apqar_SSNUM             DS    H      # subsytems authd to Area          01-DSPAP
+apqar_SSEntLen          DS    H      Length of each SS entry            01-DSPAP
+apqar_CACCSS            DS    FL1    Access state for CHG AUTH          01-DSPAP
+apqar_CANCDD            DS    FL1    Encoded state for CHG AUTH         01-DSPAP
```

```
+apqar_CAHELD            DS    FL1    Held state for CHG AUTH            01-DSPAP
+apqar_IRLMAU            DS    CL5    IRLMID of auth SS                  01-DSPAP
+apqar_FLAGS             DS    XL2                                       01-DSPAP
+apqar_RECYC             EQU   X'8000' REUSE image copies                01-DSPAP
+apqar_ICREC             EQU   X'4000' Image Copy Recommended            01-DSPAP
+apqar_IC                EQU   X'2000' Image Copy Needed                 01-DSPAP
+apqar_ICNDIS            EQU   X'1000' IC needed disabled option         01-DSPAP
+*                                     1 = IC Needed Disabled
+apqar_RECOV             EQU   X'0800' Recovery needed                   01-DSPAP
+apqar_INPRO             EQU   X'0400' HSSP CIC in progress              01-DSPAP
+apqar_GT240             EQU   X'0200' M/C FP GT240 area DEDB            01-DSPAP
+apqar_VSO               EQU   X'0100' VSO flag                          01-DSPAP
+apqar_PREOP             EQU   X'0080' PREOPEN flag                      01-DSPAP
+apqar_PRELD             EQU   X'0040' PRELOAD flag                      01-DSPAP
+apqar_LKASD             EQU   X'0020' VSO CF buffer lookaside           01-DSPAP
+apqar_MAS               EQU   X'0010' VSO area resides in multi-area    01-DSPAP
+*                                     CF structure
+apqar_RRGAL             EQU   X'0008' REORG since last ALLOC,           01-DSPAP
+*                                     only set if RSR-covered
+apqar_TSRAL             EQU   X'0004' TS recov since last ALLOC,        01-DSPAP
+*                                     only set if RSR-covered
+apqar_RSRflags          DS    XL1    Remote Site Recovery flags         01-DSPAP
+apqar_RCVTRK            EQU   X'80'  Recovery Level Tracking            01-DSPAP
+apqar_TRKSPN            EQU   X'40'  Tracking was suspended             01-DSPAP
+apqar_PURBIT            EQU   X'20'  Suspended by time                  01-DSPAP
+apqar_RCVRQ             EQU   X'10'  Receive Required                   01-DSPAP
+apqar_AUFLAG            DS    XL1    Authorization flags                01-DSPAP
+apqar_PAFLG             EQU   X'80'  Prohibit authorization             01-DSPAP
+apqar_NONRV             EQU   X'40'  Non-recoverable                    01-DSPAP
+apqar_USRRV             EQU   X'20'  User-recoverable (VERSION=1.01)    01-DSPAP
+*                                                              @PQ83837
+apqar_DSORG             DS    XL1    Data set organization              01-DSPAP
+apqar_VSAM              EQU   X'80'  1 = VSAM, 0 = NON-VSAM             01-DSPAP
+apqar_INDEX             EQU   X'40'  0 = Non-indexed (OSAM or ESDS),    01-DSPAP
+*                                    1 = Indexed(ISAM or KSDS)
+apqar_DBORG             DS    XL1    IMS DB organization                01-DSPAP
+apqar_GSGname           DS    CL8    GSG Name                           01-DSPAP
+                        DS    CL2    Reserved                           01-DSPAP
+apqar_USID              DS    F      Last ALLOC USID                    01-DSPAP
+apqar_AUSID             DS    F      Last authorized USID               01-DSPAP
+apqar_RUSID             DS    F      Last received USID                 01-DSPAP
+apqar_HUSID             DS    F      Hardened USID                      01-DSPAP
+apqar_RNUSID            DS    F      Receive needed USID                01-DSPAP
+apqar_RecovGrp          DS    CL8    Recovery Group name                01-DSPAP
+apqar_CAGrpName         DS    CL8    Change Accum group name            01-DSPAP
+apqar_GENMX             DS    H      Max number of ICs that may be      01-DSPAP
+*                                    predefined for this area
+apqar_GENNO             DS    H      Number of available ICs for        01-DSPAP
+*                                    this area
+apqar_USDIC             DS    H      Number of ICs used                 01-DSPAP
+apqar_EEQECount         DS    H      EEQE count                         01-DSPAP
+apqar_EEQELength        DS    H      EEQE entry length                  01-DSPAP
+apqar_NOADS             DS    FL1    # of ADS in the area               01-DSPAP
+apqar_AVADS             DS    FL1    # of available ADS                 01-DSPAP
+apqar_ADSLength         DS    H      ADS entry length                   01-DSPAP
+                        DS    CL2    Reserved                           01-DSPAP
+apqar_JCL               DS    0CL40  GENJCL members                     01-DSPAP
+apqar_ICJCL             DS    CL8     Image copy member                 01-DSPAP
+apqar_OIJCL             DS    CL8     Online IC member                  01-DSPAP
+apqar_RCJCL             DS    CL8     Recovery member                   01-DSPAP
```

```
+apqar_DFJCL             DS    CL8     DEFLTJCL member                  01-DSPAP
+apqar_RVJCL             DS    CL8     Receive JCL member               01-DSPAP
+apqar_RTPRD             DS    H      IC retention period               01-DSPAP
+apqar_DSID              DS    H      IMS data set ID                   01-DSPAP
+apqar_DSSN              DS    F      Data set sequence number          01-DSPAP
+apqar_CFST1             DS    CL16   VSO CF Structure 1                01-DSPAP
+apqar_CFST2             DS    CL16   VSO CF Structure 2                01-DSPAP
+apqar_adslt             DSECT        Area Data Set List                01-DSPAP
+apqar_ADSDD             DS    CL8    DDNAME of the ADS                 01-DSPAP
+apqar_ADSDN             DS    CL44   DSN of the ADS                    01-DSPAP
+apqar_ADSBT             DS    XL1    ADS flags                         01-DSPAP
+apqar_ADSAV             EQU   X'80'  Avail status of ADS               01-DSPAP
+apqar_ADSFM             EQU   X'40'  Format status of create util      01-DSPAP
+apqar_ADSCP             EQU   X'20'  Copy status of create util        01-DSPAP
+                        DS    CL3    Reserved                          01-DSPAP
+*********************************************************************
+*  END OF DSPAPQAR                                                  *
+*********************************************************************
```

DSPAPQBO - backout block

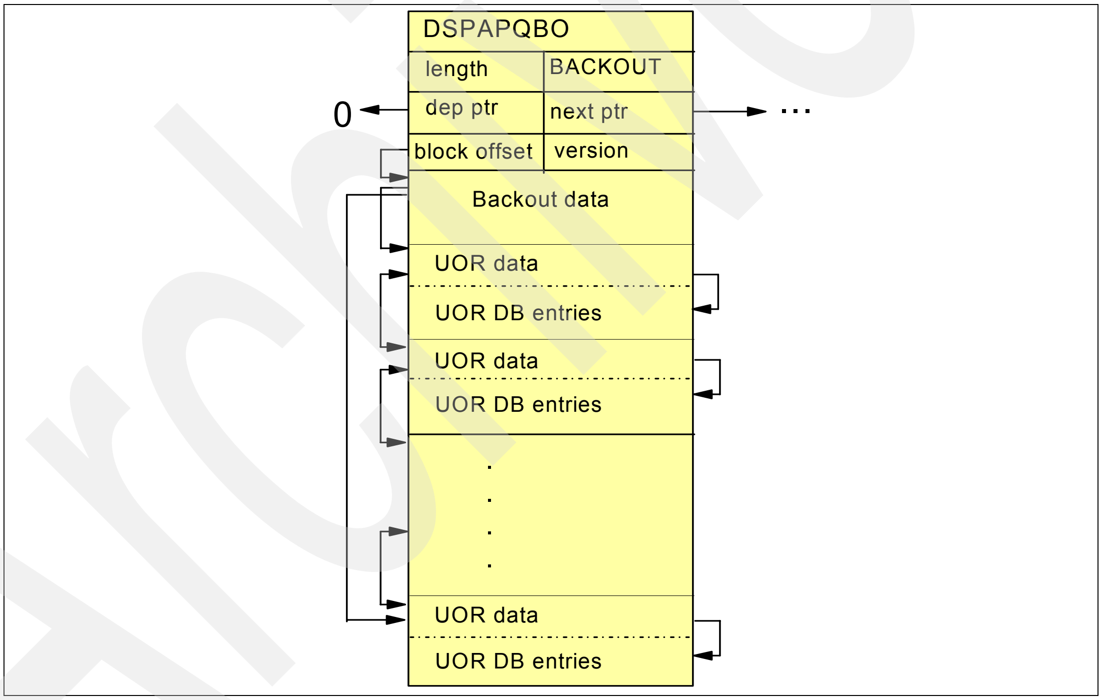


Figure A-1 Backout output

Example: A-5 DSPAPQBO - backout block

```
DSPAPQBO                                                       31479703
+*******************************************************@SCPYRT**@SCPYRT*
+*                                                                      *
+*  Licensed Materials - Property of IBM                                *
+*                                                                      *
+*  5655-J38                                                            *
+*                                                                      *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved               *
+*                                                                      *
+*  US Government Users Restricted Rights - Use, duplication or         *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQBO - Backout block                                 *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                   *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the Backout output block returned on the          *
+*   DSPAPI FUNC=QUERY TYPE=BACKOUT call.                              *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQBO'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'3500'  (DSPAPI_BlkType_BACKOUT)            *
+*     apqhd_depptr     = 0, no dependent block                        *
+*     apqhd_nextptr    = pointer to next Backout block                *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQBO             *
+*                                                                     *
+*   Within DSPAPQBO are the following fields which are used to        *
+*   access the first and the last UOR entries:                        *
+*                                                                     *
+*     apqbo_FirstUOR    = offset from DSPAPQBO to the first UOR       *
+*                         entry (apqbo_UOREntry)                      *
+*                                                                     *
+*     apqbo_LastUOR     = offset from DSPAPQBO to the last UOR        *
+*                         entry                                       *
+*                                                                     *
+*   Within each UOR entry (apqbo_UOREntry) are the following          *
+*   fields which are used to access the previous and next UOR         *
+*   entries as well as the databases associated with the UOR:         *
+*                                                                     *
+*     apqbo_NextUOR     = offset from DSPAPQBO to the next UOR        *
+*                         entry                                       *
+*     apqbo_PrevUOR     = offset from DSPAPQBO to the previous        *
+*                         UOR entry                                   *
+*                                                                     *
+*     apqbo_DS_DBOffset = offset from apqbo_UOREntry to DB entries    *
+*                         (apqbo_DBent)                               *
+*     apqbo_DBCount     = number of DB entries                        *
+*     apqbo_DBLength    = Length of each DB entry                     *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQBO                DSECT        Backout block                      01-DSPAP
+apqbo_SSID              DS    CL8    Subsystem identifier               01-DSPAP
+apqbo_FirstUOR          DS    F      Offset of first UOR entry          01-DSPAP
+apqbo_LastUOR           DS    F      Offset of last UOR entry           01-DSPAP
+apqbo_TimeFirst         DS    XL12   Earliest UOR time                  01-DSPAP
+apqbo_TimeLast          DS    XL12   Latest UOR time                    01-DSPAP
+apqbo_FLAGS             DS    XL1    Backout flags                      01-DSPAP
+apqbo_SAVER             EQU   X'80'  SAVUOR call during restart         01-DSPAP
+                        DS    CL3    Reserved                           01-DSPAP
+apqbo_UORcount          DS    F      Number of UORs                     01-DSPAP
+***********************************************************************
+*                                                                     *
+*  The following structure maps the unit of recovery entries.         *
+*  There is one such entry for each unit of recovery (i.e. there      *
+*  are apqbo_UORcount entries).  Each unit of recovery entry          *
+*  contains the offset within the backout block to the previous       *
+*  and following entries.  Field apqbo_PrevUOR is the offset of       *
+*  the previous entry and apqbo_NextUOR is the offset of the          *
+*  following entry.  For the first unit of recovery (UOR) entry,      *
+*  apqbo_PrevUOR will be zero.  Similarly, apqbo_NextUOR will be      *
+*  zero for the last entry.                                           *
+*                                                                     *
+***********************************************************************
+apqbo_UORentry          DSECT        Unit of Recovery entry             01-DSPAP
+apqbo_prefix            DS    0CL64  Prefix section                     01-DSPAP
+apqbo_NextUOR           DS    F      Offset of next UOR entry           01-DSPAP
+apqbo_PrevUOR           DS    F      Offset of previous UOR entr  y     01-DSPAP
+apqbo_DBoffset          DS    F      Offset to DB entries               01-DSPAP
+apqbo_UORTime           DS    XL12   Time stamp for this UOR            01-DSPAP
+apqbo_UORPSB            DS    CL8    PSB name                           01-DSPAP
+apqbo_UORflags          DS    XL2                                       01-DSPAP
+apqbo_DEFBO             EQU   X'8000' Deferred backout - dynamic        01-DSPAP
+*                                       backout failure
+apqbo_INFLT             EQU   X'4000' Inflight UOR                      01-DSPAP
+apqbo_INDOU             EQU   X'2000' Indoubt UOR                       01-DSPAP
+apqbo_BMP               EQU   X'1000' BMP UOR                           01-DSPAP
+apqbo_BOCAN             EQU   X'0800' BBO identified candidate          01-DSPAP
+apqbo_COLDN             EQU   X'0400' Cold start ended for UOR          01-DSPAP
+apqbo_BBOK              EQU   X'0200' Backed out OK by BBO              01-DSPAP
+apqbo_CMD               EQU   X'0100' UOR modified by command           01-DSPAP
+apqbo_BATCH             EQU   X'0080' Batch IMS UOR                     01-DSPAP
+                        DS    CL6    Reserved                           01-DSPAP
+apqbo_RTOKN             DS    0CL16  Recovery token                     01-DSPAP
+apqbo_RTSSID            DS    CL8    SSID for this token                01-DSPAP
+apqbo_UORID             DS    CL8    Unique UOR ID                      01-DSPAP
+apqbo_DBcount           DS    F      Number of DBs for this UOR         01-DSPAP
+apqbo_DBLength          DS    H      Length of each DB entry            01-DSPAP
+                        DS    CL2    Reserved                           01-DSPAP
+apqbo_DBent             DSECT        Database entry                     01-DSPAP
+apqbo_DBname            DS    CL8    ! Database name                    01-DSPAP
+apqbo_DBflags           DS    XL1    ! Flags                            01-DSPAP
+apqbo_DBOUT             EQU   X'80'  ! UOR backed out for this DB       01-DSPAP
+apqbo_DBDEF             EQU   X'40'  ! Dyn backout failure this DB      01-DSPAP
```

```
+                        DS    CL7    ! Reserved                         01-DSPAP
+***********************************************************************
+*  END OF DSPAPQBO                                                    *
```

DSPAPQCA - change accumulation execution block

Example: A-6   DSPAPQCA - change accumulation execution block

```
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQCA - CA Execution block                           *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the Change Accum block returned on the            *
+*   DSPAPI FUNC=QUERY TYPE=CAGROUP call.                              *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQCA'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'1100'  (DSPAPI_BlkType_CA)                 *
+*     apqhd_depptr     = 0, no dependent block                        *
+*     apqhd_nextptr    = pointer to next CA block                     *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQCA             *
+*                                                                     *
+*   Within DSPAPQCA are the following fields which are used to        *
+*   access CA member data and the volume information:                 *
+*                                                                     *
+*     apqca_MemberInfo  = offset from DSPAPQCA to member list         *
+*                         (apqca_Member)                              *
+*     apqca_MemberLen   = length of each member entry                 *
+*     apqca_MemberCount = number of group members                     *
+*                                                                     *
+*     apqca_Volinfo     = offset from DSPAPQCA to volume data         *
+*                         (apqca_Vol_Data)                            *
+*                                                                     *
+*   Within the volume data are the following fields which are         *
+*   used to access the list of VOLSERs:                               *
+*                                                                     *
+*     apqca_Vol_Offset  = offset from apqca_Vol_Data to the list      *
+*                         of VOLSERs (apqca_VOLSER)                   *
+*     apqca_Vol_Length  = length of each VOLSER entry                 *
+*     apqca_Vol_Count   = number of VOLSERs                           *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQCA                DSECT        CA Execution block                 01-DSPAP
+apqca_GroupName         DS    CL8    Group name                         01-DSPAP
+apqca_Memberinfo        DS    F      Offset to group member list        01-DSPAP
+apqca_MemberLen         DS    H      Length of each member entry        01-DSPAP
+apqca_MemberCount       DS    H      Number of group members            01-DSPAP
+apqca_Volinfo           DS    F      Offset to volume information       01-DSPAP
+                        DS    F      Reserved                           01-DSPAP
+apqca_DSN               DS    CL44   Data set name                      01-DSPAP
+apqca_StopTime          DS    XL12   Packed decimal date/time - for     01-DSPAP
+*                                    predefined datasets, represents
+*                                    record creation time. Otherwise,
+*                                    it is the stoptime of the last
+*                                    logtape volume used as input to
+*                                    the Change Accumulation utility
+*                                    that produced this CA as output.
+*                                    If the CA run  included  an
+*                                    'incomplete log subset' it is
+*                                    the start time of the first
+*                                    truncated log  volume.
+apqca_RunTime           DS    XL12   CA run time                        01-DSPAP
+apqca_Flags             DS    XL1    Flags                              01-DSPAP
+apqca_Error             EQU   X'80'  Error on data set                  01-DSPAP
+apqca_Subset            EQU   X'40'  Subset of logs used for CA         01-DSPAP
+apqca_Command           EQU   X'20'  SUBSET/COMP has been set or        01-DSPAP
+*                                    reset with an external cmd
+                        DS    CL3    Reserved                           01-DSPAP
+apqca_Member            DSECT        List of group members              01-DSPAP
+apqca_Mem_DBname        DS    CL8    Database name                      01-DSPAP
+apqca_Mem_DDname        DS    OCL8   DD name or                         01-DSPAP
+apqca_Mem_Areaname      DS    CL8    AREA name                          01-DSPAP
+apqca_Mem_DSSN          DS    F      Data Set sequence number           01-DSPAP
+apqca_Mem_USID          DS    F      USID of last change                01-DSPAP
+*                                    accumulated
+apqca_Mem_LRID          DS    XL8    LRID of last change                01-DSPAP
+*                                    accumulated
+apqca_Mem_PurgeTime     DS    XL12   Purge time                         01-DSPAP
+apqca_Mem_LSN           DS    XL6    Lock sequence number               01-DSPAP
+apqca_Mem_Flags         DS    XL1    Member flags                       01-DSPAP
+apqca_Mem_NoChg         EQU   X'80'  No changes accumulated             01-DSPAP
+apqca_Mem_Indoubt       EQU   X'40'  Indoubt EEQEs accumulated          01-DSPAP
```

```
+apqca_Mem_Incomp        EQU   X'20'  Incomplete CA                      01-DSPAP
+                        DS    CL5    Reserved                           01-DSPAP
+apqca_Vol_Data          DSECT        Volume data                        01-DSPAP
+apqca_Vol_FileSeq       DS    H      File sequence number               01-DSPAP
+apqca_Vol_Unit          DS    CL8    Unit type                          01-DSPAP
+apqca_Vol_Count         DS    H      Number of VOLSERs defined          01-DSPAP
+apqca_Vol_Used          DS    H      Number of VOLSERs used             01-DSPAP
+apqca_Vol_Length        DS    H      Length of each VOLSER              01-DSPAP
+apqca_Vol_Offset        DS    F      Offset to VOLSER information       01-DSPAP
+                        DS    CL4    Reserved                           01-DSPAP
+                        DSECT                                           01-DSPAP
+apqca_VOLSER            DS    CL6    List of VOLSERs                    01-DSPAP
+***********************************************************************
+*  END OF DSPAPQCA                                                    *
```

DSPAPQCG - change accumulation group block

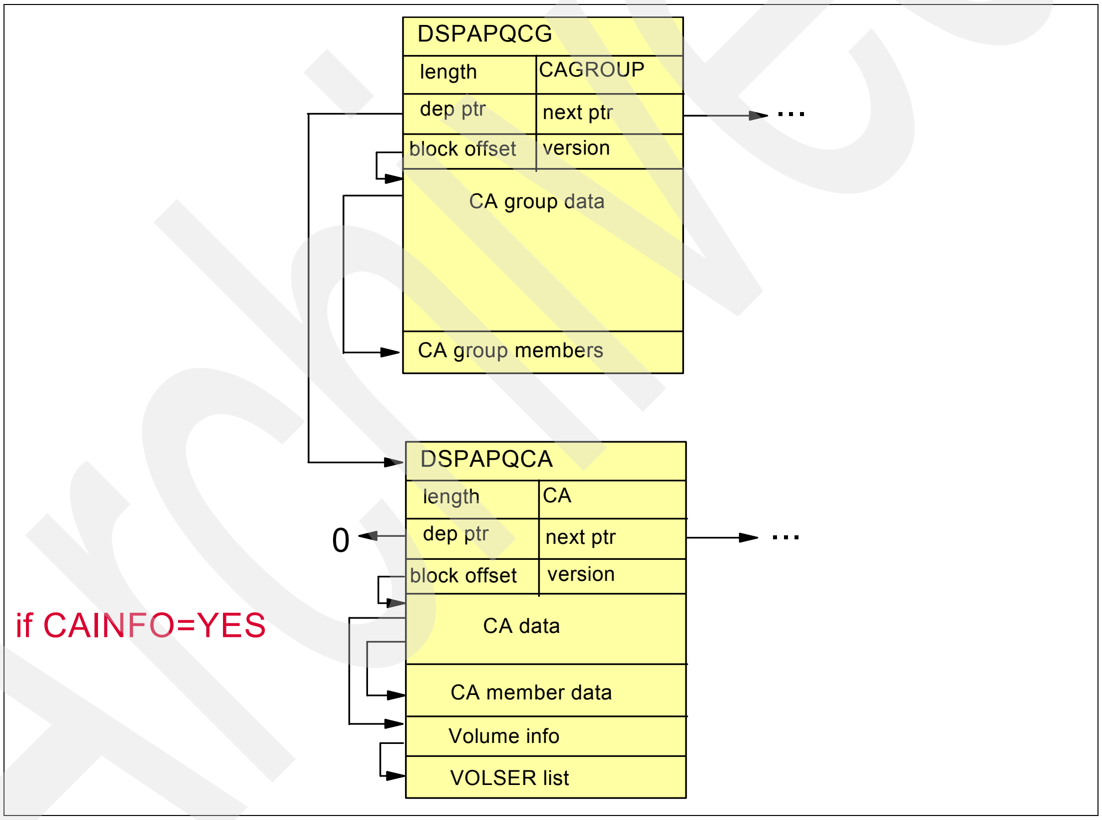


Figure A-2 CAGROUP output

Example: A-7 DSPAPQCG - change accumulation group block

```
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQCG - CA Group block                               *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the CA Group block returned on the                *
+*   DSPAPI FUNC=QUERY TYPE=CAGROUP call.                              *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQCG'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'0F00'  (DSPAPI_BlkType_CAGROUP)            *
+*     apqhd_depptr     = pointer to CA Execution block (DSPAPQCA)     *
+*     apqhd_nextptr    = pointer to next CA Group block               *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQCG             *
+*                                                                     *
+*   Within DSPAPQCG are the following fields which are used to       *
+*   access CA Group member data:                                      *
+*                                                                     *
+*     apqcg_MemberInfo  = offset from DSPAPQCG to member list         *
+*                       (apqcg_Member)                                *
+*     apqcg_MemberLen   = length of each member entry                 *
+*     apqcg_MemberCount = number of group members                     *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQCG                DSECT        CA Group block                     01-DSPAP
+apqcg_GroupName         DS    CL8    Group name                         01-DSPAP
+apqcg_Memberinfo        DS    F      Offset to group member list        01-DSPAP
+apqcg_MemberLen         DS    H      Length of group member entry       01-DSPAP
+apqcg_MemberCount       DS    H      Number of group members            01-DSPAP
+                        DS    2F     Reserved                           01-DSPAP
+apqcg_GrpMax            DS    H      Maximum number of CAs that may     01-DSPAP
+*                                    be predefined for this CA group
```

```
+apqcg_AvailCA#         DS    H      Number of available CA data      01-DSPAP
+*                                   sets for this group
+apqcg_UsedCA#          DS    H      Number of used CA data sets      01-DSPAP
+apqcg_Flags            DS    XL1    Flags                            01-DSPAP
+apqcg_Reuse            EQU   X'80'  Reuse CA data sets               01-DSPAP
+                       DS    CL1    Reserved                         01-DSPAP
+apqcg_CAJCL            DS    CL8    GENJCL CAJCL member name         01-DSPAP
+apqcg_DFJCL            DS    CL8    DEFLTJCL member name             01-DSPAP
+apqcg_Member           DSECT        List of group members            01-DSPAP
+apqcg_DBname           DS    CL8    Database name                    01-DSPAP
+apqcg_DDname           DS    0CL8   DD name or                       01-DSPAP
+apqcg_Areaname         DS    CL8    AREA name                        01-DSPAP
+***********************************************************************
+*   END OF DSPAPQCG                                                   *
```

DSPAPQDB - full function database block

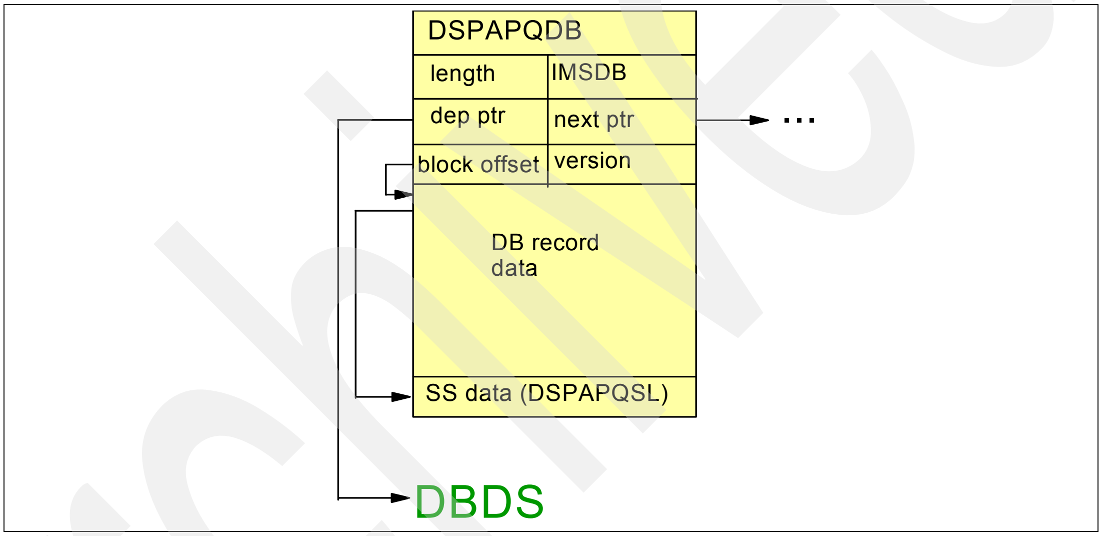


Figure A-3 Non-HALDB full function output

Example: A-8 DSPAPQDB - full function database block

```
+*******************************************************@SCPYRT**@SCPYRT*
+*                                                                    *
+*   Licensed Materials - Property of IBM                             *
+*                                                                    *
+*   5655-J38                                                         *
+*                                                                    *
+*   (C) Copyright IBM Corp. 1974,2003 All Rights Reserved            *
+*                                                                    *
+*   US Government Users Restricted Rights - Use, duplication or      *
+*   disclosure restricted by GSA ADP Schedule contract with          *
+*   IBM Corp.                                                        *
+*                                                                    *
+****************************************************************@ECPYRT*
+*                                                                    *
+*   Name:       DSPAPQDB - Full Function DB block                    *
+*                                                                    *
+*   Product:    DBRC Application Programming Interface               *
+*                                                                    *
+*   Function:                                                        *
+*                                                                     *
+*   This macro maps the Full Function DB output block returned        *
+*   on the DSPAPI FUNC=QUERY TYPE=DB call.                            *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQDB'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'1800'  (DSPAPI_BlkType_IMSDB)              *
+*     apqhd_depptr     = pointer to DBDS block (DSPAPQDS)             *
+*     apqhd_nextptr    = pointer to next database block               *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQDB             *
+*                                                                     *
+*   Within DSPAPQDB are the following fields which are used to        *
+*   access the authorized subsystem information for the DB:           *
+*                                                                     *
+*     apqdb_SSlist     = offset from DSPAPQDB to SS                   *
+*                        data (DSPAPQSL)                              *
+*     apqdb_SSEntLen   = length of each SS entry                      *
+*     apqdb_SSNUM      = number of authorized subsystems              *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQDB                DSECT        Full Function DB block             01-DSPAP
+apqdb_DBname            DS    CL8    Database name                      01-DSPAP
+apqdb_SSlist            DS    F      Offset to SS list (DSPAPQSL),      01-DSPAP
+*                                    zero is no SS auth'd
+                        DS    3F     Reserved                           01-DSPAP
+apqdb_IRCNT             DS    H      IC receive needed counter          01-DSPAP
+apqdb_AUFLAG            DS    XL1    Authorization flags                01-DSPAP
+apqdb_BKFLG             EQU   X'80'  Backout needed flag                01-DSPAP
+apqdb_PAFLG             EQU   X'40'  Prohibit authorization             01-DSPAP
+apqdb_RDFLG             EQU   X'20'  Read only SS auth                  01-DSPAP
+apqdb_NONRV             EQU   X'10'  Non-recoverable                    01-DSPAP
+apqdb_IRLMAU            DS    CL5    IRLMID of auth SS                  01-DSPAP
+apqdb_RCVCTR            DS    H      Recovery needed count              01-DSPAP
+apqdb_ICCTR             DS    H      IC needed count                    01-DSPAP
+apqdb_ICRecCtr          DS    H      IC recommended counter             01-DSPAP
+apqdb_SHRLVL            DS    FL1    Share level of DB                  01-DSPAP
+apqdb_HELDAU            DS    FL1    Held auth state                    01-DSPAP
+apqdb_HAUBIT            EQU   X'80'  High order bit flag                01-DSPAP
```

```
+apqdb_DMBNUM            DS    H      Global DMB number                  01-DSPAP
+apqdb_SSNUM             DS    H      # of SS auth DB                    01-DSPAP
+apqdb_SSEntLen          DS    H      Length of each SS entry            01-DSPAP
+apqdb_CACCSS            DS    FL1    Access state for chg auth          01-DSPAP
+apqdb_CANCDD            DS    FL1    Encode state for chg auth          01-DSPAP
+apqdb_CAHELD            DS    FL1    Held state for chg auth            01-DSPAP
+                        DS    CL3    Reserved                           01-DSPAP
+apqdb_EQECNT            DS    H      Total EQE count                    01-DSPAP
+apqdb_RSRFLG            DS    XL2    Flags                              01-DSPAP
+apqdb_RCVTRK            EQU   X'8000' Recovery level tracking           01-DSPAP
+apqdb_TRKSPN            EQU   X'4000' Tracking is suspended             01-DSPAP
+apqdb_PURBIT            EQU   X'2000' Suspended by time                 01-DSPAP
+apqdb_ICNDIS            EQU   X'1000' IC needed disabled option         01-DSPAP
+apqdb_GSGname           DS    CL8    Global Service Group name          01-DSPAP
+apqdb_USID              DS    F      Last alloc USID                    01-DSPAP
+apqdb_AUSID             DS    F      Last authorized USID               01-DSPAP
+apqdb_RUSID             DS    F      Last received USID                 01-DSPAP
+apqdb_HUSID             DS    F      Hardened by tracker USID           01-DSPAP
+apqdb_RNUSID            DS    F      Receive needed USID                01-DSPAP
+apqdb_RecovGrp          DS    CL8    Recovery Group name                01-DSPAP
+                        DS    CL4    Reserved                           01-DSPAP
+***********************************************************************
+*  END OF DSPAPQDB                                                    *
+***********************************************************************
```

DSPAPQDG - DBDS, database, and recovery group block

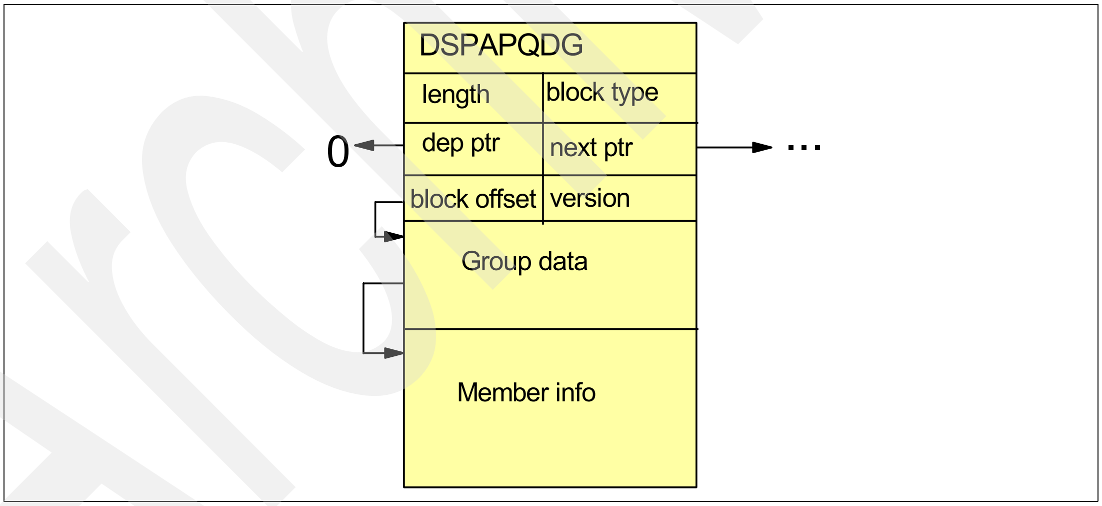


Figure A-4 DBDS, DB, and recovery group output

Example: A-9 DSPAPQDG - DBDS, database, and recovery group block

```
+*******************************************************@SCPYRT**@SCPYRT*
+*                                                                    *
+*  Licensed Materials - Property of IBM                              *
+*                                                                    *
+*  5655-J38                                                          *
+*                                                                    *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved             *
+*                                                                    *
+*  US Government Users Restricted Rights - Use, duplication or       *
+*  disclosure restricted by GSA ADP Schedule contract with           *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQDG - DBDS, Database, and Recovery Group block     *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the DBDS, Database, and Recovery Group output     *
+*   blocks returned on the DSPAPI FUNC=QUERY TYPE=DBDSGROUP |         *
+*   DBGROUP | RECOVGROUP call.                                        *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                       *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQDG'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'1600'  (DSPAPI_BlkType_DBDSGRP)            *
+*                        X'1601'  (DSPAPI_BlkType_DBGRP)              *
+*                        X'1602'  (DSPAPI_BlkType_RECOVGRP)           *
+*     apqhd_depptr     = 0, no dependent block                        *
+*     apqhd_nextptr    = pointer to next group block                  *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQDG             *
+*                                                                     *
+*   Within DSPAPQDG are the following fields which are used to        *
+*   access group member data:                                         *
+*                                                                     *
+*     apqdg_MemberInfo  = offset from DSPAPQDG to member list         *
+*                         (apqdg_Member)                              *
+*     apqdg_MemberLen   = length of each member entry                 *
+*     apqdg_MemberCount = number of group members                     *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQDG                DSECT        DBDS, DB, and Recov Group block    01-DSPAP
+apqdg_GroupName         DS    CL8    Group name                         01-DSPAP
+apqdg_Memberinfo        DS    F      Offset to group member list        01-DSPAP
+apqdg_MemberLen         DS    H      Length of group member entry       01-DSPAP
+apqdg_MemberCount       DS    H      Number of group members            01-DSPAP
+                        DS    3F     Reserved                           01-DSPAP
+                        DS    CL4    Reserved                           01-DSPAP
+apqdg_Member            DSECT        List of group members              01-DSPAP
+apqdg_DBDSG             DS    OCL16  DBDS group                         01-DSPAP
+apqdg_DBDSG_DBname      DS    CL8      Database name                    01-DSPAP
+apqdg_DBDSG_DDname      DS    OCL8     DD name or                       01-DSPAP
+apqdg_DBDSG_Areaname    DS    CL8      AREA name                        01-DSPAP
+                        ORG   apqdg_DBDSG                               01-DSPAP
+apqdg_DBG               DS    OCL16  DB group                           01-DSPAP
+apqdg_DBG_DBname        DS    OCL8     Database name or                 01-DSPAP
+apqdg_DBG_Areaname      DS    CL8      AREA name                        01-DSPAP
+                        DS    CL8      Not used                         01-DSPAP
+                        ORG   apqdg_DBDSG                               01-DSPAP
+apqdg_RecovG            DS    OCL16  Recovery group                     01-DSPAP
+apqdg_RecovG_DBname     DS    CL8      Database name                    01-DSPAP
+apqdg_RecovG_Areaname   DS    CL8      AREA name, null if not Fast Path 01-DSPAP
+***********************************************************************
+*  END OF DSPAPQDG                                                    *
```

DSPAPQDS - database data set block

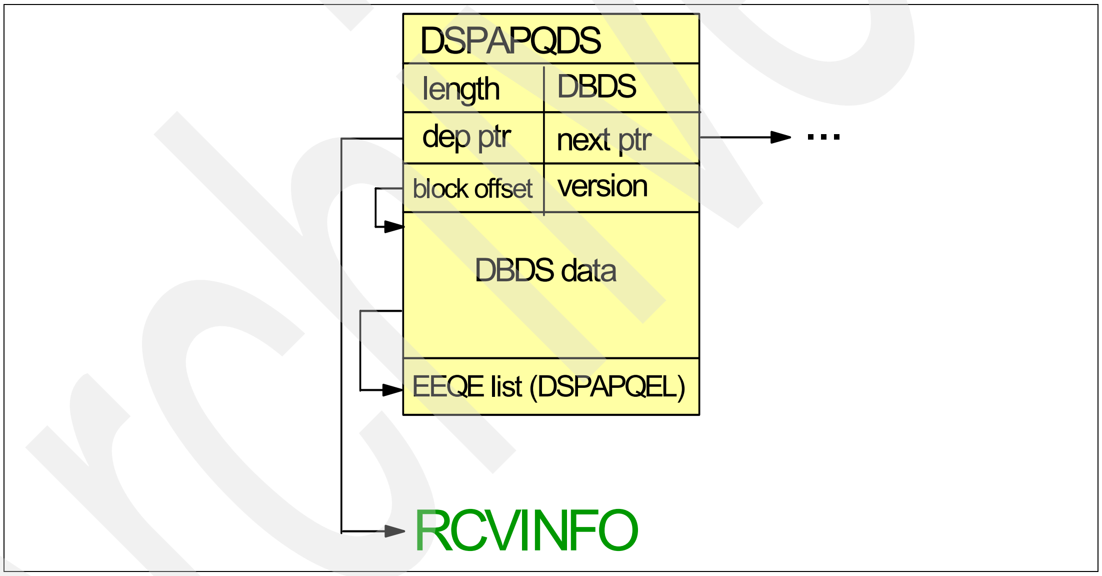


Figure A-5 DSPAPQDS database data set block

Example: A-10 DSPAPQDS - database data set block

```
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQDS - Database Data Set (DBDS) block               *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the DBDS output block returned on the             *
+*   DSPAPI FUNC=QUERY TYPE=DB call.                                   *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQDS'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'2000'  (DSPAPI_BlkType_DBDS)               *
+*     apqhd_depptr     = pointer to Recovery Information              *
+*                        block (DSPAPQRI)                             *
+*     apqhd_nextptr    = pointer to next DBDS block                   *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQDS             *
+*                                                                     *
+*   Within DSPAPQDS are the following fields which are used to        *
+*   access EEQE information for the DBDS:                             *
+*                                                                     *
+*     apqds_EEQElist   = offset from DSPAPQDS to EEQE                 *
+*                        data (DSPAPQEL)                              *
+*     apqds_EEQELength = length of each EEQE entry                    *
+*     apqds_EEQECount  = number of EEQEs                              *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQDS                DSECT        Database Data Set (DBDS) block     01-DSPAP
+apqds_DBname            DS    CL8    Database name                      01-DSPAP
+apqds_DDname            DS    CL8    DD name                            01-DSPAP
+apqds_EEQElist          DS    F      Offset to EEQE list (DSPAPQEL),    01-DSPAP
+*                                    zero if no EEQEs
+                        DS    3F     Reserved                           01-DSPAP
+apqds_DSN               DS    CL44   Data set name                      01-DSPAP
+apqds_RTPRD             DS    H      IC retention period                01-DSPAP
+apqds_DSID              DS    H      Data set ID number                 01-DSPAP
+apqds_DSSN              DS    F      Data set sequence number           01-DSPAP
+apqds_RUSID             DS    F      Recovered-to USID(trkr)            01-DSPAP
+apqds_FLAGS             DS    XL1                                       01-DSPAP
```

```
+apqds_RECYC             EQU   X'80'  REUSE image copies                 01-DSPAP
+apqds_ICREC             EQU   X'40'  Image Copy Recommended             01-DSPAP
+apqds_RCVRQ             EQU   X'20'  Receive required                   01-DSPAP
+apqds_IC                EQU   X'10'  Image Copy Needed                  01-DSPAP
+apqds_RECOV             EQU   X'08'  Recovery Needed                    01-DSPAP
+apqds_NONRV             EQU   X'04'  Non-recoverable                    01-DSPAP
+apqds_DSORG             DS    XL1    Data set organization              01-DSPAP
+apqds_VSAM              EQU   X'80'  1 = VSAM, 0 = NON-VSAM             01-DSPAP
+apqds_INDEX             EQU   X'40'  0 = Non-indexed (OSAM or ESDS),    01-DSPAP
+*                                     1 = Indexed(ISAM or KSDS)
+apqds_DBORG             DS    XL1    IMS DB organization                01-DSPAP
+                        DS    FL1    Reserved                           01-DSPAP
+apqds_GENMX             DS    H      Max number of ICs that may be      01-DSPAP
+*                                     predefined for this DBDS
+apqds_AvailIC#          DS    H      Number of available ICs for        01-DSPAP
+*                                     this DBDS
+apqds_UsedIC#           DS    H      Number of ICs used                 01-DSPAP
+apqds_EEQECount         DS    H      EEQE count                         01-DSPAP
+apqds_EEQELength        DS    H      EEQE entry length                  01-DSPAP
+apqds_FLG1              DS    XL1    Flags                              01-DSPAP
+apqds_RRGAL             EQU   X'80'  REORG since last ALLOC,            01-DSPAP
+*                                     only set if RSR-covered
+apqds_TSRAL             EQU   X'40'  TS recov since last ALLOC,         01-DSPAP
+*                                     only set if RSR-covered
+apqds_FLG2              DS    XL1    DBDS type flags                    01-DSPAP
+apqds_Part              EQU   X'80'  TYPEPART record                    01-DSPAP
+apqds_PData             EQU   X'40'   TYPEPART subtype DATA             01-DSPAP
+apqds_PILE              EQU   X'20'   TYPEPART subtype ILE              01-DSPAP
+apqds_PIndx             EQU   X'10'   TYPEPART subtype Index            01-DSPAP
+apqds_CAGrpName         DS    CL8    Change Accum group name            01-DSPAP
+apqds_JCL               DS    0CL40  GENJCL members                     01-DSPAP
+apqds_ICJCL             DS    CL8     Image copy member                 01-DSPAP
+apqds_OIJCL             DS    CL8     Online IC member                  01-DSPAP
+apqds_RCJCL             DS    CL8     Recovery member                   01-DSPAP
+apqds_DFJCL             DS    CL8     DEFLTJCL member                   01-DSPAP
+apqds_RVJCL             DS    CL8     Receive JCL member                01-DSPAP
+apqds_ODDN              DS    CL8    OLR partner DBDS                   01-DSPAP
+***********************************************************************
+*  END OF DSPAPQDS                                                    *
+***********************************************************************
```

DSPAPQEL - EEQE list

Example: A-11   DSPAPQEL - EEQE list

```
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQEL - EEQE List                                    *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the EEQE List portion of the DBDS or Area         *
+*   blocks returned on the DSPAPI FUNC=QUERY TYPE=DB call.  Refer     *
+*   to the DBDS or Area blocks - DSPAPQDS or DSPAPQAR - for more      *
+*   information on accessing the EEQE list.                           *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- ------------------------------    *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQEL                DSECT        EEQE List block                    01-DSPAP
+apqel_EEQEentry         DS    0CL13  EEQE descriptor entry              01-DSPAP
+apqel_eqefg             DS    XL1    EEQE flags                         01-DSPAP
+apqel_ertl              EQU   X'80'  Toleration error                   01-DSPAP
+apqel_errd              EQU   X'40'  Read error                         01-DSPAP
+apqel_erwt              EQU   X'20'  Write error                        01-DSPAP
+apqel_erus              EQU   X'10'  DBRC user modified                 01-DSPAP
+apqel_erpm              EQU   X'08'  DBRC permanent error               01-DSPAP
+apqel_indt              EQU   X'04'  Indoubt EEQE                       01-DSPAP
+apqel_ciind             EQU   X'02'  Index CI indicator                 01-DSPAP
+apqel_EQE               DS    XL4    EEQE                               01-DSPAP
+apqel_SSID              DS    CL8    SSID which owns the EEQE           01-DSPAP
+***********************************************************************
+*  END OF DSPAPQEL                                                    *
+***********************************************************************
```

DSPAPQFD - DEDB database block

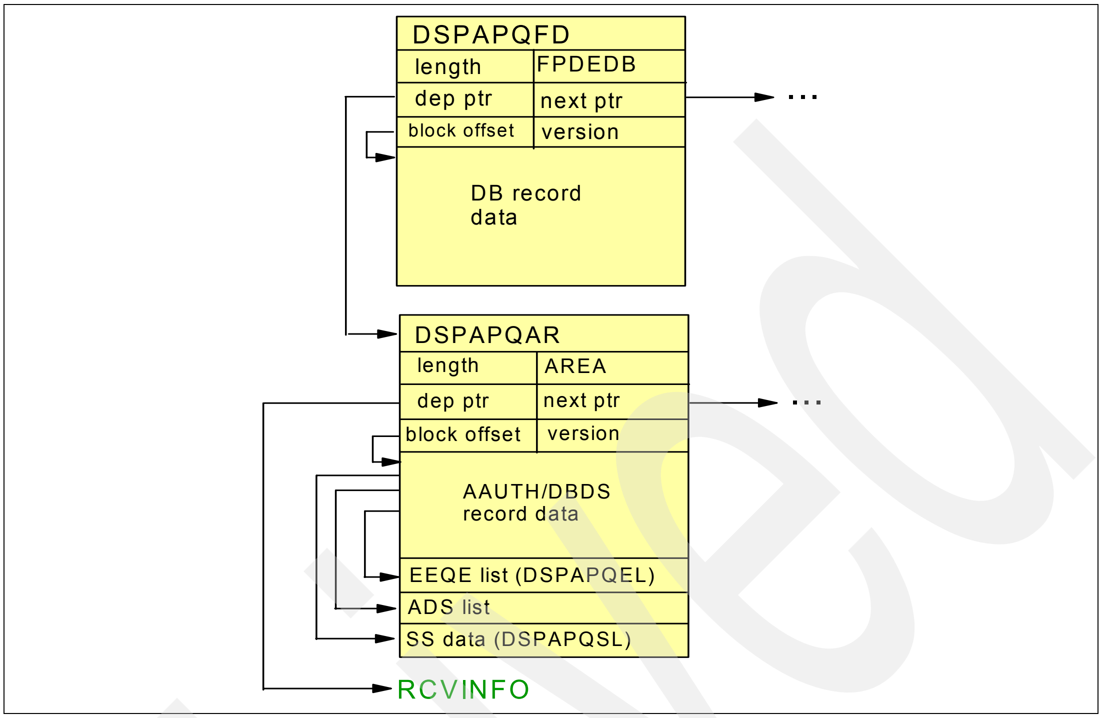


Figure A-6 Fast Path (DEDB) output

Example: A-12 DSPAPQFD - Fast Path DEDB block

```
+*******************************************************@SCPYRT**@SCPYRT*
+*                                                                      *
+*  Licensed Materials - Property of IBM                                *
+*                                                                      *
+*  5655-J38                                                            *
+*                                                                      *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved               *
+*                                                                      *
+*  US Government Users Restricted Rights - Use, duplication or         *
+*  disclosure restricted by GSA ADP Schedule contract with             *
+*  IBM Corp.                                                           *
+*                                                                      *
+****************************************************************@ECPYRT*
+*                                                                      *
+*  Name:      DSPAPQFD - Fast Path DEDB block                          *
+*                                                                      *
+*  Product:   DBRC Application Programming Interface                   *
+*                                                                      *
+*  Function:                                                           *
+*                                                                      *
+*   This macro maps the Fast Path DEDB output block returned on        *
+*   the DSPAPI FUNC=QUERY TYPE=DB call.                                *
+*                                                                      *
+*   The header of this block is mapped by DSPAPQHD which contains      *
+*   the following information:                                         *
+*                                                                      *
+*     apqhd_eyecatcher = 'DSPAPQFD'                                    *
+*     apqhd_length     = block length                                  *
+*     apqhd_blktype    = X'1802'  (DSPAPI_BlkType_FPDEDB)              *
+*     apqhd_depptr     = pointer to Fast Path Area block              *
+*                        (DSPAPQAR)                                   *
+*     apqhd_nextptr    = pointer to next database block               *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQFD             *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*    2004-03-04 1.01  PQ83837  pas  Add user-recoverable flag         *
+*                                                                     *
+***********************************************************************
+DSPAPQFD                DSECT        Fast Path DEDB block               01-DSPAP
+apqfd_DBname            DS    CL8    Database name                      01-DSPAP
+                        DS    4F     Reserved                           01-DSPAP
+apqfd_RCVCTR            DS    H      Recovery Needed Counter            01-DSPAP
+apqfd_ICCTR             DS    H      IC Needed Counter                  01-DSPAP
+apqfd_ICRecCtr          DS    H      IC Recommended Counter             01-DSPAP
+apqfd_DMBNUM            DS    H      Global DMB number                  01-DSPAP
+apqfd_EQECNT            DS    H      Total EEQE count                   01-DSPAP
+apqfd_AuthdAreas        DS    H      Number of Areas authorized         01-DSPAP
+apqfd_SHRLVL            DS    FL1    Share Level                        01-DSPAP
+apqfd_Flags             DS    XL1    Flags                              01-DSPAP
+apqfd_PAFLG             EQU   X'80'  Prohibit authorization             01-DSPAP
+apqfd_NONRV             EQU   X'40'  Non-recoverable                    01-DSPAP
+apqfd_ICNDIS            EQU   X'20'  IC needed disabled option          01-DSPAP
+apqfd_USRRV             EQU   X'10'  User-recoverable (VERSION=1.01)    01-DSPAP
+*                                                              @PQ83837
+***********************************************************************
+*  END OF DSPAPQFD                                                    *
+***********************************************************************
```

DSPAPQGG - global service group block

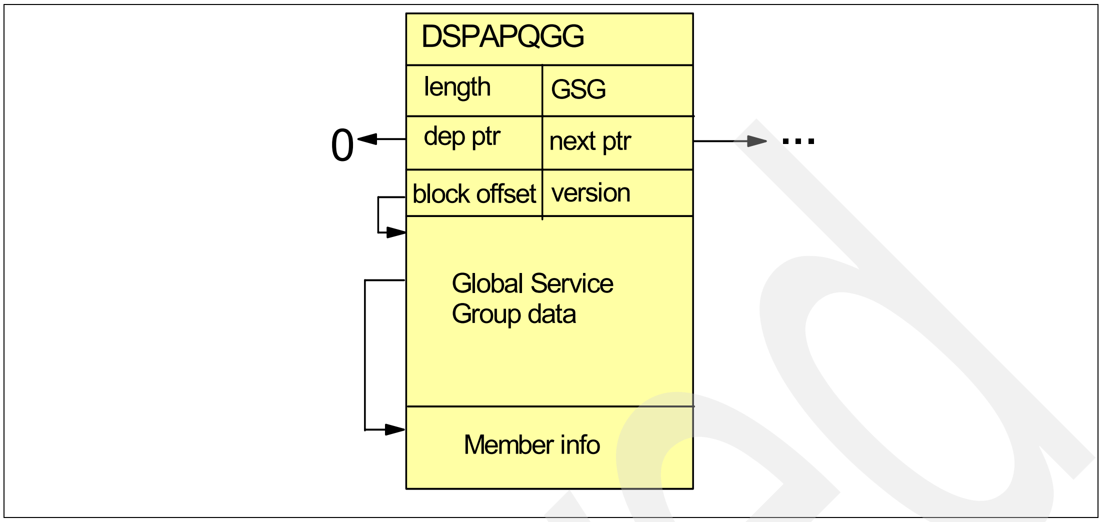


Figure A-7 Global service group

Example: A-13 DSPAPQGG - global service group block

```
+****************************************************@SCPYRT**@SCPYRT*
+*                                                                   *
+*  Licensed Materials - Property of IBM                             *
+*                                                                   *
+*  5655-J38                                                         *
+*                                                                   *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved            *
+*                                                                   *
+*  US Government Users Restricted Rights - Use, duplication or      *
+*  disclosure restricted by GSA ADP Schedule contract with          *
+*  IBM Corp.                                                        *
+*                                                                   *
+*************************************************************@ECPYRT*
+*                                                                   *
+*  Name:      DSPAPQGG - Global Service Group block                 *
+*                                                                   *
+*  Product:   DBRC Application Programming Interface                *
+*                                                                   *
+*  Function:                                                        *
+*                                                                   *
+*   This macro maps the Global Service Group output block           *
+*   returned on the DSPAPI FUNC=QUERY TYPE=GSGROUP call.            *
+*                                                                   *
+*   The header of this block is mapped by DSPAPQHD which contains   *
+*   the following information:                                      *
+*                                                                   *
+*     apqhd_eyecatcher = 'DSPAPQGG'                                 *
+*     apqhd_length     = block length                               *
+*     apqhd_blktype    = X'3A00'  (DSPAPI_BlkType_GSG)              *
+*     apqhd_depptr     = 0, no dependent block                      *
+*     apqhd_nextptr    = pointer to next group block                *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQGG           *
+*                                                                   *
+*   Within DSPAPQGG are the following fields which are used to      *
+*   access group member data:                                       *
+*                                                                   *
+*     apqgg_MemberInfo  = offset from DSPAPQGG to member list         *
+*                         (apqgg_Member)                              *
+*     apqgg_MemberLen   = length of each member entry                 *
+*     apqgg_MemberCount = number of group members                     *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQGG                DSECT        Global Service Group block         01-DSPAP
+apqgg_GroupName         DS    CL8    Group name                         01-DSPAP
+apqgg_Memberinfo        DS    F      Offset to group member list        01-DSPAP
+apqgg_MemberLen         DS    H      Length of group member entry       01-DSPAP
+apqgg_MemberCount       DS    H      Number of group members            01-DSPAP
+                        DS    2F     Reserved                           01-DSPAP
+apqgg_SDSN#             DS    F      SLDS DSN sequence number           01-DSPAP
+apqgg_PToken            DS    F      Current PRILOG token               01-DSPAP
+apqgg_MPTOK             DS    F      Min required PRILOG token          01-DSPAP
+apqgg_TTOKN             DS    F      Planned takeover token             01-DSPAP
+apqgg_LOGTime           DS    XL12   Current log start time             01-DSPAP
+apqgg_HiTime            DS    XL12   Highest time ever received         01-DSPAP
+*                                    from the active site
+apqgg_TSname            DS    CL8    Tracking subsystem ID              01-DSPAP
+apqgg_Flags             DS    XL1                                       01-DSPAP
+apqgg_ActTKO            EQU   X'80'  Active takeover in progress        01-DSPAP
+apqgg_1stSignon         EQU   X'40'  Indicates the first signon of      01-DSPAP
+*                                    an active subsystem after an
+*                                    RSR takeover
+apqgg_TrkTKO            EQU   X'20'  Tracker takeover in progress       01-DSPAP
+apqgg_RESET             EQU   X'10'  RESET.GSG issued                   01-DSPAP
+                        DS    CL7    Reserved                           01-DSPAP
+apqgg_Member            DSECT        List of group members              01-DSPAP
+apqgg_SGname            DS    CL8    Service group name                 01-DSPAP
+apqgg_SGrole            DS    CL1    Role 'A' OR 'T'                    01-DSPAP
+apqgg_SGflags           DS    XL1    Flags                              01-DSPAP
+apqgg_SGlocal           EQU   X'80'  1 = local SG                       01-DSPAP
+                        DS    CL6    Reserved                           01-DSPAP
+***********************************************************************
+*  END OF DSPAPQGG                                                    *
```

DSPAPQHB - HALDB block

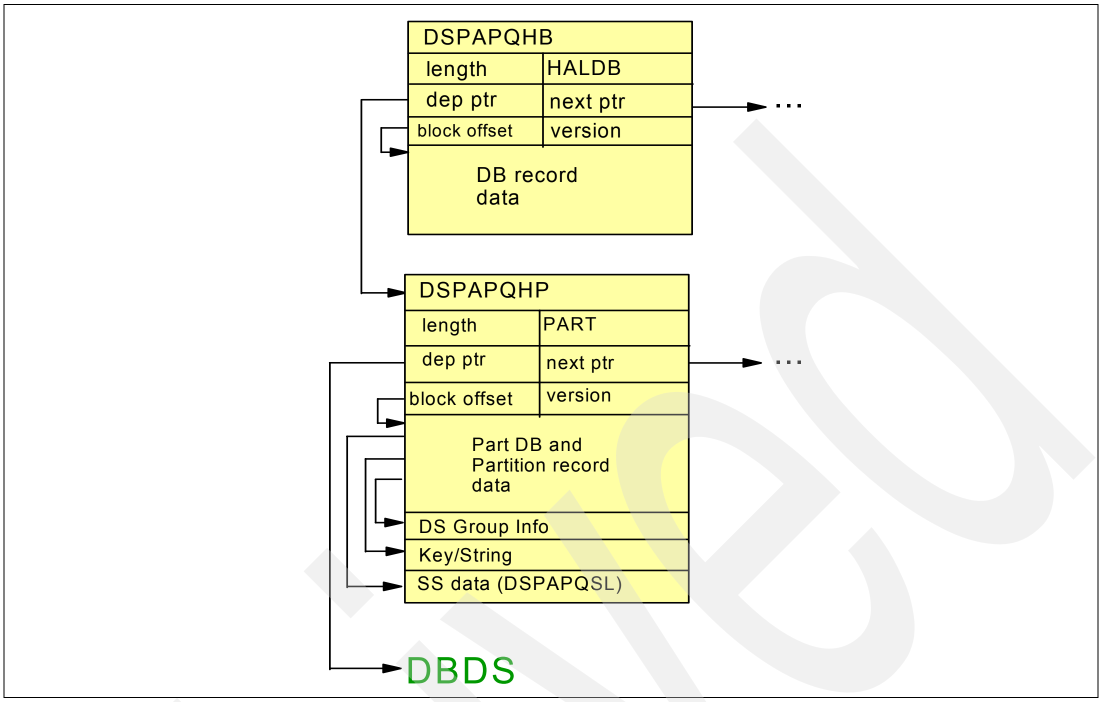


Figure A-8 HALDB block

Example: A-14 DSPAPQHB - HALDB block

```
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQHB - HALDB block                                  *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the HALDB output block returned on the            *
+*   DSPAPI FUNC=QUERY TYPE=DB call.                                   *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQHB'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'1801'  (DSPAPI_BlkType_HALDB)              *
+*     apqhd_depptr     = pointer to HALDB Partition block             *
+*                        (DSPAPQHP)                                   *
+*     apqhd_nextptr    = pointer to next database block               *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQHB             *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQHB                DSECT        HALDB block                        01-DSPAP
+apqhb_DBname            DS    CL8    HALDB name                         01-DSPAP
+                        DS    4F     Reserved                           01-DSPAP
+apqhb_flags             DS    XL1    Flags                              01-DSPAP
+apqhb_NONRV             EQU   X'80'  Non-recoverable                    01-DSPAP
+apqhb_ICNDIS            EQU   X'40'  IC needed disabled                 01-DSPAP
+apqhb_OLRcap            EQU   X'20'  HALDB is OLR capable               01-DSPAP
+apqhb_Org               DS    XL1    DB organization                    01-DSPAP
+apqhb_PSINDEX           EQU   X'80'    PSINDEX DB                       01-DSPAP
+apqhb_PHIDAM            EQU   X'40'    PHIDAM DB                        01-DSPAP
+apqhb_PHDAM             EQU   X'20'    PHDAM DB                         01-DSPAP
+apqhb_OSAM              EQU   X'10'    OSAM DB                          01-DSPAP
+apqhb_SHRLVL            DS    FL1    Share level                        01-DSPAP
+apqhb_DSGCNT            DS    FL1    # DS Group members                 01-DSPAP
+apqhb_DMBNUM            DS    H      Global DMB number                  01-DSPAP
+apqhb_PartID            DS    H      Current Partition ID               01-DSPAP
+apqhb_PART#             DS    H      Number of parts in HALDB           01-DSPAP
+apqhb_Version#          DS    H      Version number                     01-DSPAP
+apqhb_PSname            DS    CL8    Name of Part Sel Routine           01-DSPAP
+apqhb_GSGname           DS    CL8    GSG name                           01-DSPAP
+apqhb_RecovGrp          DS    CL8    Recovery Group name                01-DSPAP
+***********************************************************************
+*  END OF DSPAPQHB                                                    *
+***********************************************************************
```

DSPAPQHP - HALDB Partition block

Example: A-15   DSPAPQHP - HALDB Partition block

```
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQHP - HALDB Partition block                        *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the HALDB Partition output block returned         *
+*   on the DSPAPI FUNC=QUERY TYPE=DB call.                            *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQHP'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'1900'  (DSPAPI_BlkType_PART)               *
+*     apqhd_depptr     = pointer to DBDS block (DSPAPQDS)             *
+*     apqhd_nextptr    = pointer to next partition block              *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQHP             *
+*                                                                     *
+*   Within DSPAPQHP are the following fields which are used to        *
+*   access the partition high key or string, OSAM data set group      *
+*   information, and authorized subsystem information for the         *
+*   partition:                                                        *
+*                                                                     *
+*     apqhp_keystring     = offset from DSPAPQHP to key/string        *
+*                           (apqhp_PString)                           *
+*     apqhp_PStLn         = length of key/string                      *
+*                                                                     *
+*     apqhp_DSGInfoOffset = offset from DSPAPQHP to DSG info          *
+*                           (apqhp_DSGinfo)                           *
+*     apqhp_DSGInfoLen    = length of each DSG entry                  *
+*     apqhp_DSGcnt        = number of DSG entries                     *
+*                                                                     *
+*     apqhp_SSlist        = offset from DSPAPQHP to SS                *
+*                           data (DSPAPQSL)                           *
+*     apqhp_SSEntLen      = length of each SS entry                   *
+*     apqhp_SSNUM         = number of authorized subsystems           *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQHP                DSECT        HALDB Partition block              01-DSPAP
+apqhp_HALDBname         DS    CL8    HALDB name                         01-DSPAP
+apqhp_Partname          DS    CL8    Partition name                     01-DSPAP
+apqhp_SSlist            DS    F      Offset to SS list (DSPAPQSL),      01-DSPAP
+*                                    zero is no SS auth'd
+apqhp_keystring         DS    F      Offset to KEYSTRING,               01-DSPAP
+*                                    apqhp_PString, zero if no
+*                                    key/string
+apqhp_DSGInfoOffset     DS    F      Offset to data set group           01-DSPAP
+*                                    information
+                        DS    F      Reserved                           01-DSPAP
+apqhp_DSNbase           DS    OCL37  Base Partition DSN                 01-DSPAP
+                        DS    CL44                                      01-DSPAP
+apqhp_HDAM              DS    OCL18  PHDAM fields                       01-DSPAP
+apqhp_RMName            DS    CL8      Randomizing module name          01-DSPAP
+apqhp_RBN               DS    F        Max relative block number        01-DSPAP
+apqhp_bytes             DS    F        Max # of bytes                   01-DSPAP
+apqhp_anchr             DS    H        # of root anchor points          01-DSPAP
+apqhp_fbff              DS    FL1    Free block frequency factor        01-DSPAP
+apqhp_fspf              DS    FL1    Free space percentage factor       01-DSPAP
+apqhp_PartID            DS    H      Partition ID                       01-DSPAP
+apqhp_PStLn             DS    H      Length of Part Key/String,         01-DSPAP
+*                                    apqhp_PString
+apqhp_DSGInfoLen        DS    H      Length of each apqhp_DSGinfo       01-DSPAP
+*                                    entry
+apqhp_DSGcnt            DS    FL1    DSG count                          01-DSPAP
+apqhp_flags             DS    XL1    Flags                              01-DSPAP
+apqhp_PInit             EQU   X'80'  Partition must be initialized      01-DSPAP
+apqhp_Ordbds            EQU   X'40'  0=A-J/1=M-V DBDS active            01-DSPAP
+apqhp_Olron             EQU   X'20'  OLR active                         01-DSPAP
+apqhp_Disab             EQU   X'10'  Partition Disabled                 01-DSPAP
+apqhp_MVdbds            EQU   X'08'  1 = M-V DBDS exist                 01-DSPAP
+apqhp_OLRcap            EQU   X'04'  Partition is OLR capable           01-DSPAP
+*
+* If the Partitioned DB uses high keys, i.e. no Partition
+* Selection routine, the next two fields are used to sort
+* the partitions in key sequence.
+*
+apqhp_prev              DS    CL8    DDN of previous partition          01-DSPAP
+apqhp_next              DS    CL8    DDN of next partition              01-DSPAP
+apqhp_OLRIMS            DS    CL8    Owning IMS for OLR                 01-DSPAP
+apqhp_IRCNT             DS    H      IC receive needed counter          01-DSPAP
+apqhp_AUFLAG            DS    XL1    Authorization flags                01-DSPAP
+apqhp_BKFLG             EQU   X'80'  Backout needed                     01-DSPAP
+apqhp_PAFLG             EQU   X'40'  Prohibit authorization             01-DSPAP
+apqhp_RDFLG             EQU   X'20'  Read only SS auth                  01-DSPAP
+apqhp_NONRV             EQU   X'10'  Non-recoverable                    01-DSPAP
+apqhp_IRLMAU            DS    CL5    IRLM ID of auth SS                 01-DSPAP
+apqhp_RCVCTR            DS    H      Recovery needed count              01-DSPAP
+apqhp_ICCTR             DS    H      IC needed count                    01-DSPAP
+apqhp_SHRLVL            DS    FL1    Share level of DB                  01-DSPAP
+apqhp_HELDAU            DS    FL1    Held auth state                    01-DSPAP
+apqhp_HAUBIT            EQU   X'80'  High order bit flag                01-DSPAP
+apqhp_DMBNUM            DS    H      Global DMB number                  01-DSPAP
+apqhp_SSNUM             DS    H      # of SS auth DB                    01-DSPAP
```

```
+apqhp_SSEntLen          DS    H      Length of each SS entry            01-DSPAP
+apqhp_CACCSS            DS    FL1    Access state for chg auth          01-DSPAP
+apqhp_CANCDD            DS    FL1    Encode state for chg auth          01-DSPAP
+apqhp_CAHELD            DS    FL1    Held state for chg auth            01-DSPAP
+                        DS    CL1    Reserved                           01-DSPAP
+apqhp_EQECNT            DS    H      Total EQE count                    01-DSPAP
+apqhp_RSRFLG            DS    XL2    Flags                              01-DSPAP
+apqhp_RCVTRK            EQU   X'8000' Only recov level trackng          01-DSPAP
+apqhp_TRKSPN            EQU   X'4000' Tracking is suspended             01-DSPAP
+apqhp_PURBIT            EQU   X'2000' Suspended by time                 01-DSPAP
+apqhp_ICNDIS            EQU   X'1000' IC needed disabled option         01-DSPAP
+apqhp_NoHKey            EQU   X'0800' High key required                 01-DSPAP
+apqhp_GSGname           DS    CL8    GSG name                           01-DSPAP
+apqhp_USID              DS    F      Last alloc USID                    01-DSPAP
+apqhp_AUSID             DS    F      Last authorized USID               01-DSPAP
+apqhp_RUSID             DS    F      Last received USID                 01-DSPAP
+apqhp_HUSID             DS    F      Hardened by tracker USID           01-DSPAP
+apqhp_RNUSID            DS    F      Receive needed USID                01-DSPAP
+apqhp_ICRecCtr          DS    H      IC Recommended Counter             01-DSPAP
+apqhp_Version#          DS    H      Version number                     01-DSPAP
+apqhp_OLRActHardCtr     DS    FL1    OLR curs active count              01-DSPAP
+apqhp_OLRInactHardCtr   DS    FL1    OLR curs inact count               01-DSPAP
+apqhp_Reorg#            DS    H      Partition reorg #                  01-DSPAP
+apqhp_DSGinfo           DSECT        Data set group information         01-DSPAP
+apqhp_BLKSZ             DS    H      DS block size, OSAM only           01-DSPAP
+                        DSECT                                           01-DSPAP
+apqhp_PString           DS    OC     Partition Key/String               01-DSPAP
+***********************************************************************
+*  END OF DSPAPQHP                                                    *
+***********************************************************************
```

DSPAPQIC - image copy block

Example: A-16   DSPAPQIC - image copy block

```
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQIC - Image Copy block                             *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the Image Copy (IC) output block returned on      *
+*   the DSPAPI FUNC=QUERY TYPE=DB call.                               *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQIC'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'2D00'  (DSPAPI_BlkType_IC)                 *
+*     apqhd_depptr     = 0, no dependent blocks                       *
+*     apqhd_nextptr    = pointer to next IC block                     *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQIC             *
+*                                                                     *
+*   Within DSPAPQIC are the following fields which are used to        *
+*   access the primary/secondary IC data or the User IC data.         *
+*                                                                     *
+*   For non-User ICs (Batch, SMS, etc.):                              *
+*                                                                     *
+*     apqic_OFF1  = offset from DSPAPQIC to IC1 data (apqic_IC12)     *
+*     apqic_OFF2  = offset from DSPAPQIC to IC2 data (apqic_IC12),    *
+*                   if applicable                                     *
+*     apqic_Len12 = length of IC1/IC2 data                            *
+*                                                                     *
+*     Within the IC1/IC2 data block (apqic_IC12) are the              *
+*     following fields which are used to access the volume            *
+*     information for the IC data set:                                *
+*                                                                     *
+*       apqic_VolListOffset = offset from apqic_IC12 to volume        *
+*                             list data (apqic_VOLS)                  *
+*       apqic_VolListLen    = length of each volume list entry       *
+*       apqic_VOLCT         = number of volumes                       *
+*                                                                     *
+*   For User ICs (apqic_UserIC):                                      *
+*                                                                     *
+*     apqic_OFFU = offset from DSPAPQIC to User IC data               *
+*                (apqic_User)                                         *
+*     apqic_LenU = length of User IC data                             *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQIC                DSECT        Image Copy block                   01-DSPAP
+apqic_DBname            DS    CL8    Database name                      01-DSPAP
+apqic_DDname            DS    0CL8   DD name or                         01-DSPAP
+apqic_Areaname          DS    CL8    Area name                          01-DSPAP
+apqic_starttime         DS    XL12   IC start time, packed decimal      01-DSPAP
+apqic_stoptime          DS    XL12   IC stop time, packed decimal       01-DSPAP
```

```
+apqic_TYPE              DS    XL1    Image Copy type                    01-DSPAP
+apqic_BATCH             EQU   X'80'  Batch                              01-DSPAP
+apqic_CIC               EQU   X'40'  Concurrent                         01-DSPAP
+apqic_UserIC            EQU   X'20'  User Image Copy                    01-DSPAP
+apqic_online            EQU   X'10'  Online                             01-DSPAP
+apqic_SMSIC             EQU   X'08'  SMS IC w/ DB exclusive             01-DSPAP
+apqic_SMSCC             EQU   X'04'  SMS IC w/ DB shared                01-DSPAP
+apqic_status            DS    XL1    IC status flags                    01-DSPAP
+apqic_AVAIL             EQU   X'80'  Available IC                       01-DSPAP
+apqic_IC1               EQU   X'40'  Image Copy 1 exists                01-DSPAP
+apqic_IC2               EQU   X'20'  Image Copy 2 exists                01-DSPAP
+apqic_ERR1              EQU   X'10'  Error on image 1                   01-DSPAP
+apqic_ERR2              EQU   X'08'  Error on image 2                   01-DSPAP
+apqic_EMP2              EQU   X'04'  Image 2 defined and unused         01-DSPAP
+apqic_FLAGS             DS    XL1                                       01-DSPAP
+apqic_HSINP             EQU   X'80'  HSSP CIC in progress               01-DSPAP
+apqic_CAT               EQU   X'40'  Catalogued IC (HSSP)               01-DSPAP
+                        DS    CL1    Reserved                           01-DSPAP
+apqic_OFF1              DS    0H     Offset to image 1 data             01-DSPAP
+apqic_OFFU              DS    H      Offset to user IC data             01-DSPAP
+apqic_OFF2              DS    H      Offset to image 2 data             01-DSPAP
+apqic_CNT12             DS    F      Record count                       01-DSPAP
+apqic_USID              DS    F      Update set ID                      01-DSPAP
+apqic_Len12             DS    0H     Length of image 1/2 data           01-DSPAP
+apqic_LenU              DS    H      Length of user IC data             01-DSPAP
+                        DS    CL6    Reserved                           01-DSPAP
+apqic_IC12              DSECT        Data for image 1 or 2              01-DSPAP
+apqic_DSN12             DS    CL44   Data set name                      01-DSPAP
+apqic_FILE              DS    H      File sequence number               01-DSPAP
+apqic_RUT12             DS    CL8    Unit device type                   01-DSPAP
+apqic_VOLCT             DS    H      # of volumes predefined            01-DSPAP
+apqic_VOLUS             DS    H      # of volumes used                  01-DSPAP
+apqic_VolListLen        DS    H      Length of each volume list         01-DSPAP
+*                                    entry in apqic_VOLS
+apqic_VolListOffset     DS    F      Offset to volume list              01-DSPAP
+                        DSECT                                           01-DSPAP
+apqic_VOLS              DS    CL6    List of VOLSERs                    01-DSPAP
+apqic_User              DSECT        Data for user IC                   01-DSPAP
+apqic_UData             DS    CL80   User supplied data                 01-DSPAP
+***********************************************************************
+*  END OF DSPAPQIC                                                    *
+***********************************************************************
```

DSPAPQLA - log allocation block

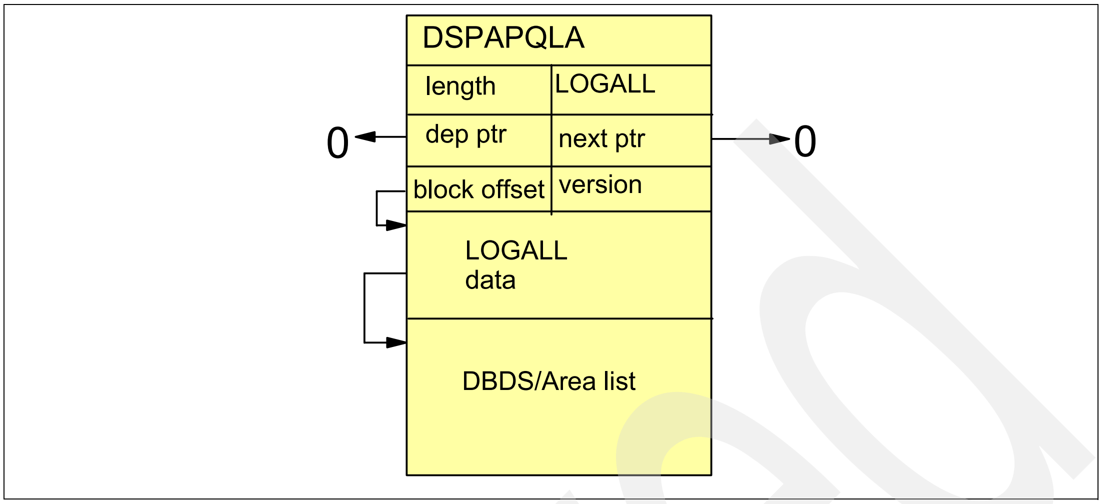


Figure A-9 LOGALL output

Example: A-17 DSPAPQLA - log allocation block

```
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQLA - Log Allocation block                         *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the Log Allocation output block returned on       *
+*   the DSPAPI FUNC=QUERY TYPE=LOG call.                              *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQLA'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'0700'  (DSPAPI_BlkType_LOGALL)             *
+*     apqhd_depptr     = 0, no dependent blocks                       *
+*     apqhd_nextptr    = 0, no next block                             *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQLA             *
+*                                                                     *
+*   Within DSPAPQLA are the following fields which are used to        *
+*   access the list of DBDSs allocated on this log:                   *
+*                                                                     *
+*     apqla_DBDSAreaInfo  = offset from DSPAPQLA to list of           *
+*                           allocated DBDSs and Areas                  *
+*                           (apqla_DBDSArea)                           *
+*     apqla_DBDSAreaCount = number of DBDS/Area entries                *
+*     apqla_DBDSAreaLen   = length of each DBDS/Area entry             *
+*                                                                      *
+*  Notes:                                                              *
+*                                                                      *
+*   Dependencies: None.                                                *
+*                                                                      *
+*   Restrictions: None.                                                *
+*                                                                      *
+*  Parameters: None.                                                   *
+*                                                                      *
+*  External References:                                                *
+*    Macros:  None                                                     *
+*                                                                      *
+*    Modules: None                                                     *
+*                                                                      *
+*  Change Activity:                                                    *
+*                                                                      *
+*    Date       Ver   PTM/APAR Init Description                        *
+*    ---------- ----- -------- ---- -----------------------------      *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                      *
+*                                                                      *
+***********************************************************************
+DSPAPQLA                DSECT        Log Allocation block               01-DSPAP
+apqla_DBDSAreainfo      DS    F      Offset to allocated DBDS/Area list 01-DSPAP
+                        DS    3F     Reserved                           01-DSPAP
+apqla_PRILOGtime        DS    XL12   PRILOG time                        01-DSPAP
+apqla_Flags             DS    XL1    Flags                              01-DSPAP
+apqla_NonRegd           EQU   X'80'  Non-registered DB updated          01-DSPAP
+apqla_DBDSAreaCount     DS    XL3    Number of DBDS/Areas               01-DSPAP
+*                                    allocated on this log
+apqla_DBDSAreaLen       DS    F      Length of DBDS/Area entry          01-DSPAP
+apqla_EarliestALLOC     DS    XL12   Earliest ALLOC time for this log   01-DSPAP
+apqla_DBDSArea          DSECT        List of allocated DBDSs and Areas  01-DSPAP
+apqla_DBname            DS    CL8    Database name                      01-DSPAP
+apqla_DDname            DS    0CL8   DD name or                         01-DSPAP
+apqla_Areaname          DS    CL8    AREA name                          01-DSPAP
+apqla_FirstALLOC        DS    XL12   Earliest ALLOC time for this       01-DSPAP
+*                                    DBDS/Area on this log
+apqla_ALLNO             DS    H      Number of ALLOCs for this          01-DSPAP
+*                                    DBDS/Area on this log
+                        DS    CL2    Reserved                           01-DSPAP
+***********************************************************************
+*  END OF DSPAPQLA                                                     *
```

DSPAPQLG - RLDS/SLDS block

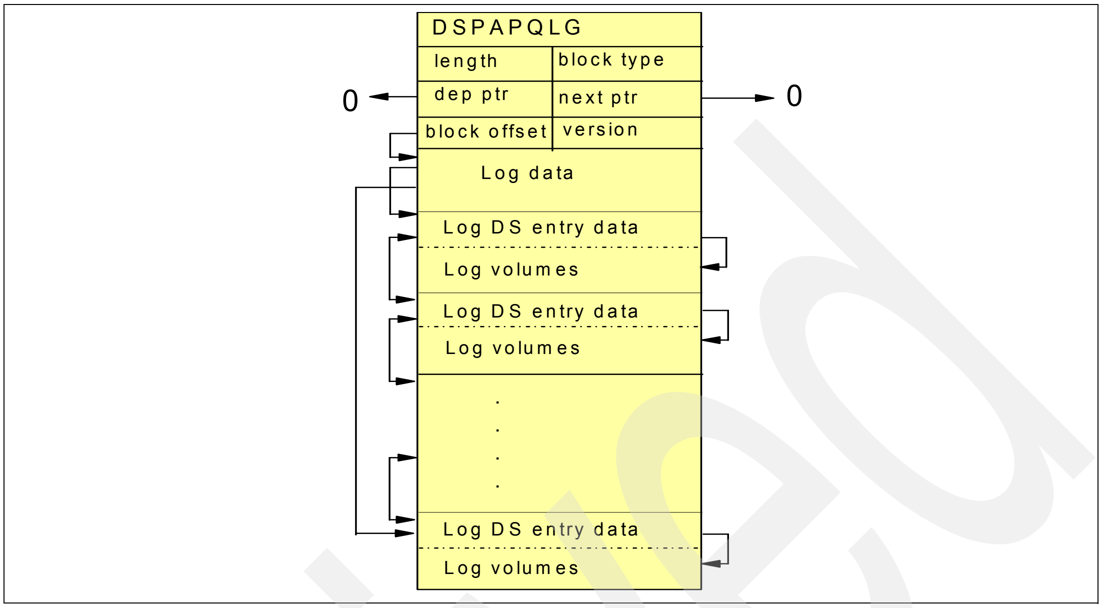


Figure A-10 Output block for the log entries

Example: A-18 DSPAPQLG - RLDS/SLDS block

```
+*******************************************************@SCPYRT**@SCPYRT*
+*                                                                      *
+*  Licensed Materials - Property of IBM                                *
+*                                                                      *
+*  5655-J38                                                            *
+*                                                                      *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved               *
+*                                                                      *
+*  US Government Users Restricted Rights - Use, duplication or         *
+*  disclosure restricted by GSA ADP Schedule contract with             *
+*  IBM Corp.                                                           *
+*                                                                      *
+****************************************************************@ECPYRT*
+*                                                                      *
+*  Name:      DSPAPQLG - RLDS/SLDS block                               *
+*                                                                      *
+*  Product:   DBRC Application Programming Interface                   *
+*                                                                      *
+*  Function:                                                           *
+*                                                                      *
+*   This macro maps the RLDS/SLDS output block returned on the         *
+*   DSPAPI FUNC=QUERY TYPE=LOG call.                                   *
+*                                                                      *
+*   The header of this block is mapped by DSPAPQHD which contains      *
+*   the following information:                                         *
+*                                                                      *
+*     apqhd_eyecatcher = 'DSPAPQLG'                                    *
+*     apqhd_length     = block length                                  *
+*     apqhd_blktype    = X'0500'  (DSPAPI_BlkType_PRILOG)              *
+*                        X'0900'  (DSPAPI_BlkType_SECLOG)              *
+*                        X'4300'  (DSPAPI_BlkType_PRISLDS)             *
+*                        X'4700'  (DSPAPI_BlkType_SECSLDS)             *
+*                        X'4400'  (DSPAPI_BlkType_PRITSLDS)            *
+*                        X'4800'  (DSPAPI_BlkType_SECTSLDS)         *
+*     apqhd_depptr     = 0, no dependent block                        *
+*     apqhd_nextptr    = 0, no next block                             *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQLG             *
+*                                                                     *
+*   Within DSPAPQLG are the following fields which are used to       *
+*   access the first and the last data set entries:                  *
+*                                                                     *
+*     apqlg_FirstLogDS = offset from DSPAPQLG to first log data       *
+*                        set entry (apqlg_DS_Entry)                   *
+*                                                                     *
+*     apqlg_LastLogDS  = offset from DSPAPQLG to last log data        *
+*                        set entry                                    *
+*                                                                     *
+*   Within each DS entry (apqlg_DS_Entry) are the following          *
+*   fields which are used to access the previous and next DS         *
+*   entries as well as the volumes associated with the DS:           *
+*                                                                     *
+*     apqlg_DS_Next = offset from DSPAPQLG to next DS entry           *
+*     apqlg_DS_Prev = offset from DSPAPQLG to previous DS entry       *
+*                                                                     *
+*     apqlg_DS_Volinfo = offset from apqlg_DS_Entry to volume         *
+*                        info (apqlg_DSVolume) of the first DS        *
+*                        volume                                       *
+*                                                                     *
+*       Within apqlg_DSVolume:                                        *
+*                                                                     *
+*        apqlg_DSVol_Next = offset from apqlg_DS_Entry to the         *
+*                           next DS volume                            *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQLG                DSECT        RLDS/SLDS block                     01-DSPAP
+apqlg_FirstLogDS        DS    F      Offset to first log DS entry       01-DSPAP
+apqlg_LastLogDS         DS    F      Offset to last  log DS entry       01-DSPAP
+                        DS    2F     Reserved                           01-DSPAP
+apqlg_SSID              DS    CL8    Subsystem ID                       01-DSPAP
+apqlg_StartTime         DS    XL12   Log start time                     01-DSPAP
+apqlg_EndTime           DS    XL12   Log end time                       01-DSPAP
+apqlg_DSNcount          DS    F      Number of data sets                01-DSPAP
+apqlg_RELVL             DS    FL1    Log Release Level                  01-DSPAP
+apqlg_Flags1            DS    XL1    Flags                              01-DSPAP
```

```
+apqlg_Online            EQU   X'80'  Online log                         01-DSPAP
+apqlg_LSTAR             EQU   X'40'  Last OLDS has been archived        01-DSPAP
+apqlg_LSTNA             EQU   X'20'  Last OLDS has not been archived    01-DSPAP
+apqlg_GAP               EQU   X'10'  There is a gap in this log         01-DSPAP
+apqlg_PRGAP             EQU   X'08'  There is a gap in a prev log       01-DSPAP
+apqlg_Flags2            DS    XL1    Flags                              01-DSPAP
+apqlg_TRKNG             EQU   X'80'  Tracking log data set              01-DSPAP
+apqlg_NTERM             EQU   X'40'  IMS subsystem has terminated       01-DSPAP
+*                                     normally
+apqlg_BKLOG             EQU   X'20'  Batch backout log                  01-DSPAP
+                        DS    CL1    Reserved                           01-DSPAP
+apqlg_FirstLRID         DS    XL8    Id of first log record             01-DSPAP
+apqlg_PToken            DS    F      PRILOG token                       01-DSPAP
+apqlg_GSGname           DS    CL8    GSG name                           01-DSPAP
+apqlg_CHKPT0            DS    XL12   Checkpoint 0 time                  01-DSPAP
+                        DS    CL8    Reserved                           01-DSPAP
+***********************************************************************
+*  The following structure maps the data set entries.  There is      *
+*  one such entry for each data set.      Each data set entry         *
+*  contains the offset to the preceeding and succeeding entries      *
+*  as well as the offset to the first of a chain of DS volume         *
+*  entries.  Field apqlg_DS_Next is the offset of the succeeding     *
+*  entry and apqlg_DS_Prev is the offset of the preceeding entry.    *
+*  Field apqlg_DS_VolInfo is the offset of the first volume entry.   *
+*  For the first data set entry apqlg_DS_Prev will be zero.          *
+*  Similarly, apqlg_DS_Next will be zero for the the last data       *
+*  set entry.                                                         *
+*                                                                     *
+***********************************************************************
+apqlg_DS_entry          DSECT        Data set entry                     01-DSPAP
+apqlg_DS_Next           DS    F      Offset to next log DS entry        01-DSPAP
+apqlg_DS_Prev           DS    F      Offset to prev log DS entry        01-DSPAP
+apqlg_DS_VolInfo        DS    F      Offset to DS volume info           01-DSPAP
+apqlg_DS_DSname         DS    CL44   Data set name                      01-DSPAP
+apqlg_DS_StartTime      DS    XL12   DS start time                      01-DSPAP
+apqlg_DS_EndTime        DS    XL12   DS end time                        01-DSPAP
+apqlg_DS_Flags1         DS    XL1    Flags                              01-DSPAP
+apqlg_DS_ERR            EQU   X'80'  I/O Error                          01-DSPAP
+apqlg_DS_DUMMY          EQU   X'40'  Log compressed, 1st DS dummy       01-DSPAP
+*
+* The next six bits apply only to SLDSs.
+*
+apqlg_DS_RSTBG          EQU   X'20'  Restart begin                      01-DSPAP
+apqlg_DS_RSTEN          EQU   X'10'  Restart end                        01-DSPAP
+apqlg_DS_COLD           EQU   X'08'  Cold start                         01-DSPAP
+apqlg_DS_NOBMP          EQU   X'04'  ERE NOBMP                          01-DSPAP
+apqlg_DS_SAVER          EQU   X'02'  Backout UORs saved                 01-DSPAP
+apqlg_DS_NOID           EQU   X'01'  Backouts not identified            01-DSPAP
+apqlg_DS_Flags2         DS    XL1    Flags                              01-DSPAP
+apqlg_DS_TrkArch        EQU   X'80'  Tracking log DS archived           01-DSPAP
+apqlg_DS_TrkFEOV        EQU   X'40'  Tracking log closed FEOV           01-DSPAP
+apqlg_DS_DFLG3          DS    CL2    Reserved                           01-DSPAP
+apqlg_DS_FLRID          DS    XL8    First log record ID                01-DSPAP
+apqlg_DS_LLRID          DS    XL8    Last log record ID                 01-DSPAP
+apqlg_DS_LastBlkSeqNo   DS    F      Last block seq number              01-DSPAP
+apqlg_DS_UnitType       DS    CL8    Unit type name                     01-DSPAP
+apqlg_DS_FileSeq        DS    H      File sequence no                   01-DSPAP
+apqlg_DS_VolCount       DS    H      Number of volumes                  01-DSPAP
+apqlg_DS_CkptCount      DS    FL1    Number of chkpts on DSN            01-DSPAP
+apqlg_DS_ChkptTypes     DS    XL1    CHKPT types.                       01-DSPAP
```

```
+apqlg_DS_SIMPL          EQU   X'80'  SIMPLE checkpoint                  01-DSPAP
+apqlg_DS_SNAPQ          EQU   X'40'  SNAPQ checkpoint                   01-DSPAP
+apqlg_DS_DUMPQ          EQU   X'20'  DUMPQ checkpoint                   01-DSPAP
+apqlg_DS_PURGE          EQU   X'10'  PURGE checkpoint                   01-DSPAP
+apqlg_DS_FREEZ          EQU   X'08'  FREEZE checkpoint                  01-DSPAP
+                        DS    H      Reserved                           01-DSPAP
+apqlg_DSVolume          DSECT        Volume information                 01-DSPAP
+apqlg_DSVol_Next        DS    F      Offset to next volume              01-DSPAP
+apqlg_DSVol_SER         DS    CL6    VOLSER                             01-DSPAP
+apqlg_DSVol_CkptCt      DS    FL1    Volume chkpt count                 01-DSPAP
+                        DS    CL1    Reserved                           01-DSPAP
+apqlg_DSVol_EndTime     DS    XL12   Volume end time                    01-DSPAP
+apqlg_DSVol_CPTID       DS    XL12   Checkpoint ID                      01-DSPAP
+apqlg_DSVol_LockSN      DS    XL6    Lock Sequence Number               01-DSPAP
+                        DS    CL6    Reserved                           01-DSPAP
+***********************************************************************
+*  END OF DSPAPQLG                                                    *
+***********************************************************************
```

DSPAPQLI - log information block

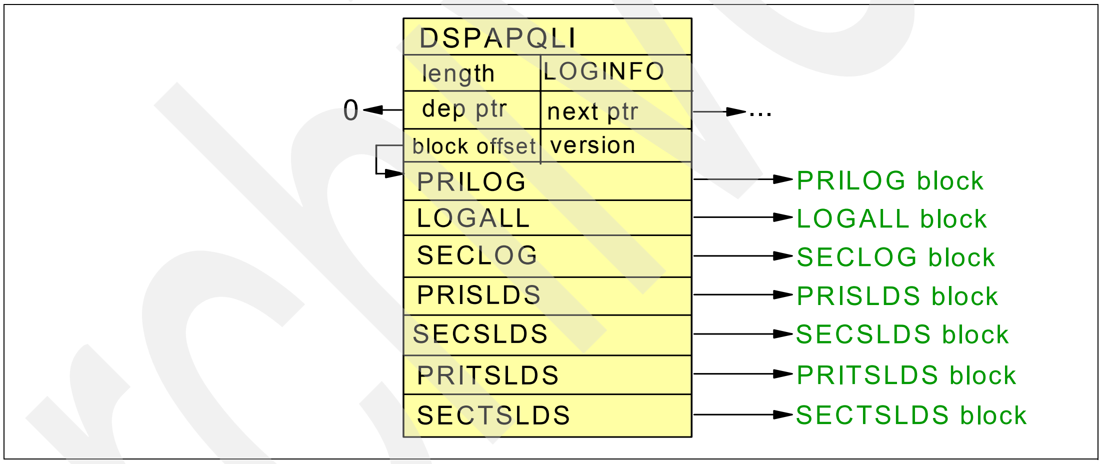


Figure A-11 DSPAPQLI log information block

Example: A-19 DSPAPQLI - log information block

```
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQLI - Log Information block                        *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the Log Information output block returned on     *
+*   the DSPAPI FUNC=QUERY TYPE=LOG call.                              *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQLI'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'0501'  (DSPAPI_BlkType_LOGINFO)            *
+*     apqhd_depptr     = 0, no dependent blocks                       *
+*     apqhd_nextptr    = pointer to the LOGINFO block for the         *
+*                        next group of log blocks                     *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQLI             *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- ------------------------------    *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQLI                DSECT        Log Information block              01-DSPAP
+apqli_SSID              DS    CL8    Log SSID                           01-DSPAP
+apqli_StartTime         DS    XL12   Log start time                     01-DSPAP
+apqli_PRILOGptr         DS    A      ptr to PRILOG block                01-DSPAP
+apqli_LOGALLptr         DS    A      ptr to LOGALL block                01-DSPAP
+apqli_SECLOGptr         DS    A      ptr to SECLOG block                01-DSPAP
+apqli_PRISLDSptr        DS    A      ptr to PRISLDS block               01-DSPAP
+apqli_SECSLDSptr        DS    A      ptr to SECSLDS block               01-DSPAP
+apqli_PRITSLDSptr       DS    A      ptr to PRITSLDS block              01-DSPAP
+apqli_SECTSLDSptr       DS    A      ptr to SECTSLDS block              01-DSPAP
+***********************************************************************
+*  END OF DSPAPQLI                                                    *
```

DSPAPQNF - not found block

Example: A-20   DSPAPQNF - not found block

```
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or       *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQNF - Not Found block                              *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the Not Found output block returned on the       *
+*   DSPAPI FUNC=QUERY TYPE=DB call.                                   *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains    *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQNF'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'18FF'  (DSPAPI_BlkType_DBNotFound)         *
+*     apqhd_depptr     = 0, no dependent blocks                       *
+*     apqhd_nextptr    = pointer to the next DB output block          *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQNF             *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQNF                DSECT        Not Found block          @KZ30205  01-DSPAP
+apqnf_DBname            DS    CL8    Database name                      01-DSPAP
+***********************************************************************
+*  END OF DSPAPQNF                                                    *
```

DSPAPQOL - online log data set block

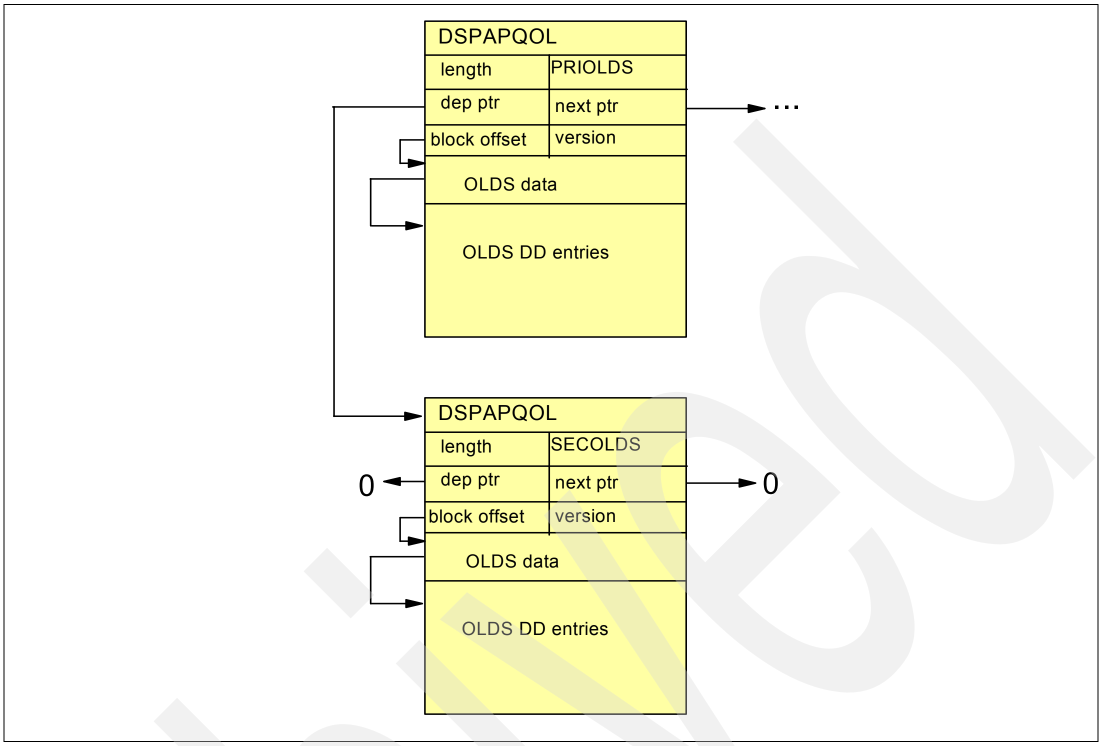


Figure A-12 OLDS output

Example: A-21 DSPAPQOL - online log data set block

```
+*******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+*****************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQOL - Online Log Data Set block                    *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the Online Log Data Set output block returned     *
+*   on the DSPAPI FUNC=QUERY TYPE=OLDS call.                          *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                       *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQOL'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'5300'  (DSPAPI_BlkType_PRIOLDS)            *
+*                        X'5700'  (DSPAPI_BlkType_SECOLDS)            *
+*     apqhd_depptr     = If PRIOLDS block, this is the pointer to     *
+*                        the SECOLDS block, if one exists.  If        *
+*                        SECOLDS block, this is zero.                 *
+*     apqhd_nextptr    = pointer to next OLDS block                   *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQOL             *
+*                                                                     *
+*   Within DSPAPQOL are the following fields which are used to        *
+*   access online log data set information:                           *
+*                                                                     *
+*     apqol_OLDSInfo   = offset from DSPAPQOL to OLDS entries         *
+*                       (apqla_OLDSentry)                             *
+*     apqol_OLDSCount  = number of OLDS entries                       *
+*     apqol_OLDSLen    = length of each OLDS entry                    *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQOL                DSECT        Online Log Dataset block           01-DSPAP
+apqol_OLDSinfo          DS    F      ! Offset to OLDS list              01-DSPAP
+                        DS    3F     ! Reserved                         01-DSPAP
+apqol_SSID              DS    CL8    ! Subsystem ID                     01-DSPAP
+apqol_OLDSlen           DS    H      ! Length of OLDS entry             01-DSPAP
+apqol_OLDScount         DS    H      ! Number of OLDS entries           01-DSPAP
+apqol_CHKPT0            DS    XL12   ! Checkpoint 0 time                01-DSPAP
+                        DS    CL8    ! Reserved                         01-DSPAP
+apqol_OLDSentry         DSECT        OLDS entry                         01-DSPAP
+apqol_DDname            DS    CL8    OLDS DD name                       01-DSPAP
+apqol_DSNAM             DS    CL44   OLDS data set name                 01-DSPAP
+apqol_OpenTime          DS    XL12   OLDS open time                     01-DSPAP
+apqol_CloseTime         DS    XL12   Close time                         01-DSPAP
+apqol_PRILOGtime        DS    XL12   Start time of associated PRILOG    01-DSPAP
+apqol_FLSN              DS    XL8    LSN of first record in OLDS        01-DSPAP
+apqol_LLSN              DS    XL8    LSN of last record in OLDS         01-DSPAP
+apqol_Flag1             DS    XL1                                       01-DSPAP
+apqol_RSTBG             EQU   X'80'  Restart begin                      01-DSPAP
+apqol_RSTEN             EQU   X'40'  Restart end                        01-DSPAP
+apqol_COLD              EQU   X'20'  Cold start                         01-DSPAP
+apqol_NOBMP             EQU   X'10'  ERE NOBMP                          01-DSPAP
+apqol_SAVER             EQU   X'08'  Backout UORs saved                 01-DSPAP
+apqol_NOID              EQU   X'04'  Backouts not identified            01-DSPAP
+apqol_TRKNG             EQU   X'02'  OLDS created by tracking SS        01-DSPAP
```

```
+apqol_Flag2             DS    XL1    OLDS flags                         01-DSPAP
+apqol_STAT              EQU   X'F0'  OLDS status                        01-DSPAP
+apqol_INUSE             EQU   X'80'  OLDS is in use                     01-DSPAP
+apqol_ARNED             EQU   X'40'  Archive needed                     01-DSPAP
+apqol_ARSCH             EQU   X'20'  Archive scheduled (GENJCL)         01-DSPAP
+apqol_ARSTD             EQU   X'10'  Archive job started                01-DSPAP
+apqol_CLERR             EQU   X'08'  Close error on OLDS                01-DSPAP
+apqol_FEOV              EQU   X'04'  Force EOV at archive               01-DSPAP
+apqol_DUMMY             EQU   X'02'  OLDS not used due to I/O err       01-DSPAP
+apqol_PRVCE             EQU   X'01'  Close error on previous OLDS       01-DSPAP
+apqol_RELVL             DS    FL1    Log release level                  01-DSPAP
+apqol_GAVER             DS    FL1    GENJCL.ARCHIVE version             01-DSPAP
+apqol_BlockSeqNo        DS    XL4    Block sequence number              01-DSPAP
+apqol_ARJOB             DS    CL8    Name of the archive job            01-DSPAP
+*                                     Generated by GENJCL.ARCHIVE
+apqol_LockSeqNo         DS    XL6    Lock sequence number               01-DSPAP
+                        DS    CL2    Reserved                           01-DSPAP
+***********************************************************************
+*  END OF DSPAPQOL                                                    *
```

DSPAPQRC - RECON status block

Example: A-22   DSPAPQRC - RECON status block

```
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQRC - RECON Status block                           *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the output area returned on the DSPAPI            *
+*   FUNC=QUERY TYPE=RECON call.  This block describes the overall     *
+*   status of the RECON as well as the status of each RECON data     *
+*   set.                                                              *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQRC'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'0100'  (DSPAPI_BlkType_RECON)              *
+*     apqhd_depptr     = 0, no dependent blocks                       *
+*     apqhd_nextptr    = 0, no next blocks                            *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQRC             *
+*                                                                     *
+*   DSPAPQRC maps the information describing the overall status       *
+*   of the RECON.  Within this block are the following fields         *
+*   which are used to access the information for the individual       *
+*   RECON data sets (mapped by apqrc_RECONDS_info):                   *
+*                                                                     *
+*     apqrc_RECONcount   = number of RECON data set elements          *
+*     apqrc_RECONinfo    = offset from DSPAPQRC to information        *
+*                          about the first RECON in the list,         *
+*                          apqrc_RECONDS_info.                        *
+*     apqrc_RECONinfolen = length of each RECON data set element      *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQRC                DSECT        RECON status header                01-DSPAP
+apqrc_DATA              DS    CL44   Initialized with "RECOVERY         01-DSPAP
+*                                    CONTROL DATASET"
+apqrc_RECONinfo         DS    F      Offset to RECON data set info      01-DSPAP
+                        DS    2F     Reserved                           01-DSPAP
+apqrc_RECONinfolen      DS    H      Length of each RECON dataset       01-DSPAP
+*                                    info element
+apqrc_RECONcount        DS    XL1    # of RECON dataset elements        01-DSPAP
+                        DS    CL1    Reserved                           01-DSPAP
+apqrc_flags             DS    XL1    Process flags...                   01-DSPAP
+apqrc_NOCHK             EQU   X'80'  1= NOCHECK log tape dsn check      01-DSPAP
+apqrc_CHK17             EQU   X'40'  1= CHECK17 log tape dsn check      01-DSPAP
+apqrc_CHK44             EQU   X'20'  1= CHECK44 log tape dsn check      01-DSPAP
+apqrc_LSTLG             EQU   X'10'  1= list log DSN                    01-DSPAP
+apqrc_INUPG             EQU   X'08'  Upgrade in progress.               01-DSPAP
+apqrc_flag2             DS    XL1    More flags...                      01-DSPAP
+apqrc_FORCE             EQU   X'80'  1 = FORCER, 0 = NOFORCER           01-DSPAP
+apqrc_CATDS             EQU   X'40'  1=CA|IC|LOGS cataloged             01-DSPAP
+apqrc_TRACE             EQU   X'20'  1 = ext. GTF trace on              01-DSPAP
+apqrc_CmdSAF            EQU   X'10'  SAF enabled                        01-DSPAP
+apqrc_CmdExit           EQU   X'08'  Cmd auth exit enabled              01-DSPAP
+                        DS    CL2    Reserved                           01-DSPAP
+apqrc_CLEAN             DS    0CL136 Fields needed for cleanup          01-DSPAP
+apqrc_CSET              DS    F      0 = updates not in progress,       01-DSPAP
+*                                    >0 = update in progress
+apqrc_TYPE              DS    F      Type of update                     01-DSPAP
+apqrc_OKEY              DS    XL32   Key of original record             01-DSPAP
+apqrc_BKEY              DS    XL32   Key of base record that is in      01-DSPAP
+*                                    the process of being changed
+apqrc_NKEY              DS    XL32   Key of new record                  01-DSPAP
```

```
+apqrc_DBID              DS    0CL16  DBID of DBDS                       01-DSPAP
+apqrc_DBD               DS    CL8    DBD name                           01-DSPAP
+apqrc_DDN               DS    CL8    DD name                            01-DSPAP
+apqrc_CAGRP             DS    CL8    CA group name                      01-DSPAP
+apqrc_DDNEW             DS    CL8    New DBDS DD name                   01-DSPAP
+apqrc_DMBNO             DS    H      DMB sequence number                01-DSPAP
+apqrc_LastReusedDMB#    DS    H      Last reused DMB number, valid      01-DSPAP
+*                                    only when apqrc_DMBNO
+*                                    is 32767
+apqrc_InitToken         DS    XL7    Recon init. token                  01-DSPAP
+apqrc_CmdHLQ            DS    CL8    Cmd auth high lvl qual             01-DSPAP
+apqrc_MVERS             DS    FL1    Minimum IMS version                01-DSPAP
+apqrc_NWFLG             DS    XL1    Fields needed for RECON I/O        01-DSPAP
+*                                    error(s)
+apqrc_NEW               EQU   X'80'  1=STARTNEW, 0=NONEW                01-DSPAP
+                        DS    CL3    Reserved                           01-DSPAP
+apqrc_SSIDN             DS    CL8    SSID for DASD                      01-DSPAP
+apqrc_DASDU             DS    CL8    Unit type for DASD                 01-DSPAP
+apqrc_TAPEU             DS    CL8    Unit type for tape                 01-DSPAP
+*-------------------------------------------------------------*
+*  Time-related  values, some  of which may be specified by   *
+*  the user through CHANGE.RECON and other commands           *
+*-------------------------------------------------------------*
+apqrc_TIME              DS    0CL24                                     01-DSPAP
+apqrc_TZDEF             DS    XL2    Input offset default               01-DSPAP
+apqrc_TMFMT             DS    XL5    Time format options                01-DSPAP
+                        DS    CL1    Reserved                           01-DSPAP
+apqrc_TPREC             DS    H      Time stamp precision               01-DSPAP
+apqrc_LOGRT             DS    XL12   Minimum log retention period       01-DSPAP
+apqrc_TZNUM             DS    H      Number of entries in time          01-DSPAP
+*                                    zone label table
+apqrc_TZTBL             DS    32CL8  Time zone label table              01-DSPAP
+apqrc_TROPT             DS    XL2    DBRC trace options                 01-DSPAP
+apqrc_IMSplex           DS    CL5    IMSplex name                       01-DSPAP
+                        DS    CL5    Reserved                           01-DSPAP
+*-------------------------------------------------------------*
+*  SIZEALERT and LOGALERT parameter values                    *
+*-------------------------------------------------------------*
+apqrc_SIZW_DSNUM        DS    F      SIZEALERT dsnum                    01-DSPAP
+apqrc_SIZW_VOLNUM       DS    F      SIZEALERT volnum                   01-DSPAP
+apqrc_SIZW_PERCENT      DS    F      SIZEALERT percent                  01-DSPAP
+apqrc_LOGW_DSNUM        DS    F      LOGALERT dsnum                     01-DSPAP
+apqrc_LOGW_VOLNUM       DS    F      LOGALERT volnum                    01-DSPAP
+apqrc_RECONDS_info      DSECT                                           01-DSPAP
+apqrc_RECONDS_DDname    DS    CL8    RECON DD name                      01-DSPAP
+apqrc_RECONDS_DSname    DS    CL44   RECON DS name                      01-DSPAP
+apqrc_RECONDS_status    DS    XL1    RECON DS status                    01-DSPAP
+apqrc_RECONDS_COPY1     EQU   X'80'  COPY 1                             01-DSPAP
+apqrc_RECONDS_COPY2     EQU   X'40'  COPY 2                             01-DSPAP
+apqrc_RECONDS_spare     EQU   X'20'  Spare                              01-DSPAP
+apqrc_RECONDS_discarded EQU   X'10'  Discarded                          01-DSPAP
+apqrc_RECONDS_unavail   EQU   X'08'  Unavailable                        01-DSPAP
+***********************************************************************
+*  END OF DSPAPQRC                                                    *
```

DSPAPQRI - DBDS/area recovery information block

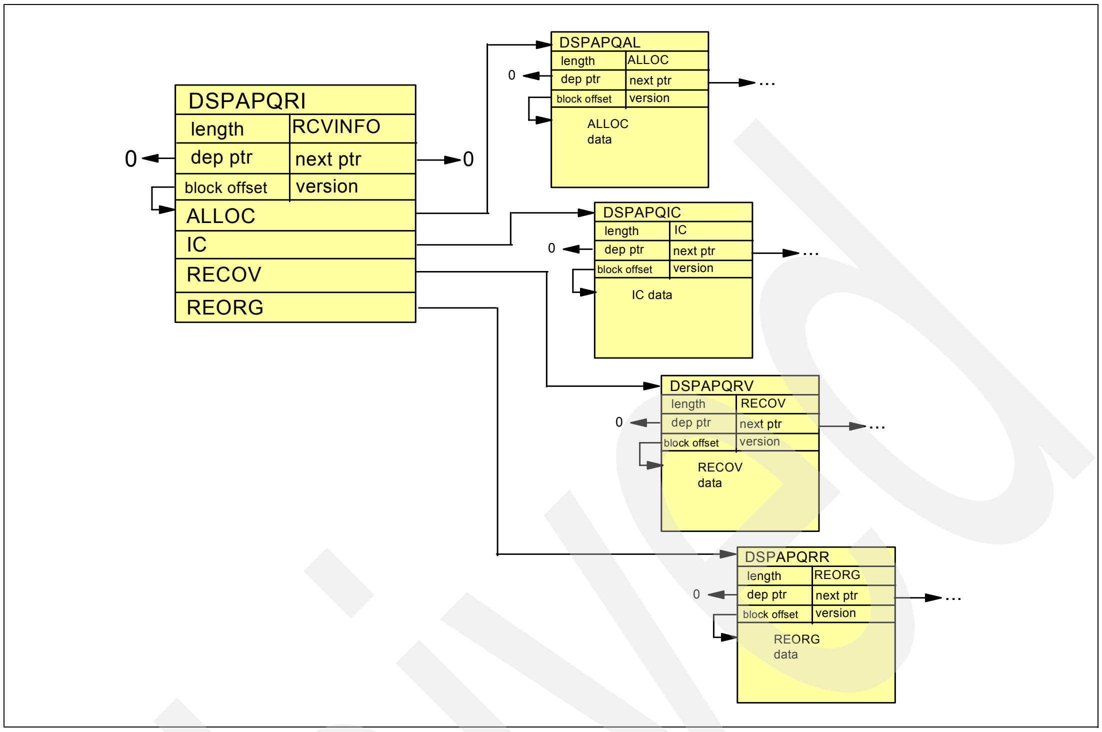


Figure A-13 RCVInfo output

Example: A-23 DSPAPQRI - DBDS/area recovery information block

```
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with           *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQRI - DBDS/Area Recovery Information block         *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the DBDS/Area Recovery Information output         *
+*   block returned on the DSPAPI FUNC=QUERY TYPE=DB call.             *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains    *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQRI'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'2801'  (DSPAPI_BlkType_RCVINFO)            *
+*     apqhd_depptr     = 0, no dependent block                         *
+*     apqhd_nextptr    = 0, no next block                              *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQRI              *
+*                                                                      *
+*  Notes:                                                              *
+*                                                                      *
+*   Dependencies: None.                                                *
+*                                                                      *
+*   Restrictions: None.                                                *
+*                                                                      *
+*  Parameters: None.                                                   *
+*                                                                      *
+*  External References:                                                *
+*    Macros:  None                                                     *
+*                                                                      *
+*    Modules: None                                                     *
+*                                                                      *
+*  Change Activity:                                                    *
+*                                                                      *
+*    Date       Ver   PTM/APAR Init Description                        *
+*    ---------- ----- -------- ---- -----------------------------      *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                      *
+************************************************************************
+DSPAPQRI                DSECT        Recovery Information block         01-DSPAP
+apqri_DBname            DS    CL8    Database name                      01-DSPAP
+apqri_DDname            DS    0CL8   DD name or                         01-DSPAP
+apqri_Areaname          DS    CL8    Area name                          01-DSPAP
+apqri_ALLOCptr          DS    A      ptr to ALLOC chain                 01-DSPAP
+apqri_ICptr             DS    A      ptr to IC    chain                 01-DSPAP
+apqri_RECOVptr          DS    A      ptr to RECOV chain                 01-DSPAP
+apqri_REORGptr          DS    A      ptr to REORG chain                 01-DSPAP
+************************************************************************
+*  END OF DSPAPQRI                                                     *
```

DSPAPQRR - DBDS reorganization block

Example: A-24   DSPAPQRR - DBDS reorganization block

```
+*******************************************************@SCPYRT**@SCPYRT*
+*                                                                      *
+*  Licensed Materials - Property of IBM                                *
+*                                                                      *
+*  5655-J38                                                            *
+*                                                                      *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved               *
+*                                                                      *
+*  US Government Users Restricted Rights - Use, duplication or         *
+*  disclosure restricted by GSA ADP Schedule contract with             *
+*  IBM Corp.                                                           *
+*                                                                      *
+****************************************************************@ECPYRT*
+*                                                                      *
+*  Name:      DSPAPQRR - DBDS Reorganization block                     *
+*                                                                      *
+*  Product:   DBRC Application Programming Interface                   *
+*                                                                      *
+*  Function:                                                           *
+*                                                                      *
+*   This macro maps the DBDS Reorganization output block               *
+*   returned on the DSPAPI FUNC=QUERY TYPE=DB call.                   *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQRR'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'3200'  (DSPAPI_BlkType_REORG)              *
+*     apqhd_depptr     = 0, no dependent blocks                       *
+*     apqhd_nextptr    = pointer to next REORG block for the DBDS     *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQRR             *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQRR                DSECT       DBDS Reorganization block          01-DSPAP
+apqrr_DBname            DS    CL8    Database name                      01-DSPAP
+apqrr_DDname            DS    0CL8   DD name or                         01-DSPAP
+apqrr_Areaname          DS    CL8    Area name                          01-DSPAP
+apqrr_RunTime           DS    XL12   The time at which the DBDS was     01-DSPAP
+*                                    reorganized.
+apqrr_Stoptime          DS    XL12   Stoptime of online reorg           01-DSPAP
+apqrr_flags             DS    XL1                                       01-DSPAP
+apqrr_ONL               EQU   X'80'  1=ONLINE/0=OFFLINE reorg           01-DSPAP
+apqrr_RECOV             EQU   X'40'  1=May be used for recovery         01-DSPAP
+                        DS    CL3    Reserved                           01-DSPAP
+apqrr_USID              DS    F      Associated USID                    01-DSPAP
+apqrr_PITR              DS    XL12   Stoptime moved - PITR              01-DSPAP
+***********************************************************************
+*  END OF DSPAPQRR                                                    *
+***********************************************************************
```

DSPAPQRV - recovery block

Example: A-25   DSPAPQRV - recovery Block

```
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQRV - DBDS Recovery block                          *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the DBDS Recovery output block returned on        *
+*   the DSPAPI FUNC=QUERY TYPE=DB call.                               *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains    *
+*   the following information:                                        *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQRV'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'3700'  (DSPAPI_BlkType_RECOV)              *
+*     apqhd_depptr     = 0, no dependent blocks                       *
+*     apqhd_nextptr    = pointer to next RECOV block for the DBDS     *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQRV             *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- ------------------------------    *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQRV                DSECT        DBDS Recovery block                01-DSPAP
+apqrv_DBname            DS    CL8    Database name                      01-DSPAP
+apqrv_DDname            DS    OCL8   DD name or                         01-DSPAP
+apqrv_Areaname          DS    CL8    Area name                          01-DSPAP
+apqrv_RunTime           DS    XL12   The time at which the DBDS was     01-DSPAP
+*                                    recovered.
+apqrv_EndTime           DS    XL12   Partial recovery only, the time    01-DSPAP
+*                                    to which the DBDS was restored
+***********************************************************************
+*  Update set IDs that were 'undone' by timestamp recovery.           *
+***********************************************************************
+apqrv_FUSID             DS    F      First undone USID                  01-DSPAP
+apqrv_LUSID             DS    F      Last undone USID                   01-DSPAP
```

```
+apqrv_FLAGS             DS    XL1    Flags                              01-DSPAP
+apqrv_PITR              EQU   X'80'  Point In Time Recovery             01-DSPAP
+***********************************************************************
+*  END OF DSPAPQRV                                                    *
+***********************************************************************
```

DSPAPQSL - DB/area authorized subsystem list block

Example: A-26   DSPAPQSL - DB/area authorized subsystem list block

```
DSPAPQSL                                                       31469001
+******************************************************@SCPYRT**@SCPYRT*
+*                                                                     *
+*  Licensed Materials - Property of IBM                               *
+*                                                                     *
+*  5655-J38                                                           *
+*                                                                     *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved              *
+*                                                                     *
+*  US Government Users Restricted Rights - Use, duplication or        *
+*  disclosure restricted by GSA ADP Schedule contract with            *
+*  IBM Corp.                                                          *
+*                                                                     *
+***************************************************************@ECPYRT*
+*                                                                     *
+*  Name:      DSPAPQSL - DB/Area Authorized Subsystem List block      *
+*                                                                     *
+*  Product:   DBRC Application Programming Interface                  *
+*                                                                     *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the authorized subsystem list portion of the      *
+*   Database, Area, and HALDB Partition output blocks returned on     *
+*   the DSPAPI FUNC=QUERY TYPE=DB call.  Refer to the DB, Area,       *
+*   or Partition blocks - DSPAPQDB, DSPAPQAR, or DSPAPQHP - for       *
+*   more information on accessing the authorized subsystem list.      *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*                                                                     *
+***********************************************************************
+DSPAPQSL                DSECT        DB/Area Auth SS List block         01-DSPAP
+apqsl_SSINFO            DS    0CL16  Subsystem list entry               01-DSPAP
+apqsl_SSNAME            DS    CL8    Subsystem ID                       01-DSPAP
```

```
+apqsl_ACCESS            DS    FL1    Access intent                      01-DSPAP
+apqsl_NCDDST            DS    FL1    Encoded state                      01-DSPAP
+apqsl_SSFLGS            DS    XL1    Flags                              01-DSPAP
+apqsl_SSROLE            EQU   X'80'  0 - Active SS, 1 - Tracking SS     01-DSPAP
+apqsl_SSXRFC            EQU   X'40'  1 - XRF Capable                    01-DSPAP
+                        DS    XL1    Reserved                           01-DSPAP
+apqsl_BKINFO            DS    OXL4   Backout information                01-DSPAP
+apqsl_BKCTR             DS    H      Backout needed count               01-DSPAP
+apqsl_BKNUM             DS    H      Backout done count                 01-DSPAP
+************************************************************************
+* Possible access intent values (apqsl_ACCESS)                         *
+************************************************************************
+apqsl_ACCRO             EQU   X'01'  READ-GO                            01-DSPAP
+apqsl_ACCRD             EQU   X'02'  READ                               01-DSPAP
+apqsl_ACCUP             EQU   X'03'  UPDATE                             01-DSPAP
+apqsl_ACCEX             EQU   X'04'  EXCLUSIVE                          01-DSPAP
+************************************************************************
+*  END OF DSPAPQSL                                                     *
+************************************************************************
```

DSPAPQSS - subsystem block

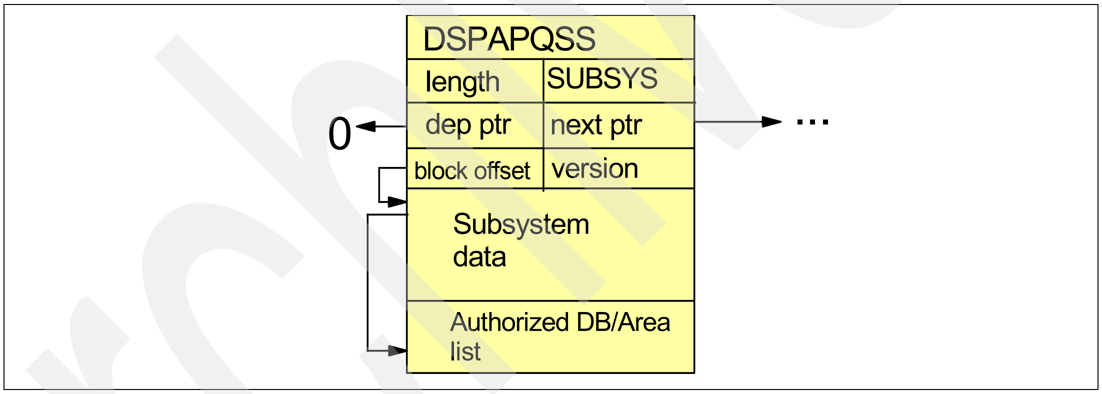


Figure A-14 Subsystem output

Example: A-27 DSPAPQSS - subsystem block

```
+*******************************************************@SCPYRT**@SCPYRT*
+*                                                                      *
+*  Licensed Materials - Property of IBM                                *
+*                                                                      *
+*  5655-J38                                                            *
+*                                                                      *
+*  (C) Copyright IBM Corp. 1974,2003 All Rights Reserved               *
+*                                                                      *
+*  US Government Users Restricted Rights - Use, duplication or         *
+*  disclosure restricted by GSA ADP Schedule contract with             *
+*  IBM Corp.                                                           *
+*                                                                      *
+****************************************************************@ECPYRT*
+*                                                                      *
+*  Name:      DSPAPQSS - Subsystem block                               *
+*                                                                      *
+*  Product:   DBRC Application Programming Interface                   *
+*                                                                      *
+*  Function:                                                          *
+*                                                                     *
+*   This macro maps the Subsystem output block returned on the        *
+*   DSPAPI FUNC=QUERY TYPE=SUBSYS call.                               *
+*                                                                     *
+*   The header of this block is mapped by DSPAPQHD which contains     *
+*   the following information:                                       *
+*                                                                     *
+*     apqhd_eyecatcher = 'DSPAPQSS'                                   *
+*     apqhd_length     = block length                                 *
+*     apqhd_blktype    = X'3F00'  (DSPAPI_BlkType_SUBSYS)             *
+*     apqhd_depptr     = 0, no dependent block                        *
+*     apqhd_nextptr    = pointer to next SUBSYS block                 *
+*     apqhd_blkoffset  = offset from DSPAPQHD to DSPAPQSS             *
+*                                                                     *
+*   Within DSPAPQSS are the following fields which are used to        *
+*   access authorized database/area information:                      *
+*                                                                     *
+*     apqss_AuthList  = offset from DSPAPQSS to list of authorized    *
+*                       DBs/Areas (apqss_AuthName)                    *
+*     apqss_AuthCount = number of authorized DB/Areas                 *
+*     apqss_AuthLen   = length of each entry                          *
+*                                                                     *
+*  Notes:                                                             *
+*                                                                     *
+*   Dependencies: None.                                               *
+*                                                                     *
+*   Restrictions: None.                                               *
+*                                                                     *
+*  Parameters: None.                                                  *
+*                                                                     *
+*  External References:                                               *
+*    Macros:  None                                                    *
+*                                                                     *
+*    Modules: None                                                    *
+*                                                                     *
+*  Change Activity:                                                   *
+*                                                                     *
+*    Date       Ver   PTM/APAR Init Description                       *
+*    ---------- ----- -------- ---- -----------------------------     *
+*    2003-10-02 1.0   KZ30219  pas  Prolog update                     *
+*    2004-03-04 1.01  PQ83837  pas  Add user-recoverable flag         *
+*                                                                     *
+***********************************************************************
+DSPAPQSS                DSECT        Subsystem block                    01-DSPAP
+apqss_SSID              DS    CL8    Subsystem identifier               01-DSPAP
+apqss_Authlist          DS    F      Offset to authd DB/Area list       01-DSPAP
+apqss_AuthCount         DS    F      Number of authorized DB/Areas      01-DSPAP
+apqss_AuthLen           DS    H      Length of auth entry               01-DSPAP
+                        DS    CL6    Reserved                           01-DSPAP
+apqss_LOGtime           DS    XL12   Start time of log                  01-DSPAP
+apqss_RELLVL            DS    FL1    Subsystem release level            01-DSPAP
+*                                    '71'X=V7R1,'81'X=V8R1
+*                                    '91'X=V9R1, etc.
+apqss_CoexLvl           DS    FL1    Coexistence level                  01-DSPAP
+apqss_IRLMCT            DS    FL1    IRLM status count                  01-DSPAP
+                        DS    CL1    Reserved                           01-DSPAP
+apqss_GSGname           DS    CL8    Global Service Group name          01-DSPAP
+apqss_IRLMID            DS    CL5    IRLM ID of SS                      01-DSPAP
+apqss_IRLMBK            DS    CL5    IRLM ID of backup SS               01-DSPAP
```

```
+apqss_FLAGS             DS    XL1    Flags                              01-DSPAP
+apqss_TYPE              EQU   X'80'  1=Online | 0=batch                 01-DSPAP
+apqss_ABTERM            EQU   X'40'  Abnormal termination               01-DSPAP
+apqss_RCVPRC            EQU   X'20'  Recovery processing started        01-DSPAP
+apqss_BKSIGN            EQU   X'10'  Backup SS signed on                01-DSPAP
+apqss_IRLMFL            EQU   X'08'  IRLM failure                       01-DSPAP
+apqss_COMMFL            EQU   X'04'  COMM failure                       01-DSPAP
+apqss_SYSFL             EQU   X'02'  SYS failure                        01-DSPAP
+apqss_ACTVST            EQU   X'01'  Status of active SS when           01-DSPAP
+*                                     backup exists, 1=abterm
+apqss_FLAGS2            DS    XL1    FLAGS 2                            01-DSPAP
+apqss_SHRING            EQU   X'80'  Sharing covered DBs                01-DSPAP
+apqss_TRKER             EQU   X'40'  Subsystem is a Tracker             01-DSPAP
+apqss_TRKTRM            EQU   X'20'  TRACKER has terminated             01-DSPAP
+apqss_TRCKED            EQU   X'10'  SSID is tracked                    01-DSPAP
+apqss_FRSTSO            EQU   X'08'  1ST signon after RSR takeover      01-DSPAP
+*                                     is in progress
+apqss_XRFCAP            EQU   X'04'  SS is XRF capable                  01-DSPAP
+apqss_BCKTKN            DS    H      Backup recovery token              01-DSPAP
+                        DS    CL2    Reserved                           01-DSPAP
+apqss_Authname          DSECT        Names of authd DB/Areas            01-DSPAP
+apqss_DBNAME            DS    CL8    DB name                            01-DSPAP
+apqss_AREANM            DS    CL8    If FP, Area name                   01-DSPAP
+apqss_SHRLVL            DS    FL1    Share level                        01-DSPAP
+apqss_DBACCS            DS    FL1    Access intent                      01-DSPAP
+apqss_DBNCOD            DS    FL1    Encoded state                      01-DSPAP
+apqss_DBSTAT            DS    FL1    DB status flags                    01-DSPAP
+apqss_DBEQCT            DS    H      DB EQE count                       01-DSPAP
+apqss_GLBDMB            DS    H      Global DMB number                  01-DSPAP
+apqss_AuthFlags         DS    XL1    Flags                              01-DSPAP
+apqss_NRDBUP            EQU   X'80'  Nonrecov DB/Area updated           01-DSPAP
+apqss_COVRD             EQU   X'40'  DB covered by GSG                  01-DSPAP
+apqss_NRECV             EQU   X'20'  Non-recoverable DB/Area            01-DSPAP
+apqss_ORDBDS            EQU   X'10'  0=A-J/1=M-V ACTIVE                 01-DSPAP
+apqss_OLRON             EQU   X'08'  0 = no OLR active                  01-DSPAP
+apqss_OLROWR            EQU   X'04'  0 = no OLR owner                   01-DSPAP
+apqss_OLROWD            EQU   X'02'  0 OLR not owned by SS              01-DSPAP
+apqss_URECV             EQU   X'01'  User-recoverable Area              01-DSPAP
+*                                     (VERSION=1.01)           @PQ83837
+                        DS    CL7    Reserved                           01-DSPAP
+***********************************************************************
+*  END OF DSPAPQSS                                                    *
```

B

Database Recovery Facility support

IMS Database Recovery Facility for z/OS (product number 5655-I44) is Version 2 of IMS Online Recovery Services for z/OS (product number 5655-E50). DRF includes the capabilities of ORS and has significant enhancements. Some of the capabilities and characteristics of DRF are the ones that were included in ORS. DRF supports IMS Version 7 with PQ73997, IMS Version 8 with PQ73990, and IMS Version 9 with PQ80051 and PQ80052.

We describe DRF here to summarize the available functions depending on IMS Version through 7 to 9.

DRF functions inherited from ORS

DRF recovers multiple database data sets in parallel. One invocation of DRF is used for these multiple recoveries. DRF may be used to invoke IMS database recoveries from an IMS online system by using /RECOVER commands. Users do not code JCL for these recoveries. DRF allocates the database data sets and recovery input data sets (image copies, change accumulations, and logs) from information in the RECONs.

DRF may be used to do two types of time stamp recoveries. A normal time stamp recovery is to a “recovery point”. This is a time which is not spanned by a RECON ALLOC record. It is a time when the database is not allocated to any IMS subsystem. The second type of time stamp recovery is a point-in-time recovery (PITR). A PITR may be done to any time, including a time at which the database was being updated by an IMS subsystem. This includes a time at which updates were in-flight. PITR recovers only updates that were committed before the PITR time.

DRF reads input data sets in parallel, which are image copies, change accumulations, and logs. These input data sets are read only once. Typically, a log or a change accumulation data set is used as input to the recovery of multiple database data sets. The user determines the number of parallel read processes.

DRF enhancements

DRF can be invoked by a batch job since IMS Version 7. DRF uses a different process for recovery. ORS used the facilities of the online system for recovering data sets. For full function databases it used the online system's database buffer pools. For Fast Path databases it used its own Fast Path buffer pool in the online system. DRF does not use online facilities for the recovery except for the processing of /RECOVER commands. These commands are used to invoke DRF in its own address spaces, since IMS Version 8.

DRF has several performance improvements over ORS. DRF allows users to specify separate limits for the number of parallel read processes used for tape and DASD devices.

DRF includes an option to create a list of inputs required for a recovery without actually doing the recovery, with VERIFY option This is useful to determine if the inputs are available and to determine if a recovery is possible.

DRF includes new reports on its processing. This includes a list of all of the inputs used in its recoveries.

Figure 13-1 on page 291 illustrates DRF process. DRF is invoked either by /RECOVER commands from an IMS online system or by a batch job step. In either case, the DRF master address space is executed. The online system and batch job may use the same catalogued procedure for this address space.

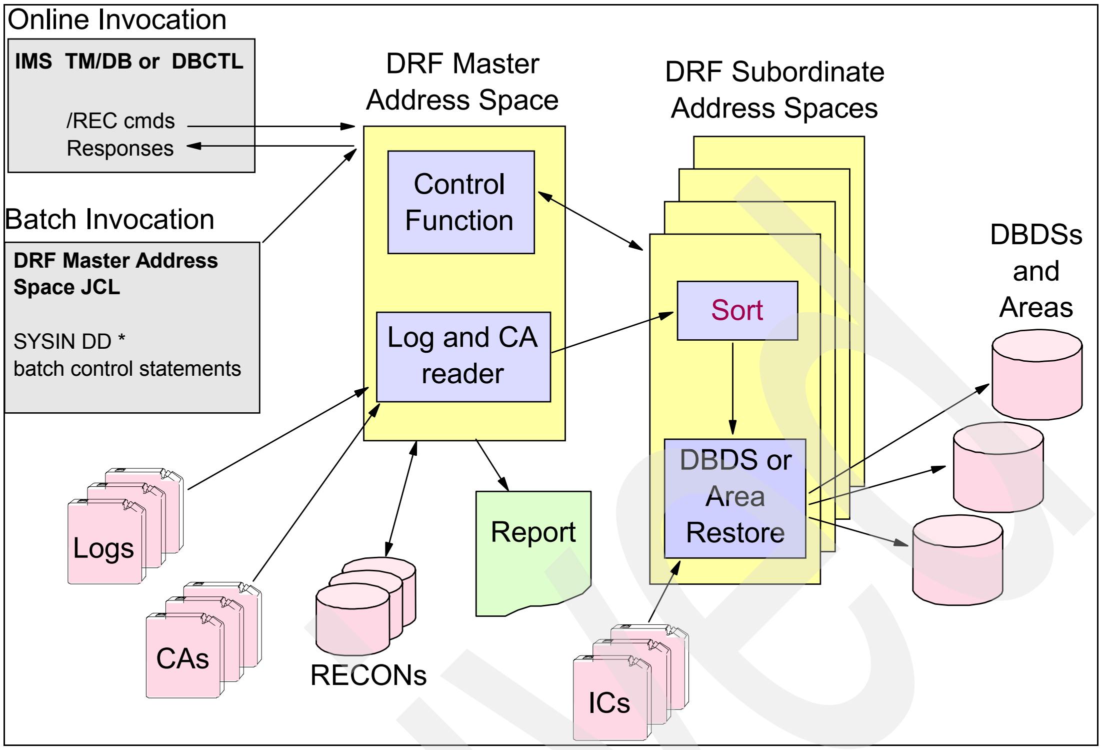


Figure 13-1 Database Recovery Facility (DRF) process

The master address space reads the RECONs and determines the inputs needed for recovery. It also reads the logs and change accumulation data sets for the recoveries. The required records from these data sets are passed to DRF subordinate address spaces. Each data set to be recovered is assigned to a subordinate address space. The number of address spaces is controlled by the NUM(nn) sub parameter of the SORT parameter in the FRXDRFxx PROCLIB member. The subordinate address spaces read the image copies, sort the log and change accumulation records, and apply all of these to recover the database data sets. The data sets are written sequentially.

Online invocation

DRF can be invoked with IMS online command, /RECOVERY, since IMS Version 8. This online command is compatible with ORS. This command is used as follows;

- `/RECOVER ADD`: used to build a list of elements to be recovered
- `/RECOVER REMOVE`: used to delete elements in the list
  - The list may contain data sets, areas, databases, HALDB partitions, and groups. Groups include CA groups, DBDS groups, and recovery groups. These groups are defined in the RECONs. Database groups cannot be used with DRF commands.
- `/RECOVER START`: used to start a recovery of all of the elements in the list
- `/RECOVER STOP`: used to stop the recovery of all of the database data sets whose recovery is in process
- `/RECOVER TERMINATE`: used to stop the DRF master address space
  - Without this command, the master address space remains after a recovery is completed.

See *IMS Command Reference* for more information about `/RECOVERY` command.

DFSORSxx PROCLIB member is used for online execution of DRF. This member is specified by IMS ORSMBR= execution parameter, and contains only one parameter for DRF, DRFNAME(nnnnnnnn). DRFNAME is the name of procedure that executes DRF. The default name for DRFNAME is FRXJCLDF. This member was also used by ORS. ORS parameters in this member are ignored by DRF.

Batch invocation

The batch execution of DRF is the execution of the DRF master address space. The batch invocations is available since IMS Version 7. This is the same address space which is invoked by the online execution of DRF. DRF supplies a sample cataloged procedure, FRXJCLDF, which may be used for this address space. If you use the procedure instead of the JCL shown on this page, you only need to specify the //SYSIN DD statement and the control statements. See Example 13-1.

Example 13-1   DRF batch invocation; FRXJCLDF

```
//MYJOB   JOB   <acctng data etc.>
//DRFSTEP  EXEC PGM=FRXSDR00,PARM='DRFMBR=AA,BPECFG=BPEDRFCG'
//STEPLIB   DD   DSN=DRF.SFRXRESL,DISP=SHR
//          DD   DSN=IMS.SDFSRESL,DISP=SHR
//PROCLIB   DD   DSN=IMS.PROCLIB,DISP=SHR
//DBDLIB    DD   DSN=IMS.DBDLIB,DISP=SHR
//SYSPRINT  DD   SYSOUT=*
//REPORT    DD   SYSOUT=*
//SYSUDUMP  DD   SYSOUT=*
//SYSIN   DD   *
ADD DB (ABC1 ABC2 ABC3)
ADD RECOVGRP (XYZ1 XYZ2)
START RCVTIME('20033611330000',PITR)
/*
```

The batch control statements are similar to online commands. Only ADD, REMOVE, and START are used.

- ADD: used to build a list of elements to be recovered
- REMOVE: used to delete elements in the list
  - The list may contain data sets, areas, databases, HALDB partitions, and groups. Groups include CA groups, DBDS groups, and recovery groups. These groups are defined in the RECONs. Database groups cannot be used with DRF commands. When an area is specified in a batch control statement the database must also be specified.
- START: used to start a recovery of all of the elements in the list

Execution enhancements

ORS could limit the number of parallel read processes. This was done with the READNUM parameter. DRF enhances this capability to limit both the total number of parallel read processes and the number used for tape inputs. The limits are set by the READNUM specifications in the FRXDRFxx PROCLIB member. The total number of processes may be overridden on the START command (batch or online). The number of tape drives may be overridden on the batch START command.

The new VERIFY option on the START command may be used to produce a report of the inputs required for a recovery without recovering the database data sets. The VERIFY option may be specified on either the batch or online version of the START command. Some installations find it useful to verify that the required inputs are available before they start a recovery.

C

IMS Tools support for IMS Version 9

This appendix provides the *minimum* software level requirements for the IMS Tools products to be used with IMS Version 9. There is a comment after each product that tells the level of support; it is either not supported, tolerate or exploit.

- **Not supported** means that the product does not function properly on IMS Version 9.
- **Tolerate** allows the product to execute as it did on a previous release of IMS, but does not utilize any of the new IMS Version 9 functions.
- **Exploit** means that the product utilizes the new functions and features available in IMS Version 9.

IMS Tools and IMS Version 9 compatibility

This list provides the *minimum* maintenance required for IMS™ Tools to support IMS for z/OS Version 9.1. You can find the table and the latest updates to it from the following Web site:

http://www-1.ibm.com/support/docview.wss?rs=434&cntext=sszjxp&uid=swg21167251&loc=en_US&cs=utf-8&lang=en+en

- Application Recovery Tool 1.2.0 Exploit with APAR PQ84689
- Data Encryption 1.1.0 Exploit, no change required
- Data Refresher 1.1.0 Exploit, no change required
- DB/DC Data Dictionary 1.6.0 Tolerate, no change required
- IMS ADF II 2.2.0 X Tolerate, no change required
- IMS Advanced ACB Generator 1.1.0 Not supported, replaced by IMS Library Integrity Utilities
- IMS Batch Backout Manager 1.1.0 Exploit with APAR PQ79075
- IMS Batch Terminal Simulator 3.1.0 Exploit with APAR PQ80182
- IMS Buffer Pool Analyzer 1.1.0 Exploit with APAR PQ80183
- IMS Checkpoint Wrapper 1.1.0 Exploit with APAR PQ80214
- IMS Command Control Facility 1.1.0 Exploit with APAR PQ79840
- IMS Connect 1.2.0 Tolerate, no change required
- IMS Database Control Suite 3.1.0 Exploit with APAR PQ80215
- IMS Database Recovery Facility 2.1.0 Exploit with APARs PQ80051, PQ80052
- IMS Database Repair Facility 1.1.0 and 1.2.0 Exploit, no change required
- IMS DataPropagator™ 3.1.0 Exploit, no change required
- IMS DEDB Fast Recovery 2.2.0 Exploit with APAR PQ80193
- IMS ETO Support 3.1.0 Tolerate with APAR PQ79849
- IMS Fast Path Basic Tools 1.2.0 Exploit with APAR PQ80197
- IMS Fast Path Online Tools 2.1.0 Exploit with APAR PQ80200
- IBM Fault Analyzer V4 Exploit with APAR PQ94701
- IBM Fault Analyzer V5 Exploit with APAR PQ95289
- IMS HD Compression Extended 2.2.0 Exploit with APAR PQ79146
- IMS HP Change Accumulation 1.2.0 Exploit with APAR PQ80221
- IMS HP Fast Path Utilities 2.1.0 Exploit, no change required
- IMS HP Image Copy 3.1.0 Exploit with APARs PQ83801, PQ80186, PQ83453, PQ86664, PQ80197, PQ89630
- IMS HP Image Copy 3.2.0 Exploit, no change required
- IMS HP Load 1.1.0 Exploit with APARs PQ80198, PQ85616
- IMS HP Load 2.1.0 Exploit, no change required
- IMS HP Pointer Checker 2.1.0 Exploit with APARs PQ80186, PQ83453, PQ86664
- IMS HP Prefix Resolution 2.1.0 and 3.1.0 Exploit, no change required
- IMS HP Sysgen Tools 1.1.0 Exploit with APAR PQ79845
- IMS HP Unload 1.1.0 Exploit with APARs PQ80191, PQ83387

- IMS Image Copy Extensions 2.1.0 Tolerate with APAR PQ80187
- IMS Index Builder 2.2.0 Exploit with APAR PQ80216
- IMS Index Builder 2.3.0 Exploit with APARs PQ80216, PQ88453
- IMS Library Integrity Utilities 1.1.0 Exploit with APARs PQ80199, PQ83869
- IMS Library Management Utilities 1.1.0 Not supported, replaced by IMS Library Integrity Utilities
- IMS MFS Reversal Utilities 1.1.0 Tolerate, no change required
- IMS Multi-Dialog Manager 1.1.0 Tolerate with APAR PQ80331
- IMS Network Compression Facility 1.1.0 Tolerate with APAR PQ79842
- IMS Online Recovery Services 1.1.0 Not supported, replaced by IMS Database Recovery Facility
- IMS Online Reorganization Facility 1.1.0 Exploit, no change required
- IMS Parameter Manager 1.1.0 Exploit, no change required
- IMS Parallel Reorganization 2.1.0 Exploit with APARs PQ80190, PQ83386, PQ80191, PQ83387, PQ80198, PQ85616
- IMS Parallel Reorganization 3.1.0 Exploit, no change required
- IMS Performance Analyzer 3.2.0 Exploit with APARs PQ79473, PQ83568, PQ85111, PQ85681
- IMS Performance Monitor 1.1.0 Exploit with APAR PQ85342
- IMS Problem Investigator 1.1.0 Exploit with APAR PQ79474
- IMS Program Restart Facility 2.1.0 Exploit with APARs PQ79074, PQ84911, PQ86995
- IMS Queue Control Facility 1.2.0 Exploit with APAR PQ69897
- IMS Queue Control Facility 2.1.0, Exploit, no change required
- IMS Sequential Randomizer Generator 1.1.0 Exploit, no change required
- IMS Workload Router 2.4.0 Tolerate, no change required

D

IMS Control Center

This appendix provides information about how to download and connect the command center to IMS Version 9. This appendix covers the following topics:

- Workstation software
- Host requirements
- What set up steps you have to do
- Using the IMS Control Center

IMS Control Center steps you need to know

You can manage your IMS systems using a graphical interface from a workstation using the IMS Control Center. The IMS Control Center uses the IMS single point of control (SPOC) functions. The IMS Control Center is part of the IBM DB2 Universal Database Version 8 Control Center. The DB2 Control Center is available with the IBM DB2 Universal Database (DB2 UDB) Administration Client, Version 8.2.

Using the IMS Control Center, you can view the members of the IMSplex and define groups of members. The Control Center supports both IMS Version 8 and later versions. From the Control Center you can issue IMS type-2 commands using command windows or wizards, depending on how much assurance you want. Wizards help you build and issue the commands. The results are displayed in the IMS Results window. In addition, using the Command Editor, you can issue both type-1 and type-2 IMS commands.

Workstation software

From the IMS home page click the IMS Control Center. This will take you into the DB2 Information Center, and the information needed to setup the IMS Control Center, and also where to download the software.

From the DB2 download site you need to choose your workstation platform and download the related DB2 administration client.

- If you have DB2 UDB:

  At a minimum, install the administration client on your workstation. For information about installing DB2 UDB, refer to the *Getting Started* information for DB2 UDB. To install the administration client, select the Custom installation when asked. Then choose to install these features under Administration tools:

  - Control Center
  - Command Editor
  - Configuration Assistant

  You can choose to install any other features you want. Be sure that the features you do not want to install are marked with a red X.

- If you do not have DB2 UDB:

  You can download the IBM DB2 UDB administration client from the DB2 UDB support Web site: DB2 UDB support downloads

Migration

If you have installed an earlier version of the IMS Control Center, you can install DB2 UDB Administration Client Version 8.2 without uninstalling the earlier version. However, installing the latest version of the IMS Control Center will remove your IMS Control Center IMSplex system definitions, tools settings, and command history from an earlier version.

Host requirements

First, check that your system has the required software at the correct levels.

IMS Version 8

- You must have APAR PQ69527 installed to enable IMS Control Center support.

- The Common Service Layer must be configured.
- IMS Connect Version 1.2 or later with APARs PQ62379 and PQ70216 installed.

IMS Version 9

- IMS Version 9 provides an integrated IMS Connect function; make sure that the Connect function is installed and configured on your system.
- You must have APARs PQ92398 installed.
- The Common Service Layer must be configured (this can be an Enhanced Command Environment).

  You must have APAR PQ69527 installed to enable IMS Control Center support.

What set up steps you have to do

Follow these steps to complete the set up for the IMS Control Center:

1. Configure the IMS Operations Manager (OM).

   Check your OM configuration to ensure the following:

   a. In the OM initialization parameters member (CSLOIxxx), the IMSplex name specified in the IMSPLEX parameter must be the same as that specified in both IMS member DFSCGxxx and in your IMS Connect configuration member (HWSCFxxx).

   b. Also in the OM initialization parameters member (CSLOIxxx), CMDTEXTDSN must point to the data set containing the command syntax translatable text that was placed in the SDFSDATA library when IMS was installed.

For more information about configuring OM, refer to *IMS Version 9: Common Service Layer Guide and Reference,* SC18-7816.

2. Configure IMS Connect.

   Modify the configuration member you use to define the environment for IMS Connect or create a new IMS Connect configuration member. The IMS Connect configuration member is the PROCLIB member you specify in the HWSCFG= parameter of your IMS Connect startup JCL. For more information about IMS Connect configuration, see Chapter 4, “IMS Connect Definition and Tailoring”, in the *IMS Connect Guide and Reference*, SC27-0946.

   In your configuration member, specify the following settings:

   c. The user message exits for IMSplex support (HWSCSLO0 and HWSCSLO1) in the EXIT keyword statement

   d. Each IMSplex you want to manage in the IMSPLEX keyword statement

Example D-1 is a sample IMS Connect configuration member with the updates for the IMS Control Center.

Example: D-1   HWSCFGxx sample

```
HWS (ID=IMSGCONN,RACF=N,XIBAREA=20)
TCPIP (HOSTNAME=TCPIP,PORTID=(7003,LOCAL),MAXSOC=2000,TIMEOUT=8800,
EXIT=(HWSSMPL0,HWSCSLO0,HWSCSLO1),IPV6=Y)
DATASTORE (ID=IMSG,GROUP=IMS9EXCF,MEMBER=HWS910G,TMEMBER=SCSIM9G)
IMSPLEX (MEMBER=IMSPLEXG,TMEMBER=PLEXG)
```

Note the port number for the IMSplex in the PORTID parameter in the TCPIP keyword statement. You will use this later when you add the IMSplex to the Control Center. You also need to know the IP address of your system.

3. If you created a new IMS Connect configuration member, update the HWSCFG= parameter in your IMS Connect startup JCL to point to the new member.
4. Stop IMS Connect and restart it with your updated configuration.
5. On your workstation with DB2 UDB installed, open the Control Center by selecting Start->Programs->IBM DB2->General Administration Tools-->Control Center.

   The Control Center is displayed see Figure D-1:

   an info-pop will ask you to choose “basic”, “advanced” or “customized”, you need to mark **advanced**.

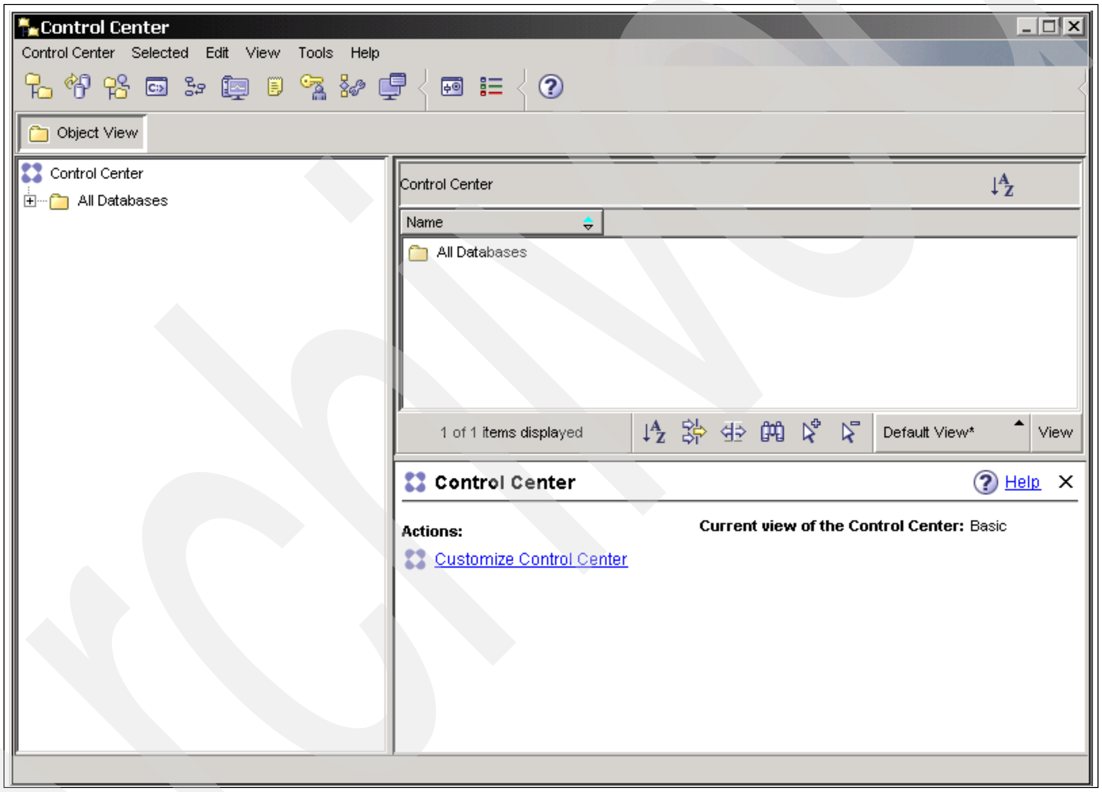
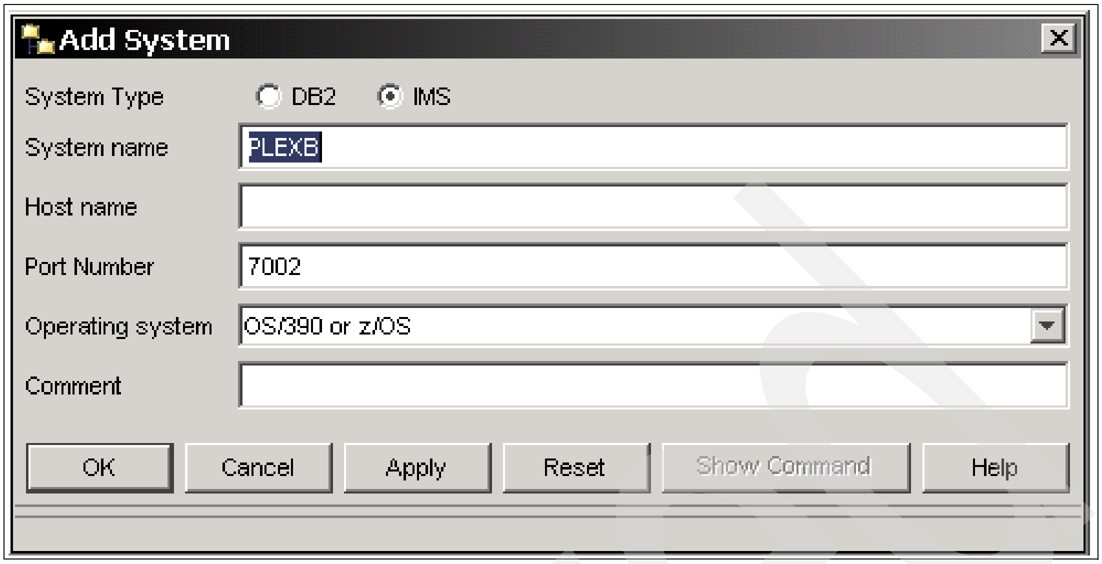


Figure D-1 DB2 Control Center

6. Add the IMSplex that you want to manage by right-clicking on the **All Cataloged Systems** folder (your Control Center folder may just be named **Systems**). Then select **Add...** from the pop-up menu.

   The Add System window is displayed in Figure D-2 on page 303:



System Type ○ DB2 ● IMS

System name PLEXB

Host name

Port Number 7002

Operating system OS/390 or z/OS

Comment

Figure D-2 ADD system

You need the saved information from the HWSCFGxx file.

7. Select IMS as the System Type and enter the information for your IMSplex.
   - Enter your IMSplex name in the System name field.
   - Enter host name or the IP address of the host system for the IMSplex in the Host name field. You might have to overwrite the IMSplex name that the system places in this field.
   - Enter the port number as specified in your IMS Connect configuration member in the Port number field.

     Click OK.

8. To log in to the IMSplex, right-click the IMS system folder in the object tree in the left frame of the Control Center and select **Login**... from the pop-up menu. A Login window is displayed.

9. Enter the User ID you use to issue IMS commands and its password.

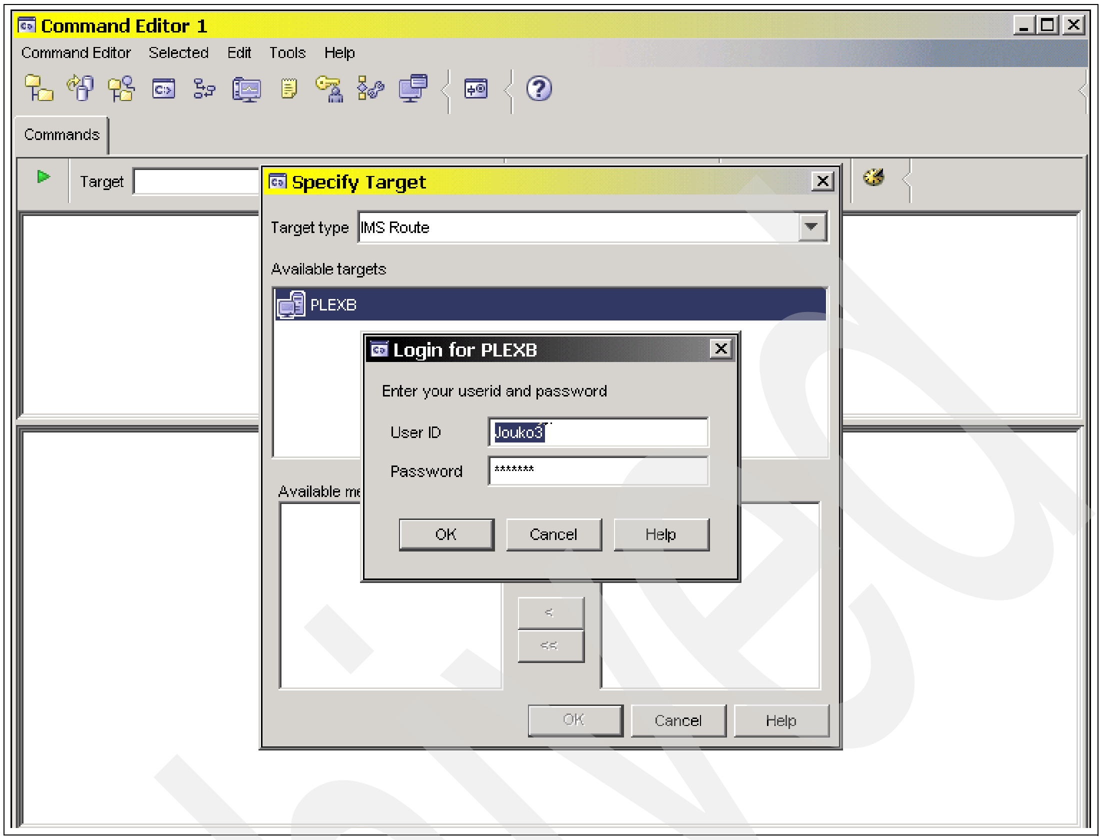
   Click OK.

You can now use the IMS Control Center and you can also use the Command Editor to issue IMS commands. For more information about using the IMS Control Center see the DB2 Information Management Software Information Center for z/OS Solutions on page:

http://publib.boulder.ibm.com/infocenter/dzichelp/index.jsp

Using the IMS Control Center

After the IMS Control Center is successfully installed and connected to the IMS Connect you are able to use the Command Editor as seen on Figure D-3 on page 304



Enter your userid and password

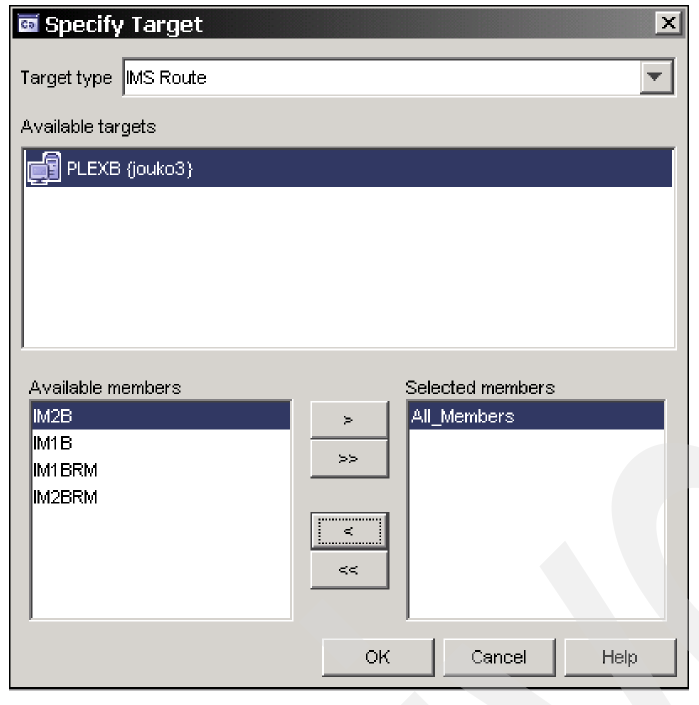
User ID Jouko3

Password *******

Target type IMS Route

Figure D-3 IMS Control Center command editor

After entering the target in our case PLEXB a security pop-up asks for user ID and password. As soon as you've entered it you'll find the screen Figure D-4 on page 305 where you may be able to specify a single member of your sysplex IMS or all members.



Target type: IMS Route

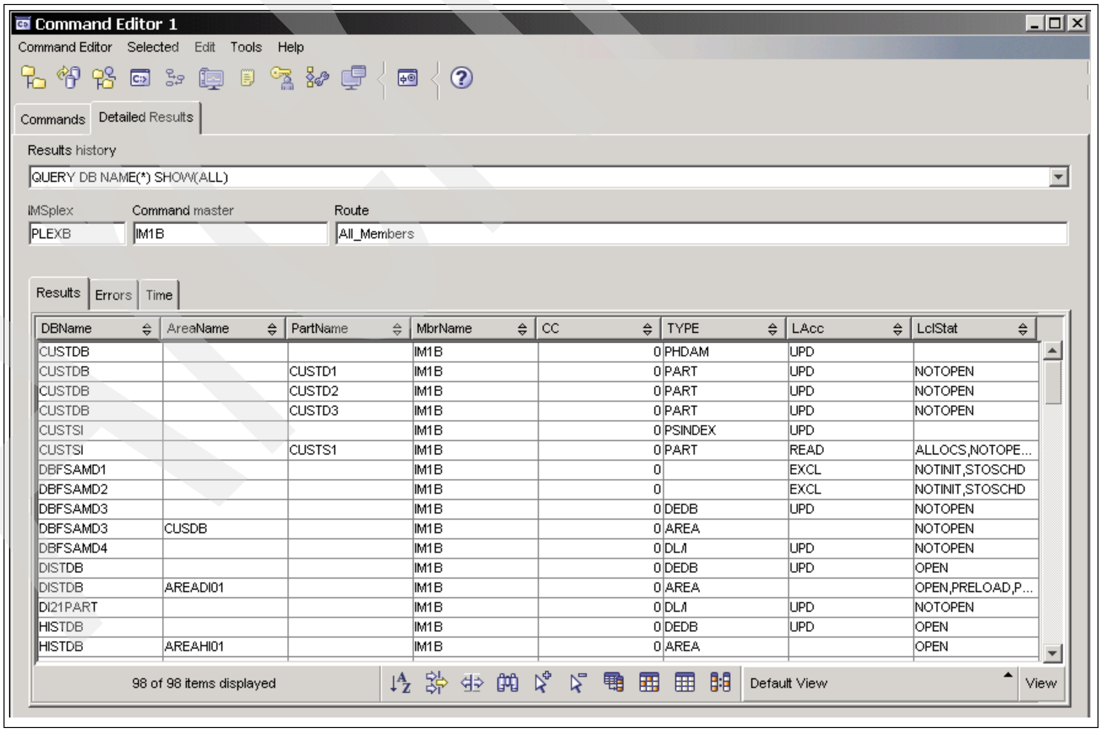
Available targets: PLEXB {jouko3}

Available members: IM2B, IM1B, IM1BRM, IM2BRM

Selected members: All_Members

Figure D-4 Specify target

Now you are able to use the new commands as introduced in “Type-2 database commands” on page 119. We provided a sample of the QRY DB command and its results in Figure D-5.



Results history: QUERY DB NAME(*) SHOW(ALL)

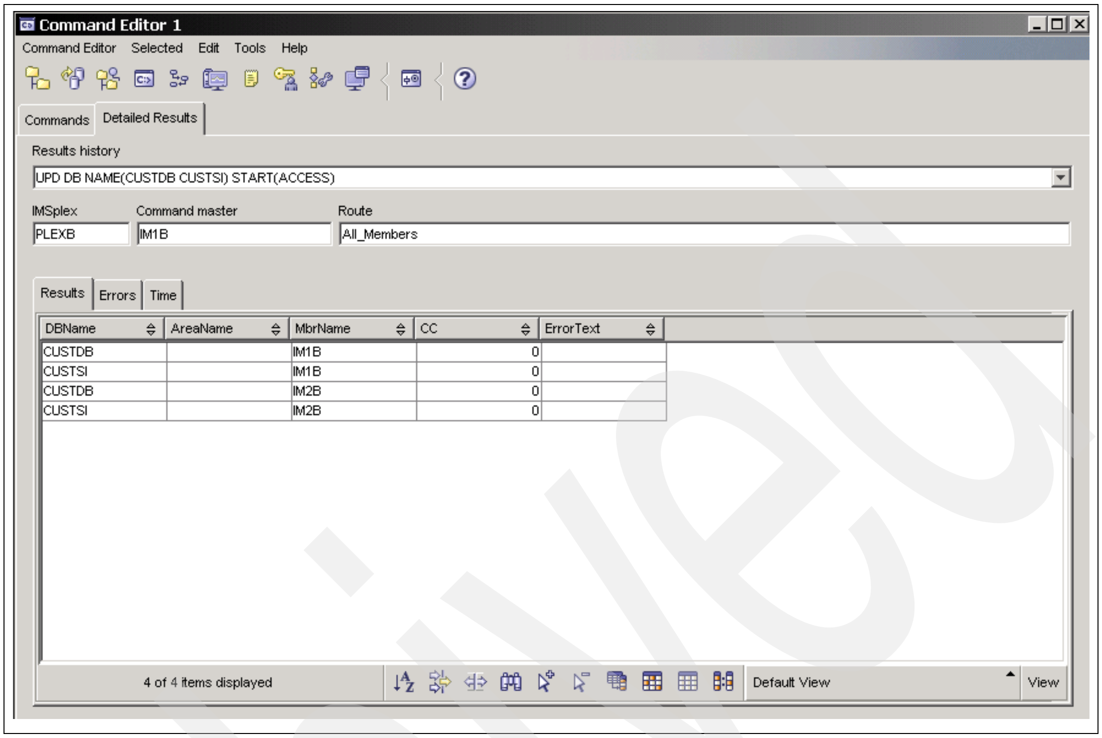
IMSplex: PLEXB Command master: IM1B Route: All_Members

DBName	AreaName	PartName	MbrName	CC	TYPE	LAcc	LclStat
CUSTDB			IM1B	0	PHDAM	UPD	
CUSTDB		CUSTD1	IM1B	0	PART	UPD	NOTOPEN
CUSTDB		CUSTD2	IM1B	0	PART	UPD	NOTOPEN
CUSTDB		CUSTD3	IM1B	0	PART	UPD	NOTOPEN
CUSTSI			IM1B	0	PSINDEX	UPD	
CUSTSI		CUSTS1	IM1B	0	PART	READ	ALLOCS,NOTOPE...
DBFSAMD1			IM1B	0		EXCL	NOTINIT,STOSCHD
DBFSAMD2			IM1B	0		EXCL	NOTINIT,STOSCHD
DBFSAMD3			IM1B	0	DEDB	UPD	NOTOPEN
DBFSAMD3	CUSDB		IM1B	0	AREA		NOTOPEN
DBFSAMD4			IM1B	0	DL/I	UPD	NOTOPEN
DISTDB			IM1B	0	DEDB	UPD	OPEN
DISTDB	AREADI01		IM1B	0	AREA		OPEN,PRELOAD,P...
DI21PART			IM1B	0	DL/I	UPD	NOTOPEN
HISTDB			IM1B	0	DEDB	UPD	OPEN
HISTDB	AREAHI01		IM1B	0	AREA		OPEN

98 of 98 items displayed

Figure D-5 QRY DB SHOW(ALL) sample

Also the new UPDATE data base command can be used via the Command Editor see Figure D-6 on page 306. Here in the “Detailed Results” tab, you find folders for results, errors, if any, returning the condition and reason codes, as well as a time folder showing you the duration and access time.



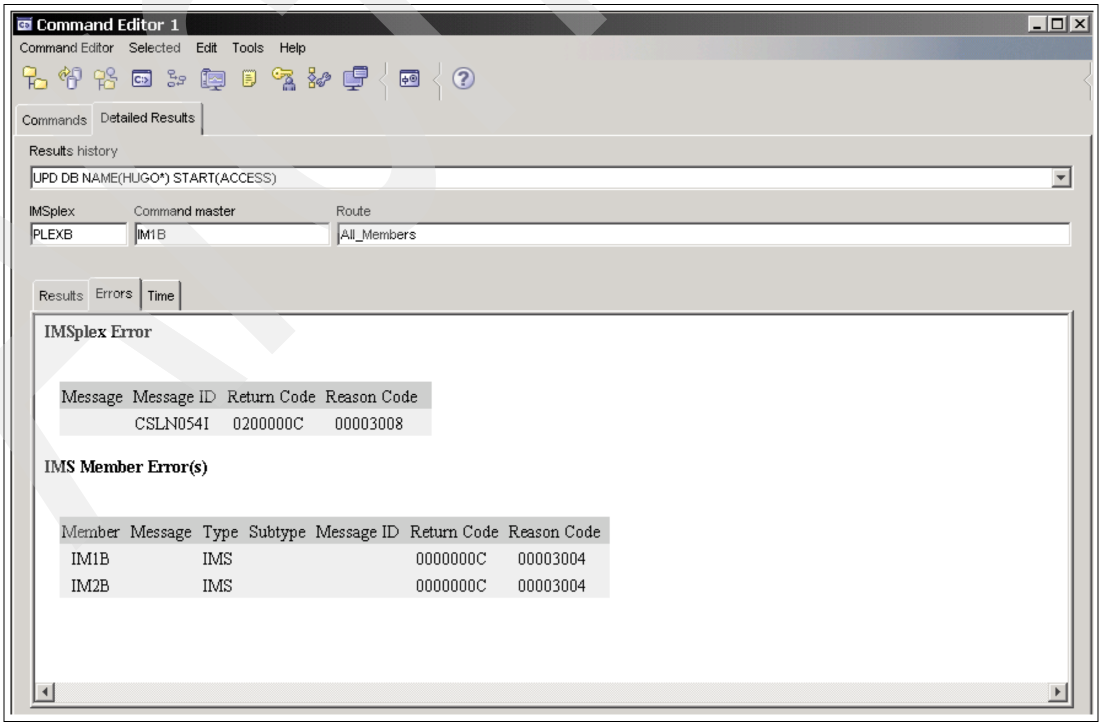
UPD DB NAME(CUSTDB CUSTSI) START(ACCESS)

IMSplex: PLEXB
Command master: IM1B
Route: All_Members

DBName	AreaName	MbrName	CC	ErrorText
CUSTDB		IM1B	0	
CUSTSI		IM1B	0	
CUSTDB		IM2B	0	
CUSTSI		IM2B	0	

Figure D-6 UPD DB NAME(CUSTDB CUSTSI) START(ACCESS)

An error would result if for example you tried to update an unknown database. You will see the return and reason codes as well as the CSLN054I message as seen in Figure D-7.



UPD DB NAME(HUGO*) START(ACCESS)

IMSplex: PLEXB
Command master: IM1B
Route: All_Members

IMSplex Error

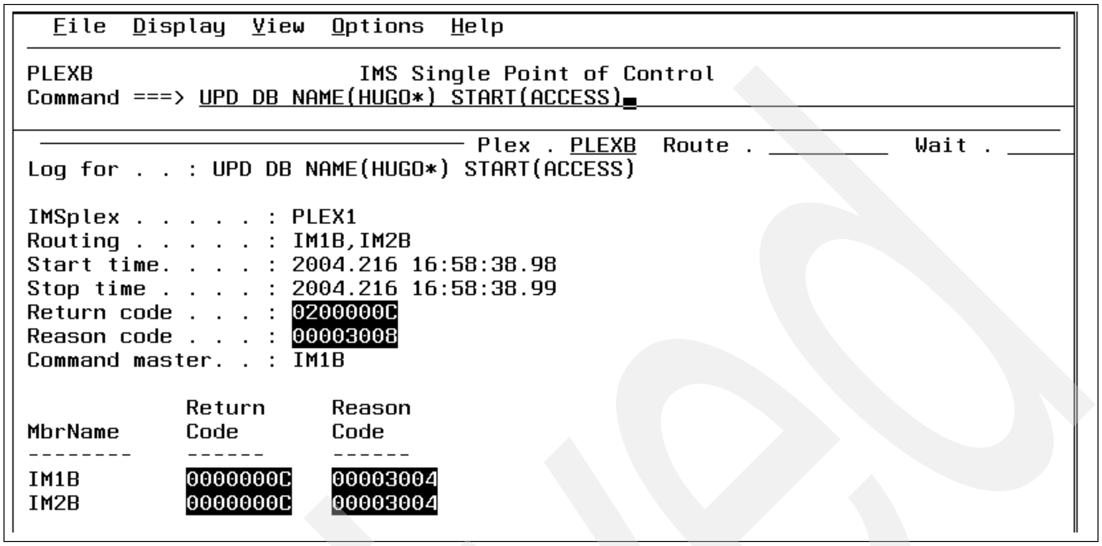
Message	Message ID	Return Code	Reason Code
	CSLN054I	0200000C	00003008

IMS Member Error(s)

Member	Message	Type	Subtype	Message ID	Return Code	Reason Code
IM1B		IMS			0000000C	00003004
IM2B		IMS			0000000C	00003004

Figure D-7 CSLN054I message sample

If you were using the TSO SPOC, you will find the return and reason codes for the same command by simply pressing PF4 on the return panel it shows you the same as shown in Figure D-8.



```
  File  Display  View  Options  Help
 ------------------------------------------------------------------------------
 PLEXB                    IMS Single Point of Control
 Command ===> UPD DB NAME(HUGO*) START(ACCESS)
 ------------------------------------------------------------------------------
 ---------------------------------- Plex . PLEXB  Route . _________  Wait . _____
 Log for . . : UPD DB NAME(HUGO*) START(ACCESS)

 IMSplex . . . . . : PLEX1
 Routing . . . . . : IM1B,IM2B
 Start time. . . . : 2004.216 16:58:38.98
 Stop time . . . . : 2004.216 16:58:38.99
 Return code . . . : 0200000C
 Reason code . . . : 00003008
 Command master. . : IM1B

             Return     Reason
 MbrName     Code       Code
 --------    ------     ------
 IM1B        0000000C   00003004
 IM2B        0000000C   00003004
```

Figure D-8 TSO SPOC sample of return and reason codes

The explanation of this return and reason code is mentioned in the CSLN054I message: “No clients returned return code 0. Check return code(s) Explanation: The command was routed to multiple clients. None of the clients that processed the command returned a return code 0 and reason code 0 to the Operations Manager. For information on the CSLOMCMD return and reason code, refer to *IMS Version 9: Common Service Layer Guide and Reference,* SC18-7816.”

Using the IMS Control Center and Command Editor with IMS Version 9

You are able to update the resources that are new to IMS Version9, that is, database, online reorganization, area, and datagroup.

Using the IMS Control Center with the latest version of DB2 UDB Administration Client Version 8.2 has the following enhancements (available whether you are using IMS V8 or V9):

- Command Editor (previous versions referred to this as the "Command Center")

  The Command Editor has a "syntax assist" capability. In the command area of the Command Editor you can start typing the command, or press the spacebar to display a list of the type-2 IMS commands, if syntax assist is on. Select the command you want using the arrow keys and use the spacebar to add it to the editor. Press the spacebar again to continue.

- Command Shortcuts

  You can set up short names for commands that you use often; then you can use the short names in the Command Editor. For example, if you routinely want to query the same series of transactions, those that start with TSP, LMH, and SRG, you can create a short name &MYQRY and specify the command QUERY TRAN NAME(TSP* LMH* SRG*). To define a short name and its command, click the Command Editor. The icon is only available when you have selected an IMS target.

  When you enter &MYQRY in the Command Editor, you will run the QUERY TRAN command you specified. You can also define the defaults to use for parameters for commonly used commands in the IMS Command Shortcuts.

- Compare Two Sets of command outputs from the Results panel.

  You can compare results from running a type-2 command at different times. You can compare results to view changes after updating members, to check status, to find differences in attributes. The comparison must be between results from the same command. A message informs you if the results sets you chose are different. The results from the second results set you selected are shown with a gray background. The individual results that are different are highlighted in the second set, when a value appears in both sets but are different. You can choose to show only the differences between the results or show the full results. To open the IMS Compare Results window: Click in the IMS Results window.

Abbreviations and acronyms

ACB	application control block
ACEE	access control environment element
AGN	application group name
AOI	automated operator interface
APAR	authorized program analysis report
APF	authorized program facility
API	application programming interface
APPC	advanced program to program communication
ARLN	automatic RECON loss notification
ARM	Automatic Restart Manager
ATM	automatic teller machine
AWE	Asynchronous Work Element
BMP	batch message program
BPE	Base Primitive Environment
BSDS	boot strap data set
CBPDO	custom built product delivery offering
CDT	class descriptor table
CF	Coupling Facility
CFRM	Coupling Facility Resource Manager
CI	control interval
CICS	Customer Information Control System
COBOL	common business oriented language
CPF	command prefix facility
CQS	Common Queue Server
CSA	common system area
CSI	consolidated software inventory
CSL	Common Service Layer
CST	Consolidated Service Test
CTC	channel-to-channel
DASD	direct access storage device
DB/DC	database/data communications
DB2	DATABASE 2
DBA	database administrator
DBCTL	database control
DBD	database description
DBDS	database data set
DBRC	data base recovery control
DBWP	database work pool
DEDB	data entry database
DL/I	Data Language/I
DLI/SAS	DL/I separate address space
DLT	database level tracking (RSR)
DMB	database management block
DMBP	data management block pool
DRA	database resource adapter
DRF	IMS Database Recovery Facility
DSAB	data set association block
EBCDIC	extended binary coded decimal interchange code
ECSA	extended common system area
EEQE	extended error queue element
EIS	Enterprise Information System
EJB	Enterprise Java Bean
EMCS	extended multiple consoles support
EMEA	Europe, Middle East and Africa
EMH	expedited message handler
EMHQ	expedited message handler queue
ESAF	external subsystem attach facility
ESCON®	Enterprise System Connection
ESO	Extended Service Offering
ETO	Extended Terminal Option
EX	execution (IVP)
FDBR	Fast Database Recovery
FICON	Fiber Connection
FMID	function modification identifier
FT	file tailoring (IVP)
FTP	File Transfer Protocol
GSAM	generalized sequential access method
HALDB	High Availability Large Database
HDAM	hierarchic direct access method
HFS	hierarchical file system
HIDAM	hierarchic indexed direct access method
HLQ	high-level qualifier
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol

IBM	International Business Machines Corporation
IFP	IMS Fast Path program
IIOP	Internet inter-ORB protocol
ILDS	indirect list data set
ILE	indirect list data set list entry
ILS	Isolated Log Sender
IMS	Information Management System
IPCS	Interactive Problem Control System
IPL	initial program load
IRLM	Integrated Resource Lock Manager
ISC	intersystem communication
ISD	independent software delivery
ISPF	Interactive Systems Productivity Facility
ITSO	International Technical Support Organization
IVP	installation verification program
J2C	J2EE Connector Architecture
J2C	J2EE Connector Architecture
J2EE	Java 2 Platform, Enterprise Edition
JBP	Java batch processing region
JCL	job control language
JDBC	Java database connectivity
JDK	Java Development Kit
JES	job entry subsystem (JES2 or JES3)
JMP	Java message processing region
JNDI	Java Naming and Directory Interface
JVM	Java Virtual Machine
KBLA	Knowledge Based Log Analysis
KSDS	key sequenced data set
LE	Language Environment
LMOD	load module
LPA	link pack area
LPAR	logical partition
LTERM	logical terminal
LU	logical unit
LU2	Logical Unit 2
MCS	multiple consoles support
MD BMP	message driven BMP
MFS	message format services
MLPA	modifiable link pack area
MNPS	Multi-Node Persistent Sessions
MOD	message output descriptor (MFS)
MOD	module (SMP/E)
MPP	message processing program
MPR	message processing region
MSC	multiple systems coupling
MSDB	Main Storage Data Base
MVS	Multiple Virtual System
NCCF	network communication control facility
NCP	network control program
NMD BMP	non-message driven BMP
ODBA	open database access
OLDS	online log data set
OLR	integrated HALDB online reorganization
OM	Operations Manager
ORS	Online Recovery Service
OSAM	overflow sequential access method
OTMA	open transaction manager access
OTMA C/I	OTMA callable interface
PCB	program communication block
PDS	partitioned data set
PDSE	partitioned data set extended
PDU	partition definition utility
PHDAM	partitioned HDAM
PHIDAM	partitioned HIDAM
PIA	package input adapter
PIC	Product Introduction Centre
PITR	point in time recovery
PMR	problem management record
PPT	program properties table
PS EDI	physical sequential enhanced data integrity (DFSMS)
PSB	program specification block
PSBW	PSB work pool
PSINDEX	partitioned secondary index
PSP	preventive service planning
PST	program specification table
PST	partition specification table
PTF	program temporary fix
QCF	IMS Queue Control Facility
QPP	Quality Partnership Program
RACF	Resource Access Control Facility
RAP	root anchor point
RAS	resource access security

RBA	relative byte address
RDS	restart data set
RDS	remote database services
RIM	related installation material
RLDS	recovery log data set
RLT	recovery level tracking (RSR)
RM	Resource Manager
RMF™	Resource Measurement Facility
RMI/IIOP	remote method invocation/Internet inter-ORB protocol
RNR	Rapid Network Recovery
RRS	Resource Recovery Service
RRSAF	resource recovery service attachment facility
RSR	Remote Site Recovery
RSU	recommended service upgrade
RYO	roll your own
SAF	Security Authorization Facility
SCI	structured call interface
SDM	System Data Mover
SDSF	spool display and search facility
SLDS	system log data set
SMB	scheduler message block
SMP/E	System Modification Program/Extended
SMQ	shared message queues
SMS	system managed storage
SMU	security maintenance utility
SOAP	simple object access protocol
SPE	small programming enhancement
SPOC	single point of control
SQL	structured query language
SRDS	structure recovery data set
SSA	sub-system alias
SSID	subsystem ID
SSL	secure sockets layer
SSOB	subsystem options block
STM	sysplex terminal management
SVC	supervisor call
SVL	Silicon Valley Laboratories
SVSO	shared VSO
TAB	transaction anchor block
TB	twin backward (pointer)
TCB	task control block
TCO	time controlled operations
TCP/IP	Transmission Control Protocol/Internet Protocol
TMS	Transport Manager System
TPNS	Teleprocessing Network Simulator
TSO	Time Sharing Option
UDB	Universal Database (DB2)
UOR	unit of recovery
UOW	unit of work
USS	Unix System Services
USS	unformatted system services (SNA)
WADS	write ahead data set
WAS	WebSphere Application Server
VG	variable gathering (IVP)
VLF	virtual lookaside facility
VOLSER	volume serial (number)
WSADIE	WebSphere Application Developer Integration Edition
VSAM	virtual storage access method
VSCR	Virtual Storage Constraint Relief
WSDL	Web Service Description Language
WSIF	Web Service Invocation Framework
VSO	Virtual Storage Option (DEDB VSO)
VTAM	virtual telecommunication access method
WTO	write-to-operator
WTOR	write-to-operator-reply
WWW	World Wide Web
XML	eXtensible Markup Language
XRC	eXtended Remote Copy
XRF	eXtended Recovery Facility

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 314. Note that some of the documents referenced here may be available in softcopy only.

- *IMS e-business Connectors: A Guide to IMS Connectivity,* SG24-6514
- *Ensuring IMS Data Integrity Using IMS Tools,* SG24-6533
- *IMS Installation and Maintenance Processes, SG24-6574*
- *IMS Version 8 Implementation Guide - A Technical Introduction of the New Features,* SG24-6594
- *IMS DataPropagator Implementation Guide,* SG24-6838
- *Using IMS Data Management Tools for Fast Path Databases,* SG24-6866
- *IMS in the Parallel Sysplex, Volume I: Reviewing the IMSplex Technology,* SG24-6908
- *IMS in the Parallel Sysplex, Volume II: Planning the IMSplex,* SG24-6928
- *IMS in the Parallel Sysplex, Volume III: IMSplex Implementation and Operations,* SG24-6929
- *The Complete IMS HALDB Guide, All You Need to Know to Manage HALDBs,* SG24-6945
- *Reorganizing Databases Using IMS Tools - A Detailed Look at the IBM IMS High Performance Tools,* SG24-6074
- *IMS Version 9 Implementation Guide - A Technical Overview,* SG24-6398

Other publications

These publications are also relevant as further information sources:

- *IMS Version 9: Administration Guide: Database Manager,* SC18-7806
- *IMS Version 9: Administration Guide: System,* SC18-7807
- *IMS Version 9: Administration Guide: Transaction Manager,* SC18-7808
- *IMS Version 9: Application Programming: Database Manager,* SC18-7809
- *IMS Version 9: Application Programming: Design Guide,* SC18-7810
- *IMS Version 9: Application Programming: EXEC DLI Commands for CICS and IMS,* SC18-7811
- *IMS Version 9: Application Programming: Transaction Manager,* SC18-7812
- *IMS Version 9: Base Primitive Environment Guide and Reference,* SC18-7813
- *IMS Version 9: Command Reference,* SC18-7814
- *IMS Version 9: Common Queue Server Guide and Reference,* SC18-7815
- *IMS Version 9: Common Service Layer Guide and Reference,* SC18-7816

- *IMS Version 9: Customization Guide,* SC18-7817
- *IMS Version 9: Database Recovery Control (DBRC) Guide and Reference,* SC18-7818
- *IMS Version 9: Diagnosis Guide and Reference,* LY37-3203
- *IMS Version 9: Failure Analysis Structure Tables (FAST) for Dump Analysis,* LY37-3204
- *IMS Version 9: IMS Connect Guide and Reference,* SC18-9287
- *IMS Version 9: IMS Java Guide and Reference,* SC18-7821
- *IMS Version 9: Installation Volume 1: Installation Verification,* GC18-7822
- *IMS Version 9: Installation Volume 2: System Definition and Tailoring,* GC18-7823
- *IMS Version 9: Master Index and Glossary,* SC18-7826
- *IMS Version 9: Messages and Codes Volume 1,* GC18-7827
- *IMS Version 9: Messages and Codes Volume 2,* GC18-7828
- *IMS Version 9: Open Transaction Manager Access Guide and Reference,* SC18-7829
- *IMS Version 9: Operations Guide,* SC18-7830
- *IMS Version 9: Release Planning Guide,* GC17-7831
- *IMS Version 9: Summary of Commands,* SC18-7832
- *IMS Version 9: Utilities Reference: Database Manager and Transaction Manager,* SC18-7833
- *IMS Version 9: Utilities Reference: System,* SC18-7834
- *IMS Connector for Java 2.2.2 and 9.1.0.1 Online Documentation for WebSphere Studio Application Developer Integration Edition 5.1.1,* SC09-7869
- *z/OS V1R5 Communications Server: SNA Network Implementation,* SC31-8777
- *z/OS V1R5.0 Security Server RACF System Programmer's Guide,* SA22-7681

Online resources

These Web sites and URLs are also relevant as further information sources:

- IMS home page:

  `http://www.ibm.com/ims`
- DB2 Information Management Software Information Center for z/OS Solutions

  `http://publib.boulder.ibm.com/infocenter/dzichelp/index.jsp`
- IBM Redbooks:

  `http://www.ibm.com/redbooks`

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IMS Version 9 Implementation Guide: A Technical Overview

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

IMS Version 9 Implementation Guide

A Technical Overview

Get familiar with the HALDB online reorganization feature

See the possibilities of these new technologies - IMS Java and XML

Explore all the new features of IMS Version 9

This IBM Redbook provides an overview of the new functions and enhancements in IBM Information Management System (IMS) Version 9. IMS Version 9 contains over 50 enhancements, in all areas of the product. that address availability, scalability, capacity, usability, manageability, operations, on demand business, and application development requirements from its huge customer base. All of the enhancements in this new version resulted from specific customer requirements. The major enhancements include:

- Improved database availability for HALDB databases is provided by the long-awaited online reorganization (OLR) enhancement.
- The functionality of the IMS Connect product is included as an integrated function of IMS.
- DBRC provides a new API that allows assembler programs to query the RECON.
- IMS Java users will have new JDBC SQL calls and DLI Model Utility support for retrieving existing IMS data in XML format as well as storing, indexing, searching, and retrieving valid XML documents into new or existing IMS databases.

We describe these enhancements and all other major changes in IMS Version 9. This redbook gives you the essential information that you need when you are migrating your IMS systems to IMS Version 9.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6398-00 ISBN 0738491578