

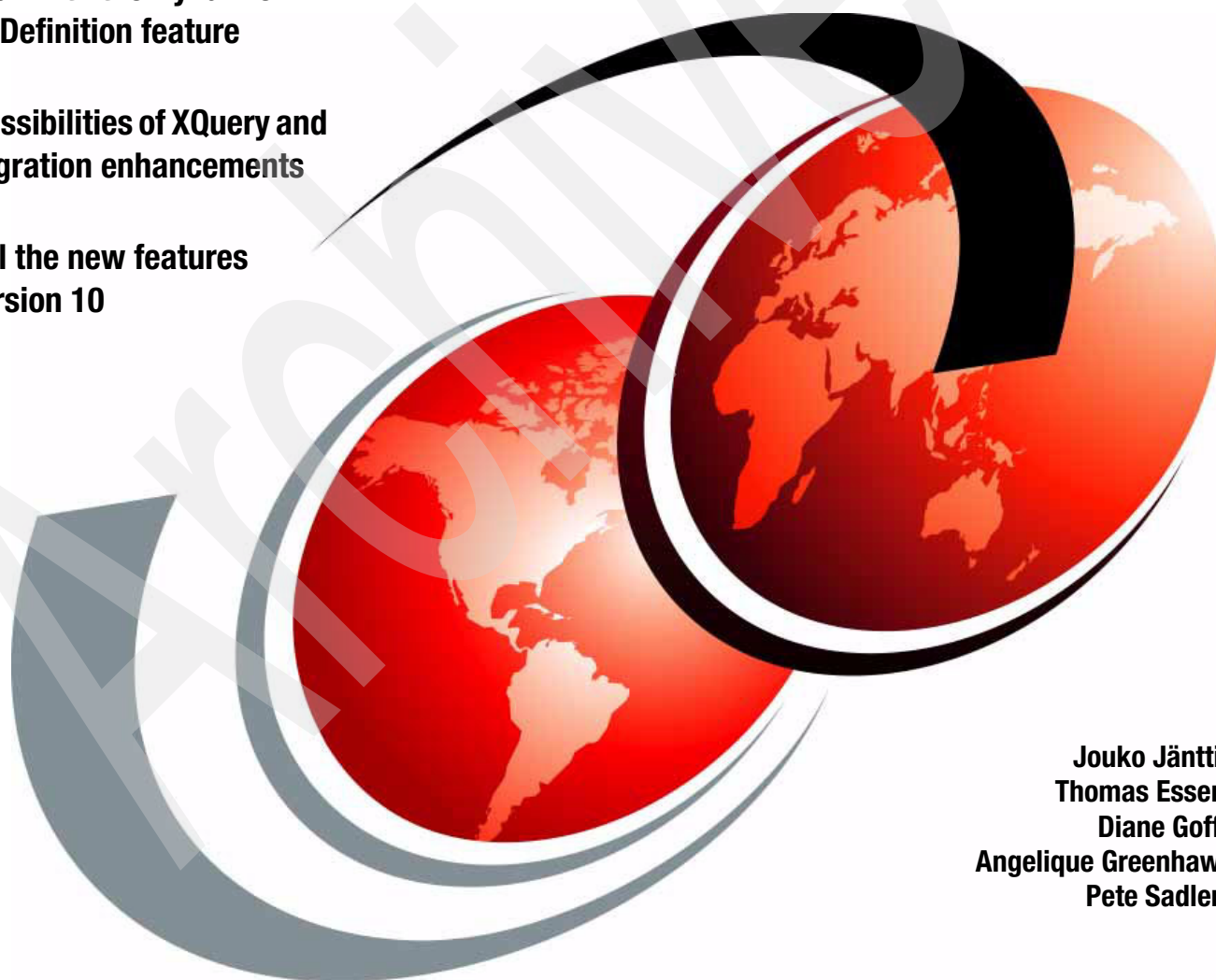
IBM IMS Version 10 Implementation Guide

A Technical Overview

Get familiar with the Dynamic
Resource Definition feature

See the possibilities of XQuery and
other integration enhancements

Explore all the new features
of IMS Version 10



Jouko Jäntti
Thomas Esser
Diane Goff
Angelique Greenhaw
Pete Sadler

Redbooks



International Technical Support Organization

IBM IMS Version 10 Implementation Guide

December 2007

Archived

Note: Before using this information and the product it supports, read the information in “Notices” on page ix.

First Edition (December 2007)

This edition applies to IMS Version 10 (product number 5635-A01) 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 book for more current information.

Contents

Notices	ix
Trademarks	x
Preface	xi
The team that wrote this book	xi
Become a published author	xiii
Comments welcome	xiv
Chapter 1. Overview of IMS Version 10 enhancements	1
1.1 IMS Version 10 product positioning	2
1.2 IMS Version 10 highlights	3
1.3 Database Manager enhancements	3
1.4 Transaction Manager enhancements	5
1.5 System enhancements	8
Chapter 2. System enhancements	13
2.1 Security in IMS Version 10	14
2.1.1 Startup parameters	15
2.1.2 Commands	15
2.1.3 Documentation changes	16
2.1.4 Replacing SMU security	17
2.1.5 SECURITY macro	17
2.1.6 RACF and IMS Version 10	18
2.1.7 IMSplex and earlier IMS releases	18
2.1.8 Security enhancements in IMS Version 10	18
2.1.9 Security auditing improvements	19
2.1.10 Enhanced security for IMS conversations	21
2.1.11 Security and DBRC	22
2.2 Analysis utilities	22
2.2.1 DFSILTA0	22
2.2.2 DFSISTS0	23
2.3 BPE external trace	25
2.3.1 BPE configuration	26
2.3.2 BPE commands	26
2.3.3 Performance impacts	27
2.3.4 New messages	27
2.4 Syntax Checker	28
2.4.1 Invocation	28
2.4.2 Support for PROCLIB members	29
2.4.3 Support for comments	30
2.4.4 New DISPLAY formats	31
2.4.5 New “Sel” field options	32
2.4.6 New pull-down options	35
2.4.7 New member selection panel	35
2.5 Enterprise Workload Management (EWLM)	36
2.5.1 Workload Manager and IMS	36
2.5.2 Virtualization Engine	36
2.5.3 How to implement EWLM to IMS Version 10	38

2.6	IMS Abend Search and Notification	39
2.6.1	Functionality and benefits	39
2.6.2	Prerequisites	41
2.6.3	Installation	41
2.6.4	Customization components	45
2.6.5	Summary	47
2.7	Knowledge-Based Log Analysis (KBLA)	47
2.7.1	Invoking KBLA	48
2.8	Large sequential data set support	48
2.9	Online change enhancements	50
2.9.1	ACBLIB member online change	50
2.9.2	Online change enhancement for work in progress	60
2.9.3	Summary of online change enhancements	60
2.10	Virtual storage constraint relief	61
2.11	XRF and FDBR enhancements	62
Chapter 3.	Database enhancements	65
3.1	ACBGEN enhancements	66
3.1.1	EPCB pool sizing information added to ACBGEN output	66
3.1.2	ACBGEN exploitation of storage above the 16 MB line	66
3.2	HALDB index/ILDS rebuild utility free space enhancement	67
3.3	Image copy enhancements	68
3.3.1	Fuzzy user image copy support	68
3.3.2	Flash copy support for Image Copy 2 (IC2)	70
3.3.3	DFSMSdss SET PATCH support for Image Copy 2	74
3.4	Sort enhancements of utilities	74
Chapter 4.	Transaction Manager enhancements	75
4.1	APPC enhancements	76
4.1.1	APPC timeout granularity	76
4.1.2	/LOCK and /UNLOCK command support	77
4.1.3	Local LU support	77
4.2	Disable support of BTAM	81
4.3	Multiple Systems Coupling (MSC) enhancements	82
4.3.1	Removal of exits replaced by DFSMSCE0	83
4.3.2	MSC bandwidth capability	83
4.3.3	MSC performance results	85
4.3.4	MSC link statistics	87
4.3.5	VTAM Generic Resources support on MSC links	92
4.3.6	Migration considerations for MSC	94
4.4	Sysplex serialized program management	94
4.5	Transaction level statistics enhancement	96
4.6	Fast Path enhancements	99
Chapter 5.	Manageability enhancements	105
5.1	Operations Manager enhancements	106
5.1.1	OM audit trail	106
5.1.2	Unsolicited output support	110
5.2	IMS systems management enhancements	113
5.2.1	Global status	113
5.2.2	IMSplex-wide parameter support	113
5.2.3	New and enhanced type-2 commands	114
5.2.4	Secondary master terminal enhancements	117

5.3	/DIS SHUTDOWN STATUS command enhancement	118
5.4	Enhanced display of system parameters	119
5.5	TSO SPOC enhancements	120
5.5.1	Support for viewing the OM audit trail	120
5.5.2	Batch SPOC utility	121
5.5.3	TSO SPOC action bar	123
5.5.4	TSO SPOC support for dynamic resource definition	125
5.5.5	New TSO SPOC preferences	125
5.5.6	New help capabilities	126
5.5.7	TSO SPOC input user exit support	126
5.5.8	TSO SPOC invocation command enhancements	127
5.6	REXX API enhancements	128
5.6.1	Support for OM unsolicited output messages	128
5.6.2	The REXX XML parser	130
Chapter 6.	Dynamic Resource Definition	133
6.1	IMS resource definitions	134
6.1.1	Managing resource definitions without DRD	134
6.1.2	DRD: A lightweight way to manage resource definitions	135
6.2	Prerequisites for DRD	136
6.2.1	Install the Common Service Layer	136
6.2.2	Allocate and define resource definition data sets	136
6.3	Enabling DRD	139
6.3.1	Define DRD parameters in DFSDFxxx PROCLIB member	139
6.3.2	Select OM applications to issue commands	143
6.3.3	Cold start IMS with DRD enabled	144
6.4	Using Dynamic Resource Definition	144
6.4.1	Descriptors	144
6.4.2	DRD commands	145
6.4.3	Importing and exporting resource definitions	149
6.4.4	Recovering runtime resource definitions when using DRD	150
6.4.5	RDDS Extraction utility (DFSURDD0)	150
6.4.6	Manage Resources application	153
6.5	DRD availability benefits summary	156
Chapter 7.	DBRC enhancements	157
7.1	Parallel RECON access	158
7.1.1	Non-parallel RECON access	158
7.1.2	Parallel RECON access	159
7.1.3	DFSMSvts	159
7.1.4	Parallel RECON access definition and setup	162
7.1.5	LIST.XXX output	172
7.1.6	Parallel RECON access performance considerations	174
7.1.7	Parallel RECON access recovery	177
7.1.8	Summary of serial compared to parallel processing	178
7.1.9	Migration and coexistence considerations for parallel RECON access	179
7.1.10	DBRC serviceability enhancements	180
7.2	DBRC API-2	183
7.2.1	Alternate RECON and IMS DD names	183
7.2.2	Alter RECON update capability	184
7.2.3	Register as a subsystem	186
7.2.4	Database authorizations	187
7.2.5	QUERY enhancements	188

7.2.6 Security enhancements	190
7.2.7 Miscellaneous enhancements	191
7.3 Timestamp precision	192
7.3.1 DBRC command and message output	194
7.4 Read-only RECON	197
7.5 DBRC migration	198
Chapter 8. Connectivity enhancements	203
8.1 OTMA enhancements	204
8.1.1 Routing enhancements	204
8.1.2 Message flood detection and control	208
8.1.3 Send-then-Commit (CM1) ACK timeout control	209
8.1.4 TPIPE storage cleanup	211
8.1.5 Member level security	211
8.1.6 OTMA security refresh by user ID	212
8.1.7 OTMA asynchronous message enhancements	212
8.1.8 OTMA /DISPLAY command enhancements	216
8.1.9 New OTMA startup option	217
8.1.10 Commit-then-Send (CM0) Ignore Purge support	217
8.2 OTMA enhancement migration considerations	218
8.3 OTMA enhancement benefits	218
8.4 IMS Connect enhancements	219
8.4.1 ACEE aging value support	219
8.4.2 Client password change request	220
8.4.3 RACF mixed case password	221
8.4.4 Send-then-Commit (CM1) ACK timeout control	221
8.4.5 Message flood control	222
8.4.6 Asynchronous message enhancements	222
8.4.7 Resume TPIPE enhancements	222
8.4.8 XML Adapter support	225
8.4.9 IMS SOA Composite Business Application support	230
8.5 IMS Connect enhancements benefits	234
Chapter 9. Integration enhancements	235
9.1 IMS TM Resource Adapter	236
9.1.1 Pre-IMS Version 10 enhancements	236
9.1.2 Enhancements for IMS Version 10	238
9.2 IMS DB Resource Adapter (IMS JDBC connector)	243
9.2.1 DB2 Java Stored Procedures returning DB2 result sets	244
9.2.2 GSAM metadata support	245
9.3 DLIModel Utility	247
9.3.1 Java tooling	252
9.3.2 XML schemas	252
9.4 IMS SOAP Gateway	254
9.4.1 IMS SOAP Gateway enhancements for IMS Version 10	256
9.5 XML database and XQUERY	259
9.5.1 IMS DB Resource Adapter XML API	259
9.5.2 IMS DB Resource Adapter XQuery API	260
9.5.3 Migration considerations for XQuery	261
Chapter 10. Installation and migration considerations	263
10.1 Hardware and software requirements	264
10.2 Hardware requirements	264
10.2.1 System console	264

10.2.2	Tape units	264
10.2.3	Coupling facilities	264
10.2.4	Direct access storage devices (DASD)	265
10.2.5	Large sequential data set support	265
10.2.6	Multiple Systems Coupling	265
10.2.7	Parallel RECON access	266
10.2.8	Terminals that are supported by IMS Version 10	266
10.2.9	Remote Site Recovery	266
10.3	Software requirements	266
10.3.1	Packaging, prerequisites, and coexistence	266
10.3.2	Supported connections	267
10.3.3	Global Online Change coexistence	268
10.3.4	OTMA coexistence SPEs	268
10.3.5	IMS Connect coexistence	268
10.3.6	Other coexistence maintenance	269
10.3.7	Remote Site Recovery (RSR) coexistence	270
10.3.8	Log records	270
10.3.9	Knowledge Based Log Analysis coexistence and migration	270
10.3.10	RECON upgrade	274
10.3.11	IMS utilities coexistence	274
10.4	IMS library changes	275
10.5	IMS IVP enhancements	277
10.5.1	IVP enhancements for existing functions	278
10.5.2	IVP support for new functions	281
10.5.3	Miscellaneous	284
10.6	System definition and execution parameter changes	284
10.6.1	Transaction scheduling	285
10.6.2	FPCTRL system definition macro eliminated	285
10.6.3	DFSDFxxx PROCLIB member	286
10.6.4	Enhanced display of system parameters	287
10.7	Installation and migration tasks	288
10.7.1	Remote Site Recovery (RSR) migration	290
Appendix A. Tools support for IMS Version 10		291
IMS tools migration and coexistence considerations		292
10.8	IMS tool enhancements	295
10.8.1	IMS database administration tools	295
10.8.2	IMS application management tools	299
10.8.3	IMS performance management tools	300
10.8.4	IMS recovery management tools	302
10.8.5	IMS information integration management tools	304
10.8.6	IMS utilities management tools	306
10.8.7	Fast Path utilities management	306
10.8.8	Full function utilities management	307
10.8.9	Administration utilities management	309
10.8.10	IMS TM management tools	310
10.8.11	Miscellaneous IMS Tools	312

Abbreviations and acronyms	313
Related publications	317
IBM Redbooks publications	317
Other publications	318
Online resources	319
How to get IBM Redbooks publications	319
Help from IBM	319
Index	321

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 illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

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

Redbooks (logo) ®

z/Architecture®

z/OS®

zSeries®

z9™

AIX®

BookManager®

CICS®

DataPropagator™

DB2®

DFS™

DFSMS™

DFSMSdfp™

DFSMSdss™

DS8000™

Enterprise Storage Server®

Enterprise Workload Manager™

ESCON®

FlashCopy®

FICON®

IBM®

IMS™

Lotus Notes®

Lotus®

MQSeries®

MVS™

Notes®

OMEGAMON®

OS/390®

Parallel Sysplex®

ProductPac®

Rational®

Redbooks®

RACF®

RAMAC®

REXX™

RMF™

S/390®

Sysplex Timer®

System z™

System z9™

System/370™

SystemPac®

Tivoli®

TotalStorage®

Virtualization Engine™

VTAM®

WebSphere®

The following terms are trademarks of other companies:

EJB, Java, JDBC, JDK, JRE, JSP, JVM, J2EE, RSM, and all Java-based trademarks are 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.

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

Preface

This IBM® Redbooks® publication provides an overview of the new functions and enhancements in IBM Information Management System (IMS™) Version 10. IBM continues to enhance IMS integration, manageability, and scalability.

IMS helps you with On Demand Business enablement, growth, availability, and systems management that current and newer environments and cost measures require. IMS Version 10 helps in addressing your On Demand Business needs through integration/openness, manageability, and scalability, providing:

- ▶ Integration with other products and platforms across the Internet, supporting open standards that benefit you, and taking best advantage of the latest industry tooling for application development and connectivity
- ▶ Manageability in staging users to autonomic computing, easing installation and use, eliminating and reducing outages, and minimizing the education curve for users of IMS
- ▶ Scalability with virtualization in assuring flexibility for growth and expansion in a heterogeneous environment while utilizing the latest hardware and software facilities to optimize performance, capacity, availability, and recovery

This book gives you the essential information that you need when you are migrating your IMS systems to IMS Version 10.

The team that wrote this book

This book 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 and also works for the IBM Silicon Valley Laboratory as a member of the IMS Worldwide Advocate Team. During the years of 2001 to 2003, he was a Project Leader specializing in IMS with the IBM International Technical Support Organization, San Jose Center. He is the lead author of the IMS-related IBM Redbooks publications listed in “IBM Redbooks publications” on page 317.

Diane Goff is a Senior IT Specialist in the IBM IMS Advanced Technical Support organization. She has presented IMS topics at technical conferences and seminars. Before joining ATS, she worked with IMS clients and also provided System z™ hardware and software support.

Angelique Greenhaw joined IBM in 2000 after she graduated from Arizona State University with a Bachelors degree in Computer Information Systems. She spent six months in IMS test before moving to IMS Systems development, where she specialized in the Online Change function for the next six years. Angie contributed to new IMS functionality and also worked as a Level 3 service representative in the online change component area. She also spent three years as the IMS development representative at SHARE, a technical conference for IBM product users. In January of 2007, Angie joined IMS Advanced Technical Support as an IT Specialist, where she is a primary resource in the areas of IMS Dynamic Resource Definition and Online Change.

Pete Sadler is an independent consultant in the United Kingdom and now also teaches the IMS classes for RSM™ Technology, which provides high-quality, high-focus education and

training in z/OS®, CICS®, DB2®, IMS, the WebSphere® portfolio, and other IBM mainframe-related subjects. Pete has over 35 years of experience in IMS as an Instructor, Systems Engineer, Country Specialist, IMS Developer, and IMS Advocate during his 39 years with IBM. For those who suffered, he originated the IMS Mastery tests. He holds a degree in Chemical Engineering from the University of London and is a UK Chartered Engineer. His areas of expertise include anything to do with IMS and its associated products. He has written several IBM Redbooks publications about IMS and related products and also presented IMS topics at many Technical Conferences in the US and Europe.

Thomas Esser is a Technical Sales, DMS working for IBM Software Group in Germany. He has 15 years of experience in IMS, including database administration. He provides technical sales support and advice for several major clients in the telecommunications, financial, and insurance industries. His areas of expertise and responsibilities include IMS-related database tools.



Figure 1 IMS Version 10 team from left: JJ, Angie, Pete, Diane, and Thomas

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

Paolo Bruni
Emma Jacobs
Deanna Polm
Sangam Racherla
Sokkieng Wang
International Technical Support Organization, San Jose Center

Rich Conway
Bob Haimowitz
International Technical Support Organization, Poughkeepsie Center

Rose Levin
Dave Cameron
Dario D'Angelo
Sally Gehring
Dean Grover
Dean Hardwick
Terry Krein
John Langley
Kin Lau
Suling Li
Janna Mansker
Dean Meltz
Tom Morrison
Huan Nguyen
David Ormsby
Betty Patterson
Maria Querales
Richard Schneider
Patrick Schroeck
Sandy Sherrill
Alan Smith
Judy Tse
Greg Vance
Pedro Vera
Mark Ziebarth
IBM Silicon Valley Laboratory

Alan Cooper
Alison Coughtrie
Andrew Wilkinson
IBM UK

Special thanks to the authors of the IMS Version 10 education material that was used as the fundamental source for this book (and also for helping to compose and reviewing the book):

Rich Lewis
Ken Blackman
Diane Goff
Bill Stillwell
Suzie Wendler
IBM Americas Technical Support

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and client satisfaction. As a bonus, you will 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 books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- ▶ Use the online **Contact us** review IBM Redbooks publications form found at:

ibm.com/redbooks

- ▶ Send your comments in an e-mail to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Overview of IMS Version 10 enhancements

This chapter provides an overview of enhancements and new features of IMS Version 10. IMS is unsurpassed in database and transaction processing availability and speed. IBM continues to enhance IMS integration, manageability, and scalability.

At a glance, IMS Version 10 Database Manager enhancements provide:

- ▶ Integrated, open access with enhanced IMS/XML database support, XQuery access to IMS data, and broadened Java™ and XML tooling to ease development
- ▶ Manageability through staging users to autonomic computing with simplified dynamic resource definition, eased operations and systems management, integrated operations across subsystems/platforms, and enhanced security and serviceability
- ▶ Scalability with more parallelism in IMS Database Recovery and Sharing Control (DBRC), and improved performance/capacity in Fast Path, IMS High Availability Large Database (HALDB), and database utilities

IMS Version 10 Transaction Manager enhancements provide:

- ▶ Integrated, open access with enhanced XML and Web services connectivity, and broadened Java and XML tooling to ease development
- ▶ Manageability through staging users to autonomic computing with simplified dynamic resource definition, eased operations and systems management, integrated operations across subsystems/platforms, and enhanced security and serviceability
- ▶ Scalability with widened bandwidth for Multiple Systems Coupling (MSC) and improved system availability, performance, and capacity in workload balancing

In this chapter, we provide the list of IMS Version 10 enhancements and cross references to the chapters, where the main enhancements are described in more detail.

1.1 IMS Version 10 product positioning

IMS, the recognized database manager and transaction server in the industry, is critical to your business. Exclusive to the z/OS platform, IMS Version 10 complements DB2 and WebSphere servers for database and transaction management. All of these are strategic IBM products and continue to be enhanced.

IMS Database Manager utilizes hierarchical organization technology, whereas DB2 and some other systems utilize relational database technology. Hierarchical and relational databases have each continued to grow with their specific characteristics and different roles to play. Hierarchical is best used for mission-critical work and for work that requires the utmost in performance. Relational is best used for decision support.

Hierarchical databases can offer a significant performance edge over relational databases when queries are known beforehand. Query optimization for relational databases ensures good performance where the query is not known in advance. Each type is best at what it does. The products supporting these technologies are enhanced to address different application requirements and continue to overlap more and more in their capabilities. However, the product originally designed for a particular capability will inherently be the best.

Relational and hierarchical technologies can work together for optimum solutions. Users can efficiently store operational data in hierarchical form, which can be accessed easily by their favorite relational decision support tools, with minimal impact on the production hierarchical data. IMS data can be accessed directly or propagated and replicated with relational data for summarizing, enhancing, and mining. IBM provides standard application interfaces for accessing IMS, as well as other data. Both relational and hierarchical IMS data can be most efficiently accessed, together or independently, using the IMS Transaction Manager (TM) and WebSphere servers.

IBM continues to invest in providing these complementary solutions. Note that most IMS clients also have DB2.

IMS Transaction Manager and WebSphere servers are both strategic application managers and are enhanced to take advantage of each other. They each have inherently different characteristics. IMS is more efficient in application management, data storage, and data access but applies strict rules for this access. WebSphere makes it easier to serve the Web and integrate data that might have been less defined in advance. Thus, they play different roles in the enterprise. Clients use both application managers: WebSphere for newer Web-based applications and IMS for more mission-critical high performance/availability and low-cost/transaction applications and data.

IMS and WebSphere products together have been providing tools to make this combination an optimum environment. Using IMS Connect and IMS TM Resource Adapter, WebSphere development tooling can develop Web applications that can serve the Web and easily access existing and new mission-critical IMS applications. Using JDBC™ and IMS Open Database Access, WebSphere applications can also access IMS DB data directly.

IBM demonstrates its commitments to you by continuing to enhance IMS. Current investments have:

- ▶ Substantially improved database availability
- ▶ Established open access to IMS transactions
- ▶ Extended Parallel Sysplex® systems support

1.2 IMS Version 10 highlights

IMS Version 10 helps in addressing your On Demand Business needs through integration/openness, manageability, and scalability, providing:

- ▶ Integration with other products and platforms across the Internet, supporting open standards that benefit you, and taking best advantage of the latest industry tooling for application development and connectivity
- ▶ Manageability in staging users to autonomic computing, easing installation and use, eliminating or reducing outages, and minimizing the education curve for users of IMS
- ▶ Scalability with virtualization in assuring flexibility for growth and expansion in a heterogeneous environment while utilizing the latest hardware and software facilities to optimize performance, capacity, availability, and recovery

With IMS Version 10, IBM offers enhancements to both major components to ensure you have the growth, availability, and systems management that newer environments and cost measures require. In this overview chapter, we have grouped the various enhancements into the categories of integration/openness, manageability, and scalability for the IMS TM and IMS DB components, plus a systems section for enhancements that apply to both IMS TM and IMS DB.

1.3 Database Manager enhancements

Focusing on integration and openness, IMS Version 10 Database Manager includes:

- ▶ Full XQuery support

Supported through the existing IMS JDBC interface, IMS XQuery support provides access using standard XQuery expressions to IMS full function data, including new IMS XML data. By doing so, you can open up IMS to the emerging market of off-the-shelf, third-party tools, using a standard shareable integration point between IMS and other industry databases.

- ▶ XML DB mapping enhancements

XML database mapping enhancements enable expanded mappings between new or existing IMS databases and visualized XML documents or collections to widen the scope of supported XML documents for new IMS databases and to ease disparate data integration across the enterprise.

- ▶ IMS Database Service Data Objects (SDO) support

IMS Database Service Data Objects (SDO) support enables the service-oriented architecture (SOA) integration of IMS data in a language and database independent way. IMS SDO support enables the distributed use of IMS data in a disconnected mode (holding no locks in the database). This provides a way of representing IMS data as industry standard business objects, allowing IMS to transparently participate in SOA as a standard Service Component Architecture (SCA) component. This support is being provided through the IMS Version 10 service process.

For more information about these IMS Database Manager integration and openness enhancements, see Chapter 8, “Connectivity enhancements” on page 203 and Chapter 9, “Integration enhancements” on page 235.

Focusing on manageability, IMS Version 10 Database Manager includes:

- Database Image Copy 2 (IC2) utility enhancements

Image Copy 2 utility is enhanced as follows:

- Support for Fast Replication

The Image Copy 2 utility will support the Fast Replication capability of Data Set FlashCopy® on Enterprise Storage Servers (ESS) and SnapShot on RAMAC® Virtual Array (RVA) DASD subsystems for faster database image copy/restore processing.

- SET PATCH support

Users can customize DFSMSdss™ functionality using a new SET PATCH control statement as input to the Image Copy 2 utility. This new method provides an alternative to the “module zap” method of customizing DFSMSdss.

- Fuzzy user image copy support

Users can register concurrent (“fuzzy”) user image copies with DBRC. User image copies are those that are taken using a tool that does not have a DBRC interface. Previously, users could register only clean user image copies with DBRC.

- HALDB Index/Indirect List Data Set (ILDS) Rebuild free space enhancement

IMS Version 10 enhances the HALDB Index/ILDS Rebuild utility (DFSPREC0) by introducing a free space option that rebuilds HALDB’s indirect list data sets (ILDS) in VSAM load mode. This free space option can improve the performance of DFSPREC0 and any subsequent reorganizations and recoveries of HALDB partitions by eliminating CI and CA splits in a newly rebuilt ILDS.

- Large sequential data set support

IMS Version 10 will support the z/OS facility to remove the size limit of 64 K tracks per volume for sequential data sets for BSAM, QSAM, and EXCP to reduce the need to use multiple volumes for single data sets and to relieve capacity restrictions. For IMS, this enhancement applies to GSAM, BSAM, and OSAM data sets.

For more information about these IMS Database Manager manageability enhancements, see Chapter 2, “System enhancements” on page 13 and Chapter 3, “Database enhancements” on page 65.

Focusing on scalability, IMS Version 10 Database Manager includes:

- DBRC enhancements

DBRC enhancements in IMS Version 10 are:

- Parallel RECON access

Prior to IMS Version 10, when multiple DBRCs accessed a shared RECON data set, all interactions with that RECON data set were performed serially, which negatively impacted performance. In IMS Version 10, users can enable parallel access to the RECON data. DBRC uses the Transactional VSAM function of DFSMS™ to provide this parallel processing of the RECON data set. Parallel RECON access is optional and is available only in an IMSplex environment.

- DBRC application programming interface (API) enhancements

DBRC API enhancements allow the RECON data set to be updated, as well as read, and to secure the RECON by limiting the level of access to individual users. This further enhances the API that was first introduced in IMS Version 9 to provide users with a single release-independent interface to the RECON.

- READONLY access to the RECON data sets

READONLY RECON access to the RECON data sets is now supported for the DBRC utility (DSPURX00) and the DBRC API. SAF access to the RECONS for these types of access can now be READ. Additionally, the previous requirement that all users have CONTROL access to the RECONS is removed. Users can now be assigned one of three levels of access: READ, UPDATE, or ALTER.

- DBRC timestamp precision enhancements

DBRC timestamp precision enhancements provide full DBRC support of 12-byte timestamps with microsecond precision. This increased precision allows more precise recording of IMS events and conforms exactly to the designed resolution in Time of Day (TOD) clock values.

- Sort enhancement for Change Accumulation and Prefix Resolution utilities

This sort enhancement increases the core size available to OEM sort utilities called by IMS Change Accumulation and Prefix Resolution. The default value of the CORE= parameter has been changed from 200 K to MAX, which does not place a limit on the amount of main storage that a sort/merge utility uses, thus increasing the available number of sort work data sets, enabling greater efficiency in change accumulation and prefix resolution and improving performance.

- Fast Path enhancements

Fast Path enhancements include an increase in the maximum number of buffers that can be defined for the DEDB buffer pool and an increase in the number of Fast Path output threads. Additional enhancements for Data Entry Database (DEDB) users include the ability to start the DEDB and all of its areas with a single command. There is also an option for not unloading the randomizer when /DBR DB is used to eliminate the need to start the areas of a DEDB individually and improve ECSA usage. For Virtual Storage Option (VSO) users, automatic compression of the SVSO private pool at XRF tracking items helps enhance performance.

For more information about these IMS Database Manager scalability enhancements, see Chapter 3, “Database enhancements” on page 65 and Chapter 7, “DBRC enhancements” on page 157.

1.4 Transaction Manager enhancements

Focusing on integration and openness, IMS Version 10 Transaction Manager includes:

- IMS SOA Composite Business Application support

IMS SOA Composite Business Application support is a new capability for IMS TM Resource Adapter clients that invokes IMS conversational transactions. Although the IMS TM Resource Adapter has supported interactions with IMS conversational transactions, IMS Version 10 enhances this capability by allowing the iterations between a conversation to span multiple shareable sockets. IMS SOA Composite Business Application support expands conversational processing to environments with connection pooling, sysplex distributor, cloned application server, and supports the process choreography function of WebSphere Process Server.

- XML adapter support for COBOL

Together with the IMS SOAP Gateway and IBM WebSphere Developer for z (WDz), the XML Adapter support for COBOL enables reuse of IMS applications as Web Services, leveraging open standards and utilizing flexible tooling support. It also provides IMS

transaction interoperability with client applications, independent of location, programming language and platform, as well as HTTPS/SSL security enhancements.

- ▶ **IMS callout support**

IMS callout support enables IMS applications as clients, interoperating with business logic outside of the IMS environment. It provides for asynchronous callout to an external application (for example, a Web Service or a WebSphere application) through the IMS SOAP Gateway or IMS TM Resource Adapter. This support includes correlation mapping between the callout request and the external application, enhanced security, and assistance on destination routing.

- ▶ **DLIModel utility enhancements**

To simplify IMS metadata generation, the DLIModel utility will provide metadata generation support for GSAM databases. The DLIModel graphical user interface (GUI), an Eclipse plug-in, is a user-friendly interface for viewing and generating IMS metadata. The DLIModel utility fully supports the parsing of Program Specification Blocks (PSBs) and Database Definitions (DBDs), now including GSAM, and the importing of COBOL copybooks while eliminating any need to program control statements. It generates IMS Java metadata classes, IMS XML Schemas (with new annotations for XQuery support), or DBD and PSB XML metadata interchanges (PSB XMLs).

- ▶ **MFS Web support**

MFS Web support provides access to existing IMS applications from IBM WebSphere Application Server environments to help protect your existing investments. MFS Web Services enhancements expose attributes and extended attributed fields for Web Services applications. MFS Web Enablement now includes access from IBM WebSphere Application Server for z/OS, in addition to Windows® and IBM AIX®. Customizable display enhancements include drop-down list boxes and infopops on familiar look-and-feel 3270 Web pages.

- ▶ **IMS TM Resource Adapter PL/I application support**

The IMS TM Resource Adapter provides PL/I support using Rational® Application Developer tooling and WebSphere Application Servers to allow IMS PL/I applications to be enabled as Web Services without IMS application changes.

- ▶ **Multiple Systems Coupling (MSC) VTAM® Generic Resources (VGR) support**

MSC VTAM VGR support provides to a remote IMS the ability to use a generic resource node name to communicate with an IMS within an IMSplex through the MSC facility, thereby simplifying connectivity. A remote IMS can use a single MSC link definition to connect to the partner link in any of the IMSs in the IMSplex.

- ▶ **APPC enhancements**

APPC enhancements include:

- LOCK and /UNLOCK command support from both LU 6.2 devices and Open Transaction Manager Access (OTMA) clients
- Local LU support, which allows greater control for specifying which IMS LU name to associate with an asynchronous outbound message, as well as ensuring that applicable SAF security checks are invoked with the appropriate LU
- Enhanced timeout granularity to release hung resources within seconds

- ▶ **Open Transaction Manager Access (OTMA) enhancements**

OTMA enhancements include:

- For any one OTMA client, you can activate message flood detection by specifying a maximum number of input messages that can be waiting concurrently for processing.

- OTMA automatically removes transaction pipes (tpipes) after they are idle for three consecutive system checkpoints, if they are eligible for removal.
 - OTMA performs timeout processing for send-then-commit messages if an acknowledgement response is not received from an OTMA client within a period of time specified by the user.
 - You can stop incoming transactions and commands from individual OTMA clients using the /STOP TMEMBER command without losing the connection or any other communications between the OTMA client and OTMA.
 - You can specify security levels for each OTMA client independently, which means greater flexibility in tailoring OTMA security.
 - The existing /DISPLAY commands for OTMA now return information for both commit-then-send and send-then-commit messages.
 - Support for the /LOCK and /UNLOCK commands is provided from OTMA clients, as well as from Advanced Program-to-Program Communication (APPC) devices.
 - The OTMA descriptors are enhanced to allow clients to define OTMA routing information. This enhancement eliminates the need to code OTMA routing exits to perform this task.
 - OTMA introduces a new optional flag in the OTMA prefix so that the client using commit-then-send (CMO) transactions with multiple PURG calls in the TP program communication block (PCB) can ignore the PURG calls to generate one output message with multiple segments.
 - OTMA implements the ACEE aging value capability to refresh user ID ACEEs. This value will be checked when a user ID initiates a Resume TPIPE. If the OTMA aging value is exceeded, the user ID is considered expired and a RACROUTE is reissued for the TPIPE and user ID.
- **IMS Connect enhancements**
- IMS Connect enhancements ease usability by providing support for Commit Mode 1, message flood detection, alternate Client ID, and enforced client ID uniqueness. Security enhancements include support for mixed-case passwords, for password changes, and for an ACEE aging value. A performance enhancement introduces the new NODELAY configuration statement parameter to improve throughput.

For more information about these IMS Transaction Manager integration and openness enhancements, see Chapter 4, “Transaction Manager enhancements” on page 75 and Chapter 8, “Connectivity enhancements” on page 203.

Focusing on manageability, IMS Version 10 Transaction Manager includes:

- **Transaction level statistics**
- Transaction level statistics allow users to log application accounting information at the conclusion of each unit of work (commit) with logging done on a transaction-by-transaction basis at the discretion of the user.
- **Fast Path enhancements**
- Fast Path (FP) enhancements include improvements to Expedited Message Handler (EMH), increased buffers, and additional availability/usability items. FP EMH enhancements add command support to reset FP response mode for static nodes and dynamic users to free up hung terminals in Fast Path Response mode without requiring an IMS restart.
- **Disable support for BTAM terminals**

IMS Version 10 does not support BTAM terminals. If any IMS macro statements specify BTAM terminals during stage 1 of system definition, IMS issues a system definition warning message, G411, with a severity code of 2. System definition continues, and IMS ignores references to BTAM terminals.

- ▶ Removal of routing exits that were replaced by DFSMSCE0 (MSC Message Routing and Control User Exit)

IMS Version 9 was the last release in which the MSC user routing exits (DFSCMTR0, DFSCMLR0, DFSCMPR0, and DFSCNPRT0) can be used. DFSMSCE0 must be used in IMS Version 10.

- ▶ Removal of the Security Maintenance Utility (SMU)

IMS Version 9 was the last release to support the Security Maintenance Utility (SMU). Clients using SMU need to migrate to RACF® or an equivalent product with IMS Version 10.

For more information about these IMS Transaction Manager manageability enhancements, see Chapter 4, “Transaction Manager enhancements” on page 75.

Focusing on scalability, IMS Version 10 Transaction Manager includes:

- ▶ Multiple Systems Coupling (MSC) bandwidth

MSC bandwidth enhancements use an improved blocking technique to increase performance response times for transaction and output message throughput. Enhancements also allow for larger link buffers with dynamic change of buffer sizes, inclusion of responses in the buffer, and sending with a single send/write. Detailed statistics are provided to ascertain performance and make adjustments.

For more information about these IMS Transaction Manager scalability enhancements, see Chapter 4, “Transaction Manager enhancements” on page 75.

1.5 System enhancements

Focusing on integration and openness, IMS Version 10 includes the following system enhancements:

- ▶ IMS Java library support

IMS Java library support enhances DB2 result sets and GSAM usage, enabling you to return IMS data to the stored procedure client encapsulated within a DB2 result set, with the conversion handled by the Java libraries.

The Java libraries are also enhanced to support GSAM databases consistent with other IMS database types, which allows all of the supported data types to be stored into and read from a GSAM database record, and fully leverage the existing Java library built-in data conversion routines.

For more information about these IMS system integration and openness enhancements, see Chapter 9, “Integration enhancements” on page 235.

Focusing on manageability, IMS Version 10 includes the following system enhancements:

- ▶ Dynamic Resource Definition (DRD)

New commands enable you to dynamically add, change, and delete resources (databases, programs, transactions, and routing codes) without having to go through the IMS system definition or online change (OLC) process. QUERY commands display both the definitional attributes and the status of resources. The Destination Creation user exit

(DFSINSX0) has been enhanced to create transactions for scheduling, not just queuing, along with programs. MSC resources can be dynamically updated and queried. The IMS TSO SPOC has been enhanced with a new user interface that can be used to manage IMS resources.

- ▶ **ACBLIB member online change**

ACBLIB member online change provides for individual ACBLIB members to be added or changed (though not deleted) without a library switch, which earlier required a full online change (OLC). Only the resources associated with the change will be quiesced. This function coexists with the existing full library switch OLC capability. This enhancement applies to Global Online Change.

- ▶ **Online change commit processing enhancement**

The online change COMMIT process has been changed so that indirectly affected transactions will no longer cause a commit to fail. Indirectly affected transactions are those with transactions on the queue, which have no attributes on the TRANSACT macro that are being changed by the online change. This enhancement has previously been available for the Shared Queues environment; IMS Version 10 adds support for the local queue (non-Shared Queues) environment.

- ▶ **Global online change usability enhancement**

You can allocate the OLCSTAT data set dynamically in IMS Version 10, which addresses the situation where a long-running multi-step batch job allocates all data sets up front, including the OLCSTAT data set, which effectively prevents global online change operations from occurring. With this new support, the OLCSTAT data set is allocated only while it is being used and is not allocated for the duration of the batch job. A batch job can continue to define the OLCSTAT data set with an OLCSTAT DD statement or define a dynamic allocation member for the OLCSTAT data set.

- ▶ **Base Primitive Environment (BPE) external trace table enhancements**

The BPE trace enhancements enable the contents of the BPE trace table to be written to external media to preserve them for improved serviceability and problem determination. You can set trace parameters in the BPE configuration PROCLIB member, enabling you to trace your system over longer periods of time and to increase your likelihood of obtaining the right diagnostic data the first time.

- ▶ **Abend Search and Notification facility**

Abend Search and Notification (ASN) helps reduce problem determination time by providing real-time automatic e-mail or mobile text message (SMS) notification upon the occurrence of an abnormal termination. The e-mail notification contains hyperlinks (URLs) to IBM-supplied Internet resources for understanding, analyzing, and resolving problems.

- ▶ **Knowledge-Based Log Analysis (KBLA) enhancements**

KBLA enhancements provide for KBLA utility support of log record changes and the KBLA ISPF environment to generate JCL to accommodate IMS Version 9 and IMS Version 10 levels and support multiple concurrent releases of IMS.

- ▶ **Enterprise Workload Manager™ (EWLM) support**

EWLM support provides for workload management tracking across environments to integrate IMS with EWLM's end-to-end workload management. This enhancement flows EWLM Correlators with an IMS transaction, using an enhanced version of the existing workload manager services to pass correlators back to the EWLM.

- ▶ **Fast Database Recovery (FDBR) region enhancement**

The default allocation for Fast Path buffer pools used in FDBR regions is changed from extended common service area (ECSA) to extended private storage. The Fast Path buffer pools include the DEDB common buffer pool, as well as the SVSO buffer pools.

FPBUFF=LOCAL is now the default in the DFSFDRxx member of the IMS.PROCLIB data set.

► Syntax Checker enhancements

Syntax checker enhancements add new support and update existing support for currently supported members and add additional PROCLIB and field options, new panels, and new keyword display options.

► Installation verification program (IVP) enhancements

IVP enhancements are provided to support new IMS functions with step-by-step and menu-driven jobs and tasks that guide users to use and explore useful sample applications.

► Operations Manager enhancements

New Operations Manager functions leverage the z/OS logger to provide an input/output audit trail for IMS commands entered from a single point of control (SPOC). The user can configure this so that each OM uses its own log, or all OMs use the same merged log for all SPOC activity. These enhancements also allow the SPOC to display unsolicited messages and use the OM to track commands/responses.

► TSO SPOC enhancements

SPOC enhancements include:

– Batch SPOC utility

A new batch SPOC utility uses OM services to submit IMS operator commands to an IMSplex. The utility accepts any commands that are supported by the OM API. The batch SPOC utility is invoked using standard JCL statements.

– Input user exit routine support

The TSO SPOC executable, DFSSPOC, is enhanced to support input user exit routines. When a TSO SPOC session is started with an input user exit routine, the exit routine has the ability to modify or reject the command parameters, set return and reason codes, and send text messages to the TSO SPOC session

– ISPF user interface for dynamic resource definition (DRD)

New TSO SPOC panels have been added to support dynamic resource definition (DRD). There is also a new ISPF application to help you manage resources dynamically.

– TSO SPOC action bar

The TSO SPOC action bar has been enhanced to include new options for managing resources, working with individual resources, setting views, and setting optional preferences.

– New TSO SPOC preferences

New TSO SPOC preferences include one of two views for command display, how to clean up command responses, and command history and command response retention.

– New help capabilities

Additional help capabilities enable you to position the cursor on the column heading in QUERY display panels to retrieve field-specific help topics.

– TSO SPOC invocation command enhancements

New CMD() and TYPE(DISPLAY) parameters for the DFSSPOC command allow application programmers to issue IMS operator commands, display the command responses, and scroll through the data.

- ▶ Global status

The global status enhancement provides the ability for IMS to maintain global command status for databases, areas, and transactions in the IMS Resource Manager Resource Structure to be applied to new or restarting IMSs in a sysplex that were down when the command was issued. This provides an autonomic provision of a single system image among IMSs in a sysplex. A new PLEXPARM parameter determines how global status is kept in the IMSplex.

- ▶ Sysplex serialized program management (SSPM)

Sysplex serialized program management (SSPM) utilizes the Resource Manager (RM) to manage serial program scheduling across the IMSplex in a Shared Queues environment so that only one iteration of a serial program is running at a time in the IMSplex.

- ▶ Selectively display system parameters

This enhancement provides improved visibility of the startup parameters' values, shows only the active feature's systems parameters, and shows the actual value of each parameter after reading the log to help determine actual values for optimal configurations and user response times. During IMS initialization, message DFS1929I is issued twice.

- ▶ /DISPLAY SHUTDOWN STATUS command enhancement for ODBA threads

The /DISPLAY SHUTDOWN STATUS command has been enhanced to show active and dubious ODBA threads that can prevent IMS from shutting down completely if IMS is not shutting down as expected.

- ▶ Secondary master terminal enhancements

The /SMCOPY command has been enhanced to disable the logging of system messages to the secondary master. The Type-2 Automated Operator (AOI) exit routine (DFSABOE00) can now specify if messages are logged to the secondary master terminal on a message-by-message basis.

- ▶ Security enhancements

Security enhancements include RACF error message reduction, faster auditing, enhanced security for IMS conversations, support for mixed-case passwords, and the removal of SMU support.

- ▶ IMS restart enhancement

Two new parameters, MSG0826 and MSG0830, are added in the <DIAGNOSTICS_STATISTICS> section of the DFSDFxxx IMS.PROCLIB member to indicate if BLDL types of DFS826I and DFS830I should be issued or suppressed. Previously, these messages were always issued and caused the job logs to fill.

- ▶ Restart exit

A new restart exit is called during all types of IMS restarts. It is called at the beginning of restart processing (before the log is read) and at the end of restart processing (before the restart complete message is issued). The exit routine must be defined to IMS using the EXITDEF parameter in the USER_EXITS section of the DFSDFxxx member in the IMS.PROCLIB data set.

- ▶ Serviceability enhancements

IMS Version 10 contains the following serviceability enhancements:

- System memory dumps of the IMS system generally contain more address spaces than previous IMS versions. IMS Version 10 selects all available, related address spaces for a system memory dump, which for most dumps increases the number of address spaces that are included in the dump output.

- IMS Version 10 adds an in-memory shutdown trace table to help diagnose IMS shutdown problems.
 - Messages DFS064I and DFS065 are enhanced to include the name of the destination that could not be found or created and to include the name of the transaction or LTERM that was stopped, respectively.
 - The output for the Queue Manager Extension to the System Contents Directory (QSCD) control block has been improved when you use the IMS Interactive Dump Formatter to format an IMS system dump. IMS Version 10 also corrects the formatting for the Logging to Secondary Master Block (LSMB) when you use the IMS Interactive Dump Formatter to format an IMS system dump.
 - When you use KBLA or the record format and print module (DFSERA30) of the File Select and Formatting Print utility (DFSERA10) to directly format SNAP subrecord types of X'67xx' log records, control blocks in the record can span segments, which can cause byte misalignment in the formatted output. IMS Version 10 corrects the formatting of such records so that they retain the same byte alignment, even when they span segments, thus making the formatted output easier to read.
- Virtual storage constraint relief
- IMS Version 10 provides a new option to move blocks from CSA to 31-bit storage, allowing CICS and Open Database Access (ODBA) applications to use the larger PSBs without putting further pressure on below-the-line storage.

For more information about these IMS systems manageability enhancements, see Chapter 2, “System enhancements” on page 13 and Chapter 5, “Manageability enhancements” on page 105.

Focusing on scalability, IMS Version 10 includes the following system enhancements:

- Application control block (ACB) generation enhancement
- ACBGEN 31-bit enhancements utilize storage above the line to address problems with insufficient storage below the line that might occur when building an ACB with a large number of PCBs per PSB.
- Statistics analysis utilities rewrite
- The Log Transaction Analysis utility (DFSILTA0) and the Statistical Analysis utility (DFSISTS0) have been rewritten to support better performance, Shared Queues, and maintainability.

For more information about these IMS system scalability enhancements, see Chapter 2, “System enhancements” on page 13.

System enhancements

This chapter covers changes to IMS system definition and execution parameters that affect functions that were available in previous releases. Definition and execution changes for new functions are discussed where the new functions are explained.

The following topics are discussed:

- ▶ Security in IMS Version 10
- ▶ Analysis utilities
- ▶ BPE external trace
- ▶ Syntax Checker
- ▶ Enterprise Workload Management (EWLM)
- ▶ IMS Abend Search and Notification
- ▶ Knowledge-Based Log Analysis (KBLA)
- ▶ Large sequential data set support
- ▶ Online change enhancements
- ▶ Virtual storage constraint relief
- ▶ XRF and FDBR Enhancements

2.1 Security in IMS Version 10

IMS Version 9 was the last release to support the Security Maintenance utility (SMU) and application group name (AGN) security. To prepare for this change, IMS Version 9 allowed you to use Resource Access Control Facility (RACF) user exit routines and RACF-driven resource access security (RAS) to implement security functions that previously required the use of SMU. IMS Version 10 does not support SMU security at all; the code that supported SMU security has been taken out of IMS Version 10.

Note: RACF is an external security product to IMS, accessed by IMS using the Security Access Facility (SAF) interface. RACF is licensed with the IBM z/OS Security Server. Where this book directs you to use RACF, you can use a different, equivalent security product if you choose.

The IMS Version 9 security enhancements also eliminated the 65 535 terminal limit of SMU-defined sign-on verification security. There is no limit to the number of terminals that you can secure with RACF.

Table 2-1 shows the principle security enhancements introduced in IMS Version 9 and the SMU security functions that they replace.

Table 2-1 RACF replacements for SMU functions

SMU function	Replacement
AGN security for dependent regions	RAS security for dependent regions implemented by RACF security profiles and exit routines
Sign-on verification security	Sign-on verification security that is defined in the system definition macros
LTERM security of Time-Controlled Operations (TCO)-script-issued commands	RACF security of TCO-script-issued commands
MSC link security for transactions	RACF and exit-routine security for both direct and non-direct routed transactions that are received from an MSC link

The impact of removing SMU affects the IMS environment in several ways:

- ▶ All SMU specifications in any of the system definition macros are ignored:
 - `COMM OPTIONS= (<NOPSWD | PASSWD | FORPSW><,NOTERMNL | TERMINAL | FORCTERM>)`
 - `IMSGEN PSWDSEC=,TERMSEC=`
 - `SECURITY PASSWD=,TERMSEC=,TRANCMD=, TYPE=(<NOAGN | RACFAGN | AGNEXIT>)`
- ▶ The Online Change Utility will no longer support the IMS.MATRIX data set. If the MATRIX data set DD cards exist in the online change utility JCL, the utility will ignore those specific DD cards.

2.1.1 Startup parameters

The changes to the startup parameters fall into several categories:

- ▶ Parameter values requesting the use of SMU continue to be accepted in IMS Version 10 but are ignored.

These include AGN (specification of an AGN name), AOI1=S (request to use SMU security for AOI type-1 command authorization) and ISIS=<0 | 1 | 2>. The ISIS parameter is still supported for the specification of combinations of SAF and user exit authorization. Only now the three values of 0, 1, and 2, which were previously associated with SMU, are ignored.

Specifications that are no longer documented but are compatible with functionality in previous releases of IMS will be accepted for compatibility purposes and function in the same manner as the supported counterparts.

Certain parameter values in previous releases allowed the use of SAF/RACF along with negating the loading of the sign-on security tables from the MATRIX data set. The specifications included: TRN=<E | X>, RCF=<B | R>, and SGN=<D | E | W>. Although these values are no longer documented in *IMS Version 10: System Definition Reference*, GC18-9966, because they request negation of loading tables from the MATRIX data set that is no longer supported, the values will be accepted for compatibility purposes and function in the same manner as the supported counterparts.

- ▶ Specifications that have changed their meaning because the SMU capability no longer applies include some of the values that can be specified.

The MSCSEC parameter previously included values that allowed SMU security to be used for non-directed routing requests. In IMS Version 10, the parameter values specify the use of SAF/RACF or no security.

2.1.2 Commands

Certain commands, such as /CHANGE PASSWORD and /DELETE PASSWORD | TERMINAL were strictly associated with SMU security. These commands are no longer supported in IMS Version 10 and, if entered, will result in error message DFS181.

The /SET, /LOCK, and /UNLOCK commands now support SAF/RACF security and no longer support SMU security. The SAF/RACF enhancements in IMS Version 9 were:

- ▶ User ID access to transactions (TIMS/GIMS), PSBs (IIMS/JIMS), LTERMS (LIMS/MIMS), and databases (PIMS/QIMS)
- ▶ Based on specification of LOCKSEC=Y | N
- ▶ Password access to resources through RACF REVERIFY capability

The SAF/RACF support for the resources associated in the /SET, /LOCK, and /UNLOCK commands was introduced in IMS Version 9 along with the LOCKSEC parameter and additional resource classes for PSBS (IIMS/JIMS) and LTERMS (LIMS/MIMS). IMS Version 10 removes the SMU check and supports the SAF/RACF security check, which includes two checks: both a validation that the user ID of the signed on user is authorized to invoke the /SET, /LOCK, or /UNLOCK command; and a second check that the user ID is authorized against the resource being accessed: transactions, LTERMS, programs, and databases.

Password security is different when using SAF/RACF versus the previous capability with SMU. SAF/RACF uses the signed on user's password in a reverify capability whereas SMU used a global password as defined in the SMU tables.

Note: The use of a password after the parameter defining the resource will continue to be supported for all keywords except PTERM and NODE. This restriction has not changed.

The SAF/RACF check is accomplished using the RACF REVERIFY support. The REVERIFY support assumes that the RACF profile for the IMS resource is defined with the parameter APPLDATA('REVERIFY'), then IMS (assuming RVFY=Y is specified as an IMS startup parameter) checks that the password is the same as the user's sign-on password. If the resource is defined to RACF but is not authorized for use, the command is rejected with the following message:

```
DFS3689W USE OF <TRANSACTION|LTERM|DB|PROG> resourcename BY <LOCK|UNLOCK>
REJECTED.
```

All SMU-related keywords are ignored when any of the restart commands (/NRE and /ERE) are issued. The ignored SMU keywords are:

- ▶ TERMINAL, NOTERMINAL, PASSWORD, NOPASSWORD, TRANCMDs, and NOTRANCMDs

For Online Change requests, the PASSWORD, TERMINAL, and TRANCMDs keywords do not apply to the IMS Version 10 system for either a /MODIFY or INIT OLC command. For the /MODIFY command, these keywords are ignored. For INIT OLC, the keywords are still processed, and the appropriate flags in the Modify Work Area (MWA) are set, so that in a sysplex environment that consists of a mixture of IMS Version 8, IMS Version 9, and IMS Version 10 systems, the keywords can be passed on to the IMS systems at the IMS Version 8 or IMS Version 9 level.

2.1.3 Documentation changes

The removal of the SMU support has also affected the documentation for certain messages and abend codes, as well as specific application program status and return/reason codes. The documentation in the appropriate publications has been changed to remove references to SMU-based security:

- ▶ Messages:

DFS062, DFS066, DFS067, DFS095, DFS109, DFS123, DFS125, DFS126, DFS162, DFS171, DFS287, DFS288, DFS1913, DFS1919, DSF2181, DFS2468, DFS2556, DFS3436, DFS3437, DFS3440, and DFS3458

- ▶ Abend codes:

ABENDU0171, ABENDU0901, ABENDU0902, ABENDU0903, and ABENDU1050

- ▶ MNOTEs:

AGT001, AGT002, AGT003, AGT004, and AGT005

IMS documentation has also changed to remove references to SMU security:

- ▶ Messages:

DFS158, DFS1990I, DFS1991I, DFS2854A, DFS3430I, DFS3480I, DFS3654, and DFS3690

- ▶ Abend codes:

ABENDU0107, ABENDU0437, and ABENDU0718

- ▶ MNOTEs:

G972

- ▶ Application program - status codes and return/reason codes:
A4 and 0108/0308

For more information about migrating off SMU security, see *IMS Version 9 Release Planning Guide*, GC17-7831.

2.1.4 Replacing SMU security

Both IMS Version 8 and IMS Version 9 provide migration paths to the use of SAF/RACF security. IMS Version 9 provides migration capabilities that are not available with IMS Version 8.

- ▶ Enhancements include the areas of AGN, AOI, Time-Controlled Operations (TCOs), MSC link receive, sign-on verification, and /LOCK and /UNLOCK:
 - IMS Version 9 is a transition release where SMU is still available.
- ▶ Migration from SMU to SAF and from IMS Version 8 to IMS Version 10 is a valid option:
 - Begin SMU to SAF migration in IMS Version 8.
 - Migration using IMS Version 9 enhancements will have to occur during the IMS Version 10 migration.

There are small programming enhancement (SPE) APARs available that provide utilities for SMU syntax conversion for IMS Version 9 and IMS Version 10. APAR numbers and the corresponding PTFs for IMS Version 9 are PK35433/UK21894 and PK38522/UK28607. For IMS Version 10, the APAR number is PK49538.

For more information about migration from former IMS versions, see Chapter 6 in *IMS Version 9 Implementation Guide*, SG24-6398; Chapter 4 in *IMS Version 9 Administration Guide: System*, SC18-7807; and Chapters 3 and 4 in *IMS Version 9 Release Planning Guide*, GC17-7831.

Refer to “Support of RACF implementation and SMU removal” on page 278 for information about the migration steps required to replace SMU security.

2.1.5 SECURITY macro

Use the SECURITY macro statement to specify that optional security features are in effect during IMS execution unless they are overridden during system initialization. You can also use the IMSGEN and COMM macros to define security options; however, the SECURITY macro definitions take precedence.

Note: If you do not specify any security in any of the three system definition macros, IMS provides a basic level of resource security called *default terminal security*. Default terminal security prohibits the entry of certain commands from any terminal other than the master terminal. This basic security function is activated upon completion of stage 2 of IMS system definition. When you implement input-access security with RACF, IMS removes the default terminal security restrictions.

The IMS system can be defined to use the RACF licensed program (or equivalent), an exit routine, or both to perform the following types of security protection:

- ▶ Transaction authorization
- ▶ Command authorization
- ▶ Sign-on verification

- ▶ PSB authorization verification
- ▶ Application resource, or DBCTL, access authority checking

Authority to issue specific IMS commands can be restricted to certain transactions, and password and terminal security requirements can be imposed. The security options that you specify must be a part of the overall security design.

For more information about the security macro, see *IMS Version 10: System Definition Reference*, GC18-9966.

2.1.6 RACF and IMS Version 10

RACF protects IMS databases from unauthorized users. RACF is called at open time for all databases in an online system. In a DB/DC system, RACF is bypassed by VSAM for all its databases. However, RACF is invoked to verify that the control region is authorized to access any OSAM data set known to it and that is being opened. OSAM does not provide a way to bypass RACF.

In an IMS batch region, RACF is invoked when VSAM or OSAM databases known to RACF are accessed. RACF verifies that the application accessing the database is authorized.

Note: If the full function databases are protected by RACF, the DLISAS procedure must be authorized to access these resources.

RACF can provide sign-on verification security by requiring user identification at sign-on. User accountability is possible by logging the user ID in database change records and by producing a log record during sign-on and sign-off at the terminal. User exit routines are available for this type of security verification with or without the use of RACF.

For additional information about RACF and IMS, see *z/OS V1R9 Security Server RACF Security Administrator's Guide*, SA22-7683.

2.1.7 IMSplex and earlier IMS releases

If an IMSplex contains IMS Version 9 and earlier IMS releases, you need to specify the NORSCC(MODBLKS) parameter in the *DFSCGxxx* member of the IMS.PROCLIB data set to disable the resource consistency checking for the MATRIX library. If your IMS Version 9 system is running without the use of SMU security, and the systems with earlier IMS releases are using SMU security in some way, the resource consistency check fails for the MATRIX library that is used by the earlier systems.

2.1.8 Security enhancements in IMS Version 10

IMS Version 10 provides several enhancements in the security area:

- ▶ Changes to the /SIGN ON command

When a user tries to sign on to an active conversation that involves a static terminal or a dynamic terminal that has the user name the same as the node name, the sign-on attempt will fail unless all conversations are held (by using the /HOLD command for the static terminal) or the user is authorized to use the transaction for the active conversation.

- Support for mixed-case passwords

IMS Version 10 supports mixed-case passwords. A new startup parameter (PSWDC=M | U) is added to the DBC, DCC, and IMS procedures to support mixed-case passwords:

- M means mixed-case
- U (default) converts to upper-case and is compatible with previous releases

For reference purposes, the applicable RACF commands include:

- SETROPTS PASSWORD(MIXEDCASE | NOMIXEDCASE). MIXEDCASE indicates that all applications on this system and those that share the RACF database support mixed-case passwords. The syntax rules must also be modified to allow mixed-case characters. When this option is activated, the RACF ALTUSER, ADDUSER, PASSWORD, and RACLINK commands will no longer translate passwords to uppercase, nor will applications that request mixed-case password support such as IMS systems that specify PSWDC=M. This option is inactive by default.
- SETROPTS PASSWORD(RULE...) defines up to eight syntax rules for new passwords and will be used to verify that the new password meets the criteria. The rules specifying mixed-case characters should only be set when the MIXEDCASE option is in effect. When the password rules are changed, there will be no immediate impact on the users. The only time that the change will have an effect is when the users change their passwords. The rules are only used to verify that a new password meets the criteria. Any existing passwords are just verified against what is entered and what is in the database. The users will not be asked to change their password just because the syntax rules have changed. If multiple rules are defined, a password that passes at least one rule is accepted.
- SETROPTS LIST can be used to display the current setting.
- Users who were used to entering their passwords in any mode – lower, upper, or mixed – might have their sign-on rejected if a specific case is expected.
- The /LOCK, /OPNDST, /SET, /SIGN, and /UNLOCK commands are enhanced to support mixed-case passwords.

For /SIGN ON and /OPNDST, this support affects:

- Re-verification of a new password
- Password in parentheses after transaction or command (excluding MFS OPT=3)
- MFS password field in message input descriptor (MID)
- MFS password field in device input format (DIF)
- Password in parentheses after an IMS command parameter

2.1.9 Security auditing improvements

In previous releases, IMS authorization requests were inconsistent with respect to the auditing capability, that is, SMF logging and writing the ICH408I message to the system console. Some SAF requests used the normal authorization request call (AUTH) that could determine whether the security product required auditing and requested the function as part of the call. Other requests used the fast authorization call (FASTAUTH) and, when auditing was needed, performed a second AUTH call. And yet other requests issued the FASTAUTH with the additional specification of LOG=ASIS to perform auditing, when required, as part of the fast authorization.

IMS Version 10 provides a consistent method of requesting auditing. IMS authorization requests have been changed, where needed, to use the FASTAUTH call with LOG=ASIS specification to honor SMF logging requests as part of the same SAF call, thereby eliminating the optional second AUTH call that was previously required just to request SMF logging.

The RACF implementation of auditing and the way that the installation requests this capability from RACF have not changed. RACF continues to perform auditing if the authorization check results in success (RC=0) or failure (RC=8) and determines that auditing is necessary based on the following conditions:

- ▶ The user's UAUDIT setting
- ▶ The AUDIT, GLOBALAUDIT, and WARNING options in effect for the resource
- ▶ The SETROPTS SECLABELAUDIT is in effect
- ▶ The AUDIT options in the resource SECLABEL profile
- ▶ The pre-processing or post-processing installation exit's indication of whether or not to do auditing.

A new DFSDCxxx startup parameter provides a system level specification for how IMS is to handle auditing for application programs that issue the AUTH calls:

- ▶ AUTHLOG = ALL | NONE | NOMSG:
 - ALL – SMF log record written and ICH408I message to console (default)
 - NONE – No SMF log record and ICH408I message
 - NOMSG – Only SMF log record written

Another issue in prior releases was that IMS would not perform auditing if the IMS user exit DFSCCTSE0 (Security Re-verification Exit Routine) existed in the system. The documentation indicated that the user exit had to perform the auditing function.

In IMS Version 10, IMS provides new flags that allow the exit routine to request that IMS issue an authorization call in a form that will allow auditing to take place. To take advantage of this capability, the DFSCCTSE0 exit routine will need to check a new flag (CTSEAUDR in CTSEFLG1) that indicates that the auditing function is needed, and when needed, set another new flag (CTSEDAUD in CTSEFLG2) to tell IMS to issue the authorization call and request auditing.

New flags for DFSCCTSE0 to facilitate auditing:

- ▶ CTSEAUDR in flag CTSEFLG1
Tells the exit that the auditing function is needed
- ▶ CTSEDAUD in flag CTSEFLG2
Allows the exit to tell IMS to perform auditing

Note: The exit will only be able to set CTSEDAUD if CTSEAUDR is on.

Security auditing in IMS Version 10 is based on:

- ▶ The installation's specifications that request auditing of resources
- ▶ The presence or absence of the IMS user exit DFSCCTSE0. If it exists, the setting of the CTSEDAUD flag
- ▶ The specification of the AUTHLOG parameter requesting that a form of auditing be done

Note: DFSCCTSE0 from prior releases will continue to work in IMS Version 10 with no modification required. Leveraging the new capability requires modification.

2.1.10 Enhanced security for IMS conversations

IMS Version 10 enhances security for IMS conversational processing in the situation where a user attempts to sign on from a terminal that is associated with an active conversation.

In prior releases, as long as a user signing on from the terminal was authorized to one of the conversations (the active one or any of the ones on hold), that user could gain access to the active conversation even if they were not authorized to the active conversation. This situation affected all static terminals, as well as Extended Terminal Option (ETO) terminals where the control block structure of the USER was not unique to a signing on user ID. For ETO, the environment in question was one where all user IDs logging on from the terminal accessed the same control blocks, such as LName=USERname=LTERMname.

The IMS Version 10 change ensures that a sign-on will be successful only if either:

- ▶ The user ID attempting the sign-on is authorized to use the active conversation.
- ▶ No active conversations exist; that is, all the conversations are held.

The impact to static and ETO terminals when an active conversation exists includes:

- ▶ **Static**

A user, who is not authorized to the active conversation, will fail sign-on. The user can issue /HOLD on the conversation and then reissue the /SIGN ON command. /RELEASE of the conversation to which the user is authorized can be entered to resume the appropriate interaction.

- ▶ **ETO**

Users, who attempt a sign-on in environments that have generic naming conventions (LName=USERname=LTERMname), must be authorized to the active conversation if one exists, otherwise the sign-on will fail. /HOLD is not allowed prior to a /SIGN ON.

A *deferred conversational program switch* is one that responds to the terminal but causes the next input from the terminal to go to another conversational program. An *immediate program switch* passes the conversation directly to another conversational program.

When an application program inserts a deferred conversation program switch, IMS:

- ▶ Validates that the user has authorization to access the new transaction for the subsequent iteration of the conversation when the next input is received:
 - Enhances the security environment for MSC and Shared Queues
- ▶ Addresses the concern that the program issuing the deferred conversation program switch could run on a back-end MSC or Shared Queues system where the authorizations might differ from the system where the switched-to transaction might actually run.

IMS Version 10 ensures that the user is authorized to access the switched-to transaction by adding an authorization check on the subsequent input from the user. This check is made to validate that the user is authorized to access the switched-to transaction. If the terminal is disconnected during the conversation, and the conversation remains active, the authorization check is made when the next sign-on is attempted.

For more information about IMS security enhancements, see the *IMS Version 10: System Administration Guide*, SC18-9718.

2.1.11 Security and DBRC

Prior to IMS Version 10, DBRC required that all users have System Authorization Facility (SAF) CONTROL authority when the RECON data sets are under the control of a security product. As of IMS Version 10, DBRC is enhanced to support READ authority for accessing the RECON data set.

For more information about IMS security and DBRC, see Chapter 7, “DBRC enhancements” on page 157.

2.2 Analysis utilities

The Log Transaction Analysis and Statistical Analysis utilities have been rewritten for IMS Version 10. This was done for several reasons:

- ▶ The new utilities have better performance characteristics.
- ▶ Their execution JCL is simplified.
- ▶ Several restrictions have been lifted.

You can also run these utilities by using Knowledge-Based Log Analysis (KBLA) by selecting option 1 (IMS Log Utilities) from the KBLA tasks panel.

2.2.1 DFSILTA0

In an IMS DB/DC or DCCTL environment, you can use the Log Transaction Analysis utility (DFSILTA0) to collect information about individual transactions, based on records in the system log. Many events are tabulated in the Log Analysis report that is produced by this utility, including total response time, time on the input queue, processing time, and time on the output queue.

The Log Transaction Analysis utility (DFSILTA0) has several improvements:

- ▶ In previous releases, the utility could produce reports for systems connected by Multiple Systems Coupling (MSC). This required that the user invoke the Log Merge utility to merge the logs from multiple systems and use its output as input to the Log Transaction Analysis utility. The requirement to run the Log Merge utility has been eliminated.
- ▶ Logs from multiple IMS systems can be input to a single execution of the utility. One execution of the utility can process logs from multiple systems using Shared Queues. In previous releases, the Log Transaction Analysis utility did not support Shared Queues. This restriction has been eliminated.

When logs from multiple systems are input to the utility, multiple LOGINxxx DD statements are used. Any valid JCL characters can be used to replace “xxx” in the DD names. The last character in “xxx” is used as the System ID in reports. When multiple log data sets from one IMS system are inputs, they are concatenated under a single DD name.

The Log Transaction Analysis utility has an option to write the output report to an output log data set, to a printer DD, or to both. In previous releases, this was controlled by the OUT= parameter on the EXEC statement. Both outputs were produced unless the OUT= parameter eliminated them. Possible values were OUT=NOREPORT, OUT=NOLOG, and OUT=(NOREPORT,NOLOG). IMS Version 10 has eliminated this parameter. If you do not want either output, just do not include its DD statement. PRINTER is used for the report and LOGOUT is used for the output log data set.

Note: The need for the output log data set has been reduced in IMS Version 10. In previous releases, it was used to create a log for input to the Statistical Analysis utility. This is available in IMS Version 10 but not required.

In Example 2-1, we show a sample JCL for DFSILTA0. In this sample, ST=(083000,,60) has been specified. The utility begins with the first checkpoint after 083000 on the input logs and reports for 60 minutes. The input logs come from four IMS systems. The second IMS system has two log input data sets. They are concatenated. The concatenation has to be in the order in which the logs were written. A separate Log Merge utility execution is not required. The report is written to the HEADING and PRINTER DD statements. As the name implies, the HEADING DD contains only headings. The PRINTER DD contains the data for the report.

Note: A TITLE DD statement can be included as in previous releases to provide input for a title line on reports.

Example 2-1 Sample Log Transaction Analysis Utility (DFSILTA0) JCL

```
//STEP0    EXEC PGM=DFSILTA0,PARM='ST=(083000,,60)'  
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR  
//HEADING DD SYSOUT=A  
//PRINTER DD SYSOUT=A  
//LOGINF1 DD DSN=IMSF1.LOG,DISP=OLD  
//LOGINF2 DD DSN=IMSF2.LOG1,DISP=OLD  
//        DD DSN=IMSF2.LOG2,DISP=OLD  
//LOGINBA DD DSN=IMSB1.LOG,DISP=OLD  
//LOGINBB DD DSN=IMSB2.LOG,DISP=OLD
```

Note: In previous releases, the utility did not report non-recoverable transactions, including APPC messages. Support for reporting non-recoverable transactions has been added in IMS Version 10.

For more information, see *IMS Version 10: Operations and Automation Guide*, SC18-9716, *IMS Version 10: System Administration Guide*, SC18-9718, and *IMS Version 10: System Utilities Reference*, SC18-9968.

2.2.2 DFSISTS0

In an IMS DB/DC or DCCTL environment, you can produce several summary reports using the IMS Statistical Analysis utility (DFSISTS0). You can use these reports to obtain actual transaction loads and response times for the system. The statistics produced are dependent on the input system log data sets.

With IMS Version 10, DFSISTS0 has several improvements. In previous releases, the Statistical Analysis utility reports were created by a six step job, which included two sorts. The reports are now produced by a single step. This step invokes internal sorts.

Support for Shared Queues and MSC is comparable to that in the Log Transaction Analysis utility for IMS Version 10 (refer to 2.2.1, “DFSILTA0” on page 22). This includes the support for multiple LOGINxxx DD statements. When logs from multiple systems are input to the utility, multiple LOGINxxx DD statements are used. Any valid JCL characters can be used to replace “xxx” in the DD names. When multiple log data sets from one IMS system are inputs, they are concatenated under a single DD name. The capability to specify multiple input logs eliminates

the requirement to use an output log from the Log Transaction Analysis utility when you want a report from multiple IMS systems. In previous releases, this was the way that log records from multiple IMS systems were given to the Statistical Analysis utility.

Note: In previous releases, the utility did not report non-recoverable transactions, including APPC messages. Support for reporting non-recoverable transactions has been added in IMS Version 10.

The message select output order statement is a new utility control statement specified in the SYSIN DD data set. The format of the message select output order statement is:

```
ORDER=TIME|SOURCE|TRAN
```

This statement determines the order in which the message select function lists or copies messages:

- ▶ If TIME is specified, the messages are produced in input time order.
- ▶ If SOURCE is specified, the messages are ordered by the originating LTERM names.
- ▶ If TRAN is specified:
 - TIME - ordered by original input sequence
 - SOURCE - ordered by originating LTERM
 - TRAN - ordered by initial transaction (message switches will use destination LTERM)

If SOURCE is specified, the messages are produced in order of the message source, either an LTERM or a generating transaction. If DEST is specified, the messages are produced in order of the message destination, again either LTERM or target transaction. If ORIGIN is specified, the messages are produced in the order of the LTERM from which the original message was received. This differs from SOURCE only where the message is the result of one or more program switches. In this case, the origin is the originating LTERM while the source is the preceding transaction. If no value is specified, ORIGIN is used.

Example 2-2 shows a sample JCL for DFSISTS0. This sample executes the Statistical Analysis utility in IMS Version 10. You see the same set of inputs that were shown in the example for the Log Transaction Analysis utility earlier (refer to Example 2-1 on page 23).

Example 2-2 Sample Statistical Analysis Utility (DFSISTS0) JCL

```
//STEP0    EXEC PGM=DFSISTS0
//STEPLIB  DD DSNAME=IMS.SDFSRESL,DISP=SHR
//SORTLIB  DD DSNAME=SYS1.SORTLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//PRINTDCB DD SYSOUT=A
//IMSLOGP  DD SYSOUT=A
//LOGINF1  DD DSNAME=IMSF1.LOG,DISP=OLD
//LOGINF2  DD DSNAME=IMSF2.LOG1,DISP=OLD
//          DD DSNAME=IMSF2.LOG2,DISP=OLD
//LOGINB1  DD DSNAME=IMSB1.LOG,DISP=OLD
//LOGINB2  DD DSNAME=IMSB2.LOG,DISP=OLD
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(5),,CONTIG)
//SORTOUT  DD DUMMY,DCB=*.LOGINF1
//SYSIN    DD *
TRANS CODE=(ALL,I,0)
NON PRINT=HEX
/*
```

For more information, see *IMS Version 10: Operations and Automation Guide*, SC18-9716, *IMS Version 10: System Administration Guide*, SC18-9718, and *IMS Version 10: System Utilities Reference*, SC18-9968.

2.3 BPE external trace

The Base Primitive Environment (BPE) is a critical component of several IMS-related components, such as IMS Connect, the Common Queue Server (CQS), and the other address spaces introduced with the Common Service Layer (CSL). In prior releases, problem determination using the BPE traces often required an increase in the number of trace pages and a corresponding requirement to recreate the problem and capture an MVS™ dump at the correct moment. This process increased the chance of multiple repeats, thereby decreasing productivity and increasing the time to find the problem resolution.

IMS Version 10 enhances the BPE Trace facilities to provide a means to write the trace tables to external storage, tape, or DASD, as pages of trace entries are filled, thereby increasing the capacity associated with tracing. Additionally, commands are provided to display the trace table attributes and update the trace level settings. To view data, trace formatting is provided through IPCS exits and formatting routines. The same interface used to format BPE internal traces can be used to format external trace data sets:

- ▶ BPE Trace facilities enhancements:
 - Ability to write trace tables to external storage as pages of trace entries are filled
 - Commands to DISPLAY and UPDATE the trace table attributes and settings
 - IPCS support to format and print the trace entries

The existing TRCLEV statement is associated with a particular trace table and has been enhanced with the EXTERNAL=YES | NO parameter that specifies whether that trace table is to be externalized or not. The EXTTRACE statement is for the data set definition options and is needed for the function to be activated. If a user specifies EXTERNAL=YES on a table and does not specify an EXTTRACE statement, an error message is produced. The BPE external trace data sets are defined as Generation Data Groups (GDGs) and must be specified in the BPE configuration member in PROCLIB:

- ▶ External data sets use GDGs:
 - GDG name and other trace parameters are specified in the BPE configuration PROCLIB member (for example, in BPECFGxx)
 - New parameter EXTTRACE
 - Existing TRCLEV parameter
 - New EXTERNAL sub-parameter

Note: As in previous releases, each component can create a separate BPE configuration member or share one.

- ▶ Creating GDG definitions requires:
 - Defining GDG base
 - Defining GDG model

The EXTTRACE keyword definition is an enhancement to the BPECFGxx configuration definition set. One EXTTRACE keyword can be specified for each IMS component running with BPE. Alternatively, one EXTTRACE definition can be shared among multiple IMS components.

2.3.1 BPE configuration

The new EXTERNAL sub-parameter in the TRCLEV parameter included in BPE configuration member (BPECFGxx) provides the specification of whether or not the trace entries are to be externalized. Example 2-3 provides sample TRCLEV statements with the new EXTERNAL keyword.

Example 2-3 Sample TRCLEV statements in BPE configuration member

```
# DEFINITIONS FOR SYSTEM TRACES

# CQS:
TRCLEV=(CQS,MEDIUM,CQS,EXTERNAL=YES)
TRCLEV=(ERR,HIGH,CQS,EXTERNAL=NO)
TRCLEV=(INTF,LOW,CQS,EXTERNAL=YES)
TRCLEV=(STR,HIGH,CQS,EXTERNAL=YES)

# HWS:
TRCLEV=(ENVT,MEDIUM,HWS,EXTERNAL=YES)
TRCLEV=(HWSO,HIGH,HWS,EXTERNAL=NO)
TRCLEV=(OMDR,LOW,HWS,EXTERNAL=YES)
TRCLEV=(OTMA,HIGH,HWS,EXTERNAL=YES)
```

2.3.2 BPE commands

BPE commands are invoked through the z/OS MODIFY command:

- ▶ TRACETABLE resource type - internal BPE trace tables managed by:
 - BPE, for instance, DISP, CBS, STG, and LATC
 - IMS component using BPE, for example, IMS Connect, CQS, OM, RM, SCI...

Two command verbs operate on the TRACETABLE resource type:

- ▶ DISPLAY - shows the trace level and number of trace table pages of specified trace tables
- ▶ UPDATE - modifies the trace level attribute of specified trace tables

The BPE DISPLAY TRACETABLE command is an existing BPE command that is used to display the current attribute settings for the requested trace tables. IMS Version 10 modifies the command output by adding a new column called EXT to indicate whether the external trace is set on (YES) or off (NO) for a particular trace table. Example 2-4 shows an example of this command and its output.

Example 2-4 An example of DISPLAY TRACETABLE command

```
F IM1BCQS,DIS TRTAB NAME(*) OWNER(CQS)
BPE0030I TABLE  OWNER  LEVEL  ÅPAGES  EXT  CQ1BCQS
BPE0000I CQS      CQS      ERROR    4  NO  CQ1BCQS
BPE0000I ERR      CQS      HIGH     4  NO  CQ1BCQS
BPE0000I INTF     CQS      ERROR    8  NO  CQ1BCQS
BPE0000I STR      CQS      ERROR    8  NO  CQ1BCQS
BPE0032I DIS TRTAB COMMAND COMPLETED CQ1BCQS
```

The BPE UPDATE TRACETABLE command is used to change the trace level setting for the requested trace tables. The command syntax for UPDATE TRACETABLE has been extended to allow specification of the option EXTERNAL parameter from YES to NO or vice versa:

```
F jobname,UPDATE TRACETABLE .... EXTERNAL(YES | NO)
```

EXTERNAL is an optional parameter that specifies whether the trace entries for the trace tables affected by the command should be written to the external trace data set. If the keyword is not specified on an UPD TRTAB command, there is no change made to the external trace setting for the affected tables.

Note: Specification of EXTERNAL(YES) requires that the BPE configuration PROCLIB member contain the parameters to define a trace data set.

The command syntax for UPDATE TRACETABLE also has been extended to allow specification of the option OPTION(REREAD). REREAD is the only valid value for OPTION. REREAD causes the PROCLIB member to be reread for the information in the specified EXTTRACE statement. This allows the external trace data set specification to be changed or added without terminating the address space and restarting it. The syntax is as follows:

```
F jobname,UPDATE TRACETABLE .... OPTION(REREAD)
```

2.3.3 Performance impacts

Performance is always a consideration, especially when tracing is activated. The BPE external trace function is designed to minimize the performance overhead incurred at trace points in the mainline code. Running with external tracing enabled results at an approximate cost of 15 extra instructions per trace point compared to running with incore tracing.

The majority of the processing required to externalize BPE trace data occurs under a new task control block (TCB) in the BPE address space. This TCB is separate from any existing BPE or IMS component TCB. Work under existing TCBs, therefore, is not directly impacted by work performed by the new external trace TCB. The processing under the external trace TCB is more likely to be I/O bound than CPU bound.

If the writing of the external trace records cannot keep up with the rate of the generation of the records, external trace data will be lost ("skipped") rather than delaying mainline processing. The goal is to make running with BPE external trace enabled minimally disruptive to system performance.

2.3.4 New messages

Several new message are added for BPE processing:

```
BPE0043E ERROR DURING EXTERNAL TRACE PROCESSING IN MODULE module
BPE0044I BPE EXTERNAL TRACE FUNCTION NOT ACTIVE
BPE0046I EXTERNAL TRACE DATA SET dsn OPENED ON VOL vol
BPE0046I EXTERNAL TRACE DATA SET dsn CLOSED ON VOL vol
BPE0047E ERROR DURING UPDATE TRACETABLE REREAD PROCESSING
```

If an error occurs during UPDATE TRACETABLE processing, the text of the message produces one of these messages:

```
UNABLE TO GET NECESSARY STORAGE
ERROR READING BPE CONFIGURATION PROCLIB MEMBER member address space
```

```

ERROR PARSING BPE CONFIGURATION PROCLIB MEMBER member processing
UNKNOWN ERROR rc FROM PARSING MODULE BPEPCF10
LOAD FAILED FOR BPEPCF10, BPELOADC RC=rc
BPE CONFIGURATION PROCLIB MEMBER NAME WAS NOT CODED ON STARTUP PARMS
PREVIOUS UPDATE COMMAND REREAD PROCESSING IN PROGRESS
NO MATCHING EXTTRACE STATEMENT FOUND FOR ADDRESS SPACE

```

Note: If the BPE external trace function is enabled, an installation's operational procedure documentation should be updated to reflect the possibility of several new error messages associated with external tracing.

For more information about the changes for the BPE external trace, refer to *IMS Version 10: System Definition Reference*, GC18-9966.

2.4 Syntax Checker

The Syntax Checker is an IMS capability that can define, verify, and validate the parameters and their values in the members of IMS.PROCLIB. The Syntax Checker assists with IMS release-to-release migrations by providing the ability to convert supported IMS.PROCLIB members from one release to the other. The Syntax Checker, which is shown in Figure 2-1 on page 29, is enhanced in the following ways:

- ▶ Support for IMS Version 10 is added and support for IMS Version 7 is dropped.

Note: The IMS Version 10 Syntax Checker supports IMS Versions 8, 9, and 10.

- ▶ New display formats
- ▶ New keyword "Sel" Options
- ▶ Expand, Contract, Insert:
 - New selections in View menu
 - New member selection panel

2.4.1 Invocation

The invocation of the Syntax Checker in IMS Version 10 is no different than in previous releases:

- ▶ Continues to be invoked as in previous releases:
 - ISPF Option 6


```
EXEC 'hlq.SDFSEEXEC(DFSSCSRT)' 'HLQ(hlq)'
```

Where *hlq* is the high-level qualifier of the IMS data sets
 - From the IMS Application Menu, a common interface to IBM-supplied IMS applications that run on TSO and ISPF, such as the IMS Syntax Checker. Using the IMS Application Menu, you can start the IMS Syntax Checker by selecting the application and pressing Enter. To provide access to the IMS Application Menu, include the IMS.SDFSEEXEC data set in the SYSPROC DD concatenation. To start the IMS Application Menu, you can either use a TSO command or an EXEC command:
 - TSO %DFSAPPL HLQ(myhlq)
 - EXEC 'IMS.SDFSEEXEC(DFSAPPL)' 'HLQ(myhlq)'

The HLQ parameter is required for the first use of either command. If you do not

specify it for subsequent uses, the command uses the most recently specified high-level qualifier.

- ▶ Coexists on the same z/OS along with previous releases of IMS:
 - Can be used to check parameters in IMS Version 10 and previous IMS releases

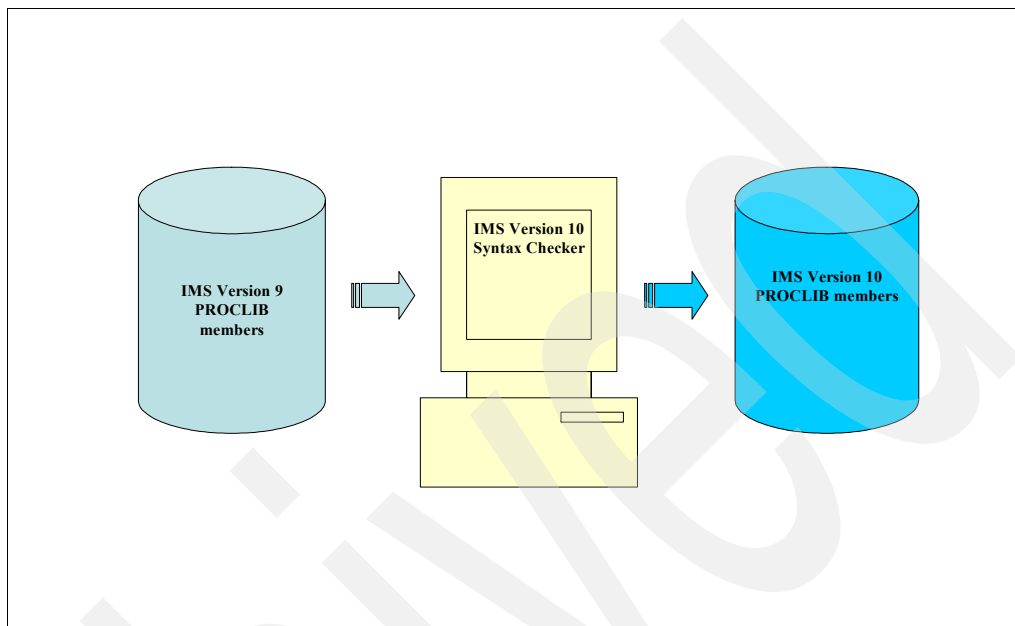


Figure 2-1 IMS Version 10 Syntax Checker and previous IMS releases

Note: When you use the Syntax Checker to check parameters for earlier releases of IMS, you must verify that the correct release number is displayed.

2.4.2 Support for PROCLIB members

In IMS Version 10, the Syntax Checker continues to support the following members:

- ▶ DFSPBxxx
- ▶ DFSDCxxx
- ▶ DFSSQxxx

To facilitate migration to IMS Version 10, the Syntax Checker's support for the currently processed members includes any new parameters in IMS Version 8 and IMS Version 9 that have been added since general availability of those releases. Support of IMS Version 7 PROCLIB members is dropped.

In addition, support for 10 other members of IMS.PROCLIB has been added:

- ▶ DFSDFxxx - Dynamic Resource Definition, Common Service Layer, Shared Queues, Restart exits, and Diagnostics and Statistics parameters (Support for IMS Version 10 only)
- ▶ DFSCGxxx - Common Service Layer Parameters
- ▶ CSLOIxxx - OM Initialization Parameters
- ▶ CSLRIxxx - RM Initialization Parameters
- ▶ CSLSIxxx - SCI Initialization Parameters
- ▶ CQSIPxxx - CQS Initialization Parameters

- ▶ CQSSLxxx - CQS Local Structure Definition PROCLIB Member
- ▶ CQSSGxxx - CQS Global Structure Definition PROCLIB Member
- ▶ IMS Connect Configuration Member

Note: This member does not have a specific member name prefix. The user can select any name. This is supported for IMS Version 9 and IMS Version 10 only. The name can be specified to the Syntax Checker during execution.

- ▶ BPE User Exit List Member

Note: This member does not have a specific member name prefix. The user can select any name. The name can be specified to the Syntax Checker during execution.

DFSDFxxx is a new member in IMS.PROCLIB with IMS Version 10. DFSDFxxx supports the parameters for Dynamic Resource Definition, Common Service Layer, Shared Queues, User Exits, and Diagnostics and Statistics. The Syntax Checker recognizes these parameters and provides validation support. The parameters are separated within the member by a special section tag:

- ▶ <SECTION=DYNAMIC_RESOURCES>
- ▶ <SECTION=SHARED_QUEUES>
- ▶ <SECTION=USER_EXITS>
- ▶ <SECTION=COMMON_SERVICE_LAYER>
- ▶ <SECTION=DIAGNOSTICS_STATISTICS>

2.4.3 Support for comments

The IMS Version 10 Syntax Checker recognizes comments in several of the members of PROCLIB. Comments can be in two forms:

- ▶ When an asterisk character (*) is encountered in column 1, the entire line is considered a comment. A member can have multiple consecutive comment lines.
- ▶ On the same line as a parameter, the text between a paired “/*” and “*/” is considered a comment.

In Example 2-5, we show samples of using comments within a PROCLIB member.

Example 2-5 Samples of using syntax checker comments

```
*-----*
```

```
* IMS COMMON SERVICE LAYER PROCLIB MEMBER *
```

```
*-----*
```

```
CMDSEC=N, /* NO CMD AUTHORIZATION CHECKING */
```

The panel for the Shared Queues member, DFSSQxxx, now supports the addition of comments. In Figure 2-2 on page 31, we show the panel for the Shared Queues member, DFSSQxxx.

File Edit View Help		

IMS 10.1 Parameters for ANY		
Command ==>		
Press enter (without other input) to check for errors.		
Data Set Name . . . : IMSV10.PROCLIB(DFSSQXXX)		
IMS Release . . . : 10.1		
Sel Codes: C = Comment D = Delete I = Insert		
P = Process + = Expand - = Contract / = Select		
Sel	Keyword	Value Description
-	CQS	= _____ CQS Start Proclib
-	CQSSN	= _____ CQS Subsystem Name
-	* EMHQ comment	_____
-	EMHQ	= _____ EMHQ same line comment
-	MSGQ	= _____ Shared Message Queues Structure Name
-	SQGROUP	= _____ XCF IMS Shared Queues Group Name Suffix
-	WAITRBLD	= _____ EMHQ Structure Rebuild Wait

Figure 2-2 The panel for the Shared Queues member, DFSSQxxx

Note: This is added only for IMS Version 10 and does not affect IMS Version 8 or IMS Version 9.

2.4.4 New DISPLAY formats

To facilitate viewing keyword values when sub-keywords and sub-keyword lists are available, two modes are provided:

- The Contract mode displays the keyword values on one line. If the values associated with the keyword extend beyond the line, they are not shown but an ellipsis “...” is displayed to give the viewer an indication that there is more to the keyword that is not displayed.

IMSPLEX (NAME = PLEX1, RSRCSTRUCTURE (STRNAME = ...

- The Expand mode uses multiple lines, and as many as are needed, to display all the keywords, sub-keywords, and sub-keyword lists that are part of the definition.

IMSPLEX (
NAME = PLEX1,
RSRCSTRUCTURE (STRNAME = RSRC1))

The formats for the keywords, sub-keyword, and sub-keyword lists are shown here:

- Not repeatable keyword
keyword_list (sub_keyword = value)
- Repeatable keyword. Some members support repeatable keywords, for example:
 - IMS Connect configuration, multiple DATASTOREs
 keyword_list (...)

```
keyword_list ( ... )
```

- ▶ Repeatable sub-keyword. New sub-keywords are automatically shown.

```
keyword_list (
    sub_keyword = value,
    sub_keyword = ____ )
```

- ▶ Positional sub-keywords

```
Keyword_list=( __, ____ )
```

- ▶ Keyword list can have a sub-keyword list in it

```
keyword_list (
    sub_keyword_list (
        sub_keyword = value ) )
```

- ▶ Keyword list can be followed by “(“ or “= (“

```
keyword_list ( ...
keyword_list = ( ...
```

- ▶ Proper indentation

```
keyword_List (
    sub_keyword = value,
    sub_keyword_list (
        sub_keyword_2 = value ) )
```

- ▶ New display format for the Save keyword:

- Used by all members in PROCLIB processed by Syntax Checker
- Always shown in expanded mode

```
*<VERSION>10.1
*<DATE>06/03/15
*<TIME>01:27
*<SYSUID>USER385
```

Note: Every member that is processed by the Syntax Checker displays the result of a SAVE using the same format in expanded mode. In previous releases, this was displayed on one line.

2.4.5 New “Sel” field options

The “Sel” field in the Parameters panel provides new options to invoke the Expand and Contract modes:

- ▶ The “Expand” “Sel” option, “+” (plus) displays the keyword on multiple lines. When a keyword is expanded, the sub-parameters are displayed in a predefined order.
- ▶ The “Contract” “Sel” option, “-” (minus sign) displays the keywords on one line. As much of the keyword as will fit on one line is displayed followed by “...”.

Note: The options are not available for all members or keywords.

Figure 2-3 on page 33 show you an example of the Parameters panel with the new Sel options. There are two displays provided. The first one is a contracted version of the IMS Message Queue structures and the second is an expanded version of the IMS EMH Queue structure definition.


```

File Edit View Help
-----
                                IMS 10.1 Parameters
Command ==>
Press enter (without other input) to check for errors.

Data Set Name . . . : IMSV10R1.PROCLIB(CQSSG001)
IMS Release . . . : 10.1
Sel Codes: C = Comment D = Delete I = Insert
           P = Process + = Expand - = Contract / = Select
Sel Keyword   Value           Description
*** DEFINITION FOR IMS MESSAGE QUEUE STRUCTURES ***
STRUCTURE( STRNAME= QMSGIMS01, OVFLWSTR= QMSGIMS010FLW, SRDSDSN1=...
*** DEFINITION FOR IMS EMH QUEUE STRUCTURES ***
STRUCTURE(
  STRNAME= QEMHIMS01,
  OVFLWSTR= QEMHIMS010FLW,
  SRDSDSN1= CQS.QEMH.IMS01.SRDS1,
  SRDSDSN2= CQS.QEMH.IMS01.SRDS2,
  LOGNAME= SYSLOG.QEMH01.LOG

```

Figure 2-3 Example of Parameters panel with the new Sel options syntax

Figure 2-4 shows you how a request to insert a keyword value can be accomplished. After the member is selected from IMS.PROCLIB, in this case CSLRI000, an Insert request (I) can be entered in front of the desired keyword.

```

File Edit View Help
-----
                                IMS 10.1 Parameters
Command ==>
Press enter (without other input) to check for errors.

Data Set Name . . . : IMSV10R1.PROCLIB(CSLRI000)
IMS Release . . . : 10.1

Sel Codes: C = Comment D = Delete I = Insert
           P = Process + = Expand - = Contract / = Select Sel Keyword   Value
Description
I  CQSSN = CQS1 ,           CQS subsystem name
-  IMSPLEX (                IMSpIex Definitions
-  NAME = PLEX1 ,           IMSpIex group name
-  RSRCSTRUCTURE (          Resource structure definitions
-  STRNAME = RSNAME         ) ) Resource structure name
-  RMNAME = RM1             RM address space name

```

Figure 2-4 Syntax of how to insert a keyword

To assist in the selection of a keyword and ensure that only valid choices are used, a pop-up menu is displayed with a selection of valid options. In Figure 2-5 on page 34, we show how you can request an insert of the ARMRST keyword.

```

File Edit View Help
-----
                          Insert keyword

Select your keyword and press Enter.

1      1. ARMRST
        2. CQSSSN
        3. IMSPLEX
        4. RMNAME

F1=Help  F3=Exit  F7=Up  F8=Down
-----

```

Figure 2-5 Syntax of selection of Insert keyword

After you have selected a valid choice on the previous panel, the Syntax Checker next displays the original panel with the new keyword in place and available to accept a value as you can see in Figure 2-6. Note that Syntax Checker inserts the new keyword in alphabetical order with respect to the other keywords.

```

File Edit View Help
-----
                          IMS 10.1 Parameters

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

Data Set Name . . . : IMSV10R1.PROCLIB(CSLRI000)
IMS Release . . . : 10.1

Sel Codes: C = Comment D = Delete I = Insert
           P = Process + = Expand - = Contract / = Select Sel Keyword

Description
- ARMRST = _ , Use ARM to restart RM after an abend
- CQSSSN = CQS1 , CQS subsystem name
- IMSPLEX ( IMSplex Definitions
  NAME = PLEX1 , IMSplex group name
- RSRCTSTRUCTURE ( Resource structure definitions
  STRNAME = RSNAME ) ) Resource structure name
- RMNAME = RM1 RM address space name

```

Figure 2-6 Selection of insert keyword with imbedded choice

2.4.6 New pull-down options

New pull-down options are displayed as you can see in Figure 2-7 when View is selected from the top of the panel. The Expand All and Contract All options are new:

- Expand All - expands all the keywords in the display
- Contract All - contracts all the keywords in the display

Note: This is not available for all members.

```
File Edit View Help
-----
Command === | *. Display All | for ANY
              | 2. Display Selected |
              | 3. Display New |
Press enter | 4. Expand All | eck for errors.
              | 5. Contract All |
Data Set Na ----- IB(CSLRIDKE)
IMS Release . . . : 10.1

Sel Codes: C = Comment D = Delete I = Insert
           P = Process + = Expand - = Contract / = Select
```

Figure 2-7 Pull-down options are new with IMS Version 10

2.4.7 New member selection panel

Two of the new PROCLIB member names that can be processed by the Syntax Checker do not have required names or prefixes. They are defined by you when requesting the function and then specified as part of the startup procedure. These members include the BPE Exit List Member and the IMS Connect Configuration Member.

When the Syntax Checker detects an unknown member name, a new Member Selection list is displayed to assist in the identification of the member.

In Figure 2-8, you see the panel that is displayed to assist the user in identifying the member type.

```
File Help
-----
                                IMS Parameter Syntax Checker

Command ==>

Enter the following information and press enter.

Select member type . . . . . _ 1. BPE Exit List Member
                                2. IMS Connect Configuration Member
```

Figure 2-8 Identifying member type

For more information, see *IMS Version 10: Release Planning Guide*, GC18-9717, and *IMS Version 10: System Definition Guide*, GC18-9998.

2.5 Enterprise Workload Management (EWLM)

For many years, z/OS has provided a workload management function through its Workload Management (WLM) component to help you manage workload distribution, balance workloads, and distribute resources to competing workloads.

Via WLM policies, you describe your workload characteristics in terms of priorities and response time goals and then WLM will distribute the system resources (CPU, I/O, and storage) accordingly so that high priority workloads will meet their goals.

z/OS provides this support automatically after you specify in the panel-driven application, which WLM provides, how you want your workloads processed.

In the past few years, IBM has provided these capabilities, which are called Enterprise Workload Management (EWLM), to other platforms as part of the IBM Virtualization Engine™ (VE) product.

Note: z/OS now supports EWLM with z/OS V1.R8 and z/OS V1.R7 (APAR OA12935).

EWLM monitors application-level transactions, operating system processes, or both.

The granularity of the performance data depends on the type of work that it monitors. EWLM provides the most granular performance data when it monitors application-level transactions, because the applications use the ARM 4.0 standard Application Program Interface (API). These APIs allow EWLM to monitor a transaction as it *hops* from one application to another to continue processing. This allows you to determine at which *hop* a performance problem might exist.

When monitoring any type of work processed in the EWLM domain, EWLM is able to provide you with an end-to-end view of the application and servers that process work. In addition, you can view how the work flows from one application or server to another.

2.5.1 Workload Manager and IMS

With IMS Version 10, IMS now provides the necessary workload management information to EWLM, so IMS as a subsystem under z/OS can now participate in a cross-platform EWLM and the Virtualization Engine environment.

To better understand this IMS Version 10 support, the next topics contain an overview of the IBM Virtualization Engine and EWLM.

2.5.2 Virtualization Engine

The IBM Virtualization Engine is a comprehensive portfolio that can help you aggregate pools of resources and get a consolidated view of them throughout your IT environment. The IBM Virtualization Engine includes technologies that are built into IBM hardware, IBM software, and IBM TotalStorage solutions. The attributes of Virtualization Engine are:

- ▶ IBM Virtualization Engine is about simplifying and optimizing end-to-end heterogeneous IT infrastructures:
 - Helps you to get more out of IT investments
 - Enables dramatic improvements in efficiency, performance, integration, utilization, and management of computing resources

- ▶ IBM Virtualization Engine allows clients to aggregate pools of diverse physical resources into a single logical entity.
- ▶ IBM Virtualization Engine provides simplified virtual resource management and increased utilization of IT assets.
- ▶ IBM Virtualization Engine is the delivery vehicle of the key IBM virtualization technologies.

In Figure 2-9, you see the complete portfolio of IBM Virtualization Engine.

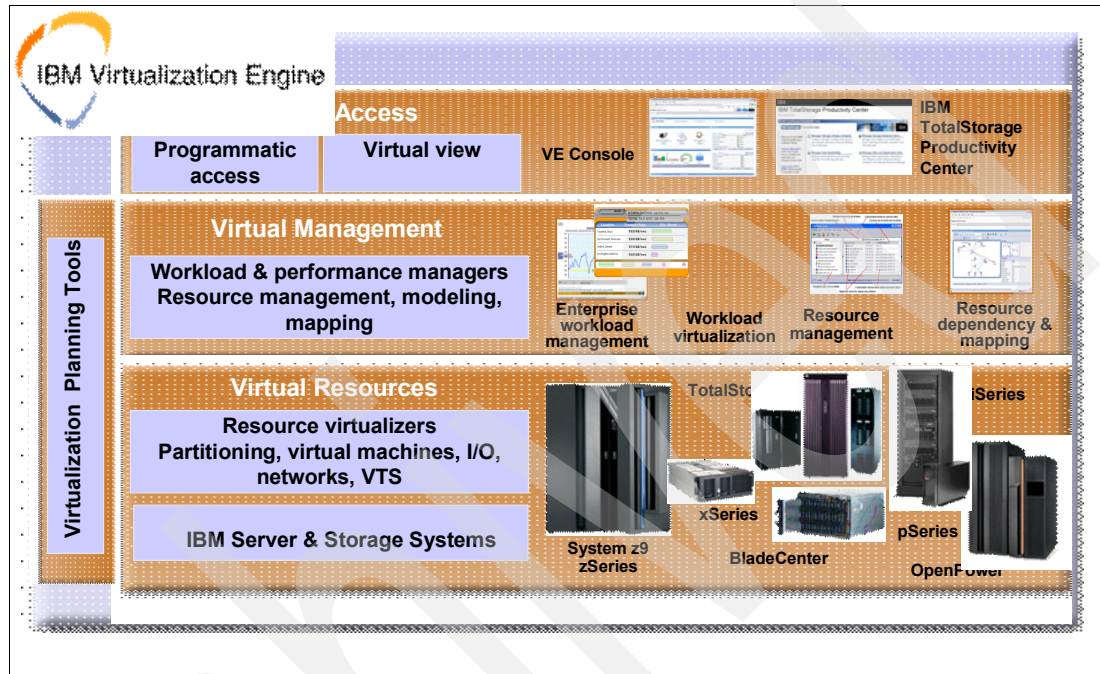


Figure 2-9 IBM Virtualization Engine – complete portfolio

Enterprise Workload Manager

EWLM is a technology that brings self-optimization in the form of performance and response time management to an entire Virtualization Engine-enabled IT environment. It allows system administrators to define business-oriented performance objectives for workloads running across different platforms and monitors and manages actual performance against those objectives.

In Figure 2-10, you see the overview of Enterprise Workload Manager.

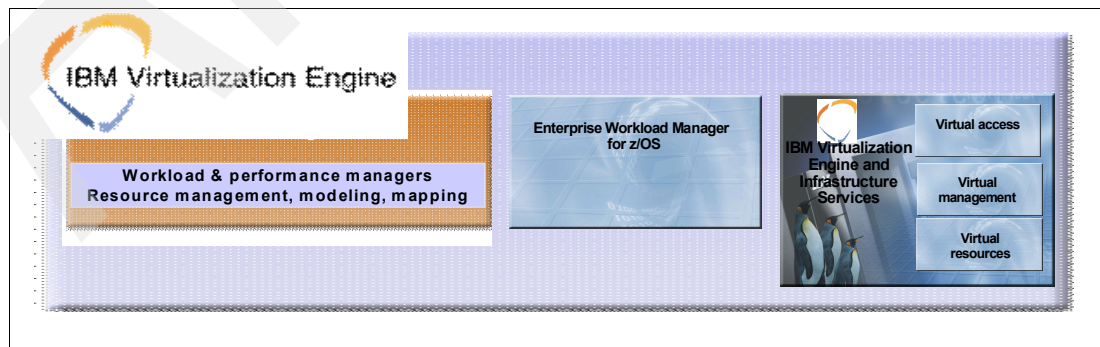


Figure 2-10 Enterprise Workload Manager – overview

EWLM and z/OS WLM integrated reports

EWLM reports on end-to-end performance in a multi-platform environment. With z/OS support and now IMS Version 10 support of EWLM, performance data from mainframe platforms that have IMS workloads will be included in these reports:

- ▶ z/OS IMS reporting into EWLM reporting:
 - z/OS WLM is enhanced so that z/OS IMS performance reporting is included in the end-to-end performance reporting of EWLM.
- ▶ EWLM reports now deliver performance data on a multi-tier application including IMS on z/OS.
- ▶ Performance problems in the z/OS application tier can be managed through the new z/OS WLM support for service class correlation.

In Figure 2-11, you see an overview of EWLM and z/OS WLM integrated reports.

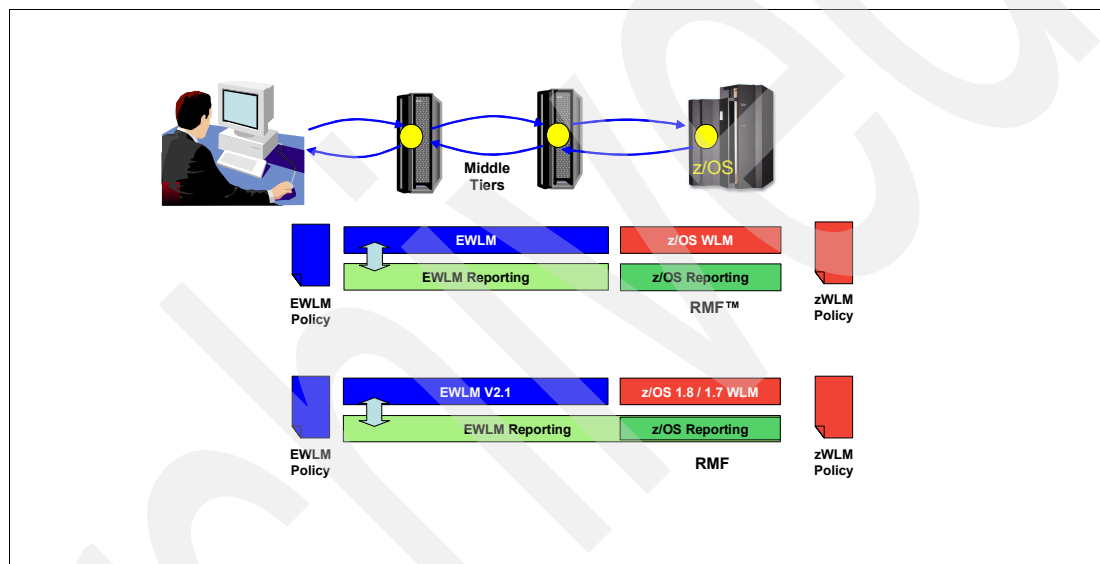


Figure 2-11 EWLM and z/OS WLM – integrated reports

2.5.3 How to implement EWLM to IMS Version 10

To run IMS with EWLM, you need to have EWLM on the platforms that are in the transaction flow. You also need the EWLM domain manager. If you are running IMS TM Resource Adapter (formerly IMS Connector for Java) on IBM WebSphere Application Server (WAS) for z, you need the EWLM agent on z/OS. Similarly, if you are running IBM WebSphere Application Server on another platform, you have to provide the agent for that platform.

Note: EWLM is an independently shipped program product; it is not shipped with z/OS. To run EWLM, order the product *IBM Virtualization Engine Enterprise Workload Manager for z/OS, V2.1.0*, 5655- EWM.

For more information about EWLM, see *z/OS V1R8.0 MVS Planning: Workload Management*, SA22-7602, and *IBM Enterprise Workload Manager*, SG24-6785.

IMS Version 10 will provide EWLM information for EWLM-enabled IMS transactions:

- ▶ Transactions include the 64-byte EWLM correlator in the *OMUSR_EWLMCORR* field in the user data section of the OTMA header.

- ▶ Transactions must come in through IMS Connect or OTMA:
 - IMS TM Resource Adapter (IMS Connector for Java) through IMS Connect and OTMA
 - MQBridge through OTMA

For more information about OTMA, see *IMS Version 10: Communications and Connections Guide*, SC18-9703. For more information about IMS and WLM, see *IMS Version 10: System Administration Guide*, SC18-9718.

2.6 IMS Abend Search and Notification

IMS Version 10 provides a new diagnostic tool for system administrators, the IMS Abend Search and Notification (ASN) capability. It automatically notifies a designated set of contacts of an IMS abnormal termination (abend) and includes direct access to guidance on how to resolve the abend. IMS ASN also enables you to create a notification on demand and have it sent to a designated set of contacts containing information about an IMS abend or failure. In this section, we discuss IMS ASN:

- ▶ Functionality and benefits
- ▶ Prerequisites
- ▶ Installation
- ▶ Customization components

2.6.1 Functionality and benefits

You can configure IMS ASN to automatically notify you when an IMS abends, or you can use it proactively to obtain guidance for an abend that has already occurred. These two methods of using ASN are referred to as *event driven ASN* and *on demand ASN*, respectively. IMS ASN does not interfere with existing IMS programs, transactions, or database-related activity and is activated only when an IMS abend occurs. Messages are sent internally from the IMS system task without including sensitive information, such as client data. We now explore each of the IMS ASN methods in further detail.

Event driven ASN

When an abend occurs, IMS sends a message that contains an abend code to an operator's console or to the master terminal operator (MTO). This process delays a response, because the operator who can respond to the abend might not physically be at the console when the message is sent. Therefore, action in response to the abend is delayed until the message reaches the correct operator.

Event driven ASN enables IMS to send an e-mail or SMS text message to a designated e-mail address, or mobile devices, respectively, when an abend occurs, in order to:

- ▶ Notify the correct operator of an abend
- ▶ Provide a Web address (URL) to the location of additional informational resources, such as IMS product documentation, technical notes in an IBM technical support database, and information in the preventive service planning (PSP) database.

You can see an sample of an e-mail message sent by IMS ASN in Figure 2-12 on page 40.



Figure 2-12 Sample e-mail sent by IMS ASN, notifying the user of an S0C4 abend

When an IMS control region abends, only the first abend will trigger an e-mail or SMS text message to be sent. IMS ASN will not group abends into one message or send multiple messages for the same abend associated with different tasks. IMS ASN targets a single IMS cycle, and after an abend, another message will be sent only if the IMS is restarted and then abends again. A new Dump Suppression Table that was added to module DFSFDMP0 in IMS Version 10 determines whether the abend requires a notification to be sent. The module issues the abend message and invokes the ASN procedure, which in turn constructs information to create the electronic message using pre-constructed text. Information that is specific to the IMS abend will be populated in this text and sent to the user.

Restriction: IMS ASN does not support abends that occur in MPP regions.

On demand ASN

You can also use IMS ASN to dynamically research abends and build links to the same Web addresses that event driven ASN uses. Again, these URLs contain links to resources, such as IMS product documentation, technical support databases, and information in the preventive service planning (PSP) database. This method is known as *on demand ASN*.

You can use on demand ASN via the TSO interface, which provides one panel for you to specify information about the abend that you are researching. The final result is the creation of an e-mail that is then sent to you or another designated group of contacts. You can access on demand ASN via the introductory IMS ASN menu (shown in Figure 2-15 on page 43) by selecting option 2 in the *Tasks* field. After selecting this option, you see a panel like the one shown in Figure 2-13.

IMS abend search and notification - on demand interface		
COMMAND ===>		
*Product Name	IMS	REQUIRED (example: IMS)
Provide one or more of the following web search arguments:		
PSP for FMID	HMK9900	(example: JMK7701)
Abend code	S0C4	(example: S0C4)
Return code (RC)	12	(example: 44)
Module name	DFSKBLAK	(example: DFSSAMP0)
APAR number (PE or other)	PQ06421	(example: PQ99999)
Message ID		(example: DFS555I)
Generic search argument	TRANSACTION	(example: SQL+OVERFLOW)
Recipient Information		
Recipient e-mail address	GREENHAW@US.IBM.COM	(example: name@company.com)
Specify additional addresses? _		(Y-Yes/N-No)
JOB JCL statement	_	(E-Edit/Y-Yes/N-No)

Figure 2-13 The IMS ASN on demand interface

2.6.2 Prerequisites

Before you enable IMS ASN, you must ensure that a few hardware and software requirements are met. The following are prerequisites for IMS ASN:

- ▶ IMS Version 10 must be installed.
- ▶ Because IMS ASN generates a graphically formatted e-mail, the recipients of the e-mail require an e-mail client that supports Multipurpose Internet Mail Extensions (MIME) Version 1.0 or higher to view the graphic format. If the e-mail client does not support MIME Version 1.0 or higher, a text version of the e-mail is displayed instead. For optimal results, the e-mail manager should support graphics and images.
- ▶ Because the e-mail is sent using Simple Mail Transfer Protocol (SMTP), the IMS system must reside on a z/OS system that has enabled TCP/IP. SMTP is a standard application that is delivered with the TCP/IP stack.
- ▶ Lotus® Notes® clients must use Lotus Notes Release 6.0 or later with patch MJUK42GR8K, in addition to MIME 1.0 support.

2.6.3 Installation

To set up and enable IMS ASN, you must first allocate two required data sets, then complete the IMS ASN system setup panels via ISPF.

Required data sets

You must allocate and authorize the following two data sets:

- ▶ A runtime data set that contains the information from the choices made during the setup process when working with the IMS ASN system setup panel. The name of this runtime data set is included in all of the JCL jobs that are generated by IMS ASN. The DCB for the runtime data set is RECFM=FB,LRECL=80.

Users of the IMS ASN system setup panel need update access to the runtime data set. All other users of the IMS ASN function need read-only access. Examples of such users are:

- Started tasks that initiate abend-event driven invocation of the function
- Users of the IMS ASN on demand panels
- The user ID associated with the IMS control region
- ▶ A skeleton data set that contains the following members:
 - DFSSPRCI, which specifies the name of the IMS abend search and notification procedure
 - DFSSPCLI, which specifies a JCLLIB statement that points to where the procedure resides
 - DFSIAPRM, which contains keywords and their corresponding values to be parsed into the e-mail and text message body

The DCB for the skeleton data set is RECFM=FB,LRECL=80. The skeleton data set must be concatenated to the ISPSLIB. Some system environment configurations might prevent the concatenation. In that case, name the skeleton data set *hlq.IASNSLIB*, where *hlq* is the high-level qualifier used to install IMS.

After it is named, the data set is automatically concatenated to ISPSLIB for all users of the IMS abend search and notification panel. Therefore, the installer of IMS abend search and notification must have RACF ALTER authorization for the *hlq* that is used to install IMS.ALTER. This allows the installer to read, write, create, or delete the data set.

Complete system setup

For the next step of the installation, you must access and fill out the IMS ASN system setup ISPF panel to populate the runtime data set and skeleton data set. The IMS ASN system setup panel can be accessed by either of the following methods:

- ▶ Select option 9 Abend Search and Notification (ASN) from the IMS Application Menu, which you can see in Figure 2-14.
- ▶ From ISPF option 6, enter the following command, where *hlq* is the high-level qualifier used to install IMS:

```
exec 'hlq.SDFSEEXEC(DFSRASNO)' 'HLQ(hlq)'
```

Help	
IMS Application Menu	
Command ==> _____	
Select an application and press Enter.	
1	Single Point of Control (SPOC)
2	Manage resources
3	Knowledge-Based Log Analysis (KBLA)
4	HALDB Partition Definition Utility (PDU)
5	Syntax Checker for IMS parameters (SC)
6	Installation Verification Program (IVP)
7	IVP Export Utility (IVPEX)
8	IPCS with IMS Dump Formatter (IPCS)
9	Abend Search and Notification (ASN)
To exit the application, press F3.	

Figure 2-14 The IMS Application Menu

You should now see the introductory IMS Abend Search and Notification panel, as shown in Figure 2-15.

```

      IMS Abend Search and Notification    IMS Version 10.1
Command ==> _____

                                           TIME....16:01:59
                                           DATE....2007/09/19
                                           USERID..DARIO

Select one of the following tasks and press ENTER .

Tasks . . - 1. IMS ASN System Setup
              2. IMS ASN On-Demand Interface

To Exit this menu, press the END key.
For Help information, place cursor on any field and press PF1 .

```

Figure 2-15 The introductory IMS ASN panel

From here, type 1 in the Tasks field to select **IMS ASN System Setup**. Next, you see a panel like the one shown in Figure 2-16.

```

      IMS abend search and notification - system setup
COMMAND ==> _____

Identify IMS ASN skeleton lib data set (must be concatenated to ISPSLIB):
*Skeleton lib DS _____

Identify the following data sets and members (must be accessible at runtime):
*Runtime DS _____
PROCLIB Mbr _____ Restore? _ URLS Mbr _____ Restore? _
SYSUT1 Mbr _____ Restore? _ CONTROL Mbr _____ Restore? _
SMS Mbr _____ Restore? _ SMSCNTL Mbr _____ Restore? _

Identify the fully qualified name of your installation's IMS.SDFSRESL:
*IMS.SDFSRESL DSN _____

E-mail and Recipient Information
*Recipient e-mail address. . . _____
Specify additional addresses? _ (Y-Yes/N-No)
SMS recipient address . . . . _____
Specify additional SMS rcpts? _ (Y-Yes/N-No)
External Writer ID for Local SMTP _____
FROM e-mail. . . . . _____
External SMTP Server _____
SMTP Port. . . . . _____

```

Figure 2-16 The IMS abend search and notification - system setup panel

Use the IMS abend search and notification - system setup panel to perform the following tasks:

1. Name and customize the IMS ASN procedure JCL (a sample named DFSIASN0 is shipped in IMS.SDFSRLIB).
2. Name and customize one or more SYSUT1 PDS members with the e-mail message text. You can either use the IBM-supplied SYSUT1 PDS member as is, or customize it.

3. Name and customize one or more CONTROL PDS members with e-mail routing information. You can either use the IBM-supplied CONTROL PDS member as is, or customize it.
4. Name and customize one or more SMS PDS members with the text for the text messages. You can either use the IBM-supplied SMS PDS member as is, or customize it.
5. Name and customize one or more SMSCNTL PDS members with the text message routing information. You can either use the IBM-supplied SMSCNTL PDS member as is, or customize it.
6. Name and customize one or more URLS PDS members with the URL information to be mapped into the e-mail and text messages. You can either use the IBM-supplied URLS PDS member as is, or customize it.

The IMS abend search and notification - system setup panel has general online help, as well as field-sensitive online help. To access the online help, place the cursor on the command line (for general help) or in an input field (for field-sensitive help) and press F1.

For detailed information about the DFSIASN0 procedure and these PDS members, see the *IMS Version 10: System Definition Guide*, GC18-9998.

Complete the following steps to finish the setup process:

1. Specify IASNPROC= *member_name* in the diagnostics section of the DFSDFxxx IMS.PROCLIB member, where *member_name* is the 8-character name of the IMS ASN procedure. The name must be eight characters. This task can be performed using the Syntax Checker or by manually inserting the output of the IMS abend search and notification - system setup panel (which resides in the DFSMSxxx member of the runtime data set) into the DFSDFxxx IMS.PROCLIB member. For information about the DFSDFxxx IMS.PROCLIB member, see the *IMS Version 10: System Definition Guide*, GC18-9998.
2. Copy the IMS ASN procedure, which was created and customized through the use of the IMS abend search and notification - system setup panel, to a concatenated z/OS procedure library.
3. Provide the name of the skeleton data set to the users of the IMS ASN on demand interface by one of the following methods:
 - a. Concatenate the data set name in the ISPSLIB DD statement in the logon procedures used by the TSO users.
 - b. Concatenate the data set name directly by statements in the default CLIST of those TSO users.
 - c. Name the skeleton data set in the format *hlq.IASNSLIB*, where *hlq* is the high-level qualifier that is used to install IMS. After it is named, the data set is automatically concatenated to ISPSLIB for all users of the IMS abend search and notification panel. If you perform setup a second time and use a data set name other than *hlq.IASNSLIB* for the skeleton data set, and *hlq.IASNSLIB* still exists, ISPF looks for the skeleton library members in *hlq.IASNSLIB* before looking in the other data set.
 - d. Provide the TSO users a program that will perform the concatenation.

To disable the IMS ASN function, set *member_name* either to blanks or a null value in the IASNPROC= parameter in the diagnostics section of the DFSDFxxx IMS.PROCLIB member.

2.6.4 Customization components

You can customize the e-mail notification that is sent and add a specific URL link, by modifying both the URLS and the SYSUT1 members. You can access and modify the members using either of the following methods:

- ▶ Run the setup panel, which displays the customizable members in sequence for editing.
- ▶ Directly access the members in ISPF in the Runtime partitioned data set and use the names specified in the setup panel.

To create a new link to be inserted into the SYSUT1 member, modify the URLS member using the following steps:

1. Access the URLS member DFSIAURL. The URLS member is the second member in sequence from the setup panel.
2. Insert the link using the following format:

```
*This is a new link
#NEWURL
http://www.new.com/
```

A sample of this is shown in Figure 2-17.

Menu	Utilities	Compilers	Help
BROWSE KINLAU.IASN.SDFSSWAN(DFSIAURL) - 01.01 Line 00000034 Col 001 080			
Command ==> _____ Scroll ==> CSR			
#FMIDURL			
http://www-1.ibm.com/support/search.wss?atrn=SWPTFNumber&atrsv=#FMID			
&dscrch=TRUE&apar=exclude&dc=DB540+DB400&dtm			
*Find description for the specified message ID			
#MSGURL			
http://publibz.boulder.ibm.com/cgi-bin/zoslib/lookat.cmd?required=msgid			
&missing_fields_redirect=%2Fservers%2Feserver%2Fzseries%2Fzos%2Fbkserve			
%2Flookat%2Flookatidmiss.html&submit=Go&release=ZOS%2FV1R7&searchIndex=			
INDEX&rank=RANK&msgid=#MSG			
*Search technical support for the specified string			
#GENURL			
http://www-1.ibm.com/support/search.wss?q=#GEN&rs=0&lang=en+en			
&loc=en-US&from=tss&ics=iso-8859-1&cs=utf-8&cc=us&search=search			
*Link to PSP, no search			
#PSPURL			
http://www-1.ibm.com/support/docview.wss?rs=0&dc=DB540&q1=HMK1010			
&uid=isg1_IMS1010_HMK1010&loc=en_US&cs=utf-8&lang=			
*Order PTF electronically			
#PTFURL			
https://techsupport.services.ibm.com/server/390.elecptfor			
*New Link			
#NEWURL			
http://www.newcompany.com/			

Figure 2-17 Creating an e-mail link in the IMS ASN panel using the URLS member

This creates a new link that can be inserted into the SYSUT1 member to become part of the IMS ASN e-mail. Note that the first line, which is preceded with an asterisk (*), is a comment line. Following the comment line is the name of the link, preceded with the number sign (#). The name can be up to eight alphanumeric characters, including the #. Following the link name is the URL link to be associated with the name. This link replaces any occurrence of the link name that exists in the SYSUT1 member.

To add the link created in the URLS member into the e-mail using the SYSUT1 member, follow these steps:

1. Access the SYSUT1 member DFSIAEML. The SYSUT1 member is the third member in sequence from the setup panel and follows the URLS member.
2. Scroll down to the text and HTML sections. These two sections are marked by "Content-Type: text/plain" and "Content-Type: text/html", respectively.
3. Add the following text in the text section:
 9. This is a new link
 - #NEWURL
4. Add the HTML syntax in the HTML section. You must follow HTML syntax rules and the name of the link #NEWURL must be on a line by itself, for example:

```
<li>
<a href="
#NEWURL
">This is a new link.
</a></li>
```

This creates a new link at the bottom of the list in each IMS ASN e-mail. You can see an example of how to insert the name into the SYSUT1 member in Figure 2-18.

Menu	Utilities	Compilers	Help
BROWSE	KINLAU.IASN.SDFSSWAN(DFSIAEML) - 01.01	Line 00000059	Col 001 080
Command ==>		Scroll ==>	CSR
4. APARs related to #MOD			
#MODURL			
%MOD			
%APAR			
5. Search results for PE APARs in current maintenance level - #APAR			
#FIXURL			
%APAR			
%MSG			
6. Search results for Message ID #MSG			
#MSGURL			
%MSG			
%FMID			
7. PSP search results for FMID #FMID			
#FMIDURL			
%FMID			
%GEN			
8. General search results for #GEN			
#GENURL			
%GEN			
9. This is a new link			
#NEWURL			

Figure 2-18 Adding an e-mail link in the IMS ASN panel using the SYSUT1 member in the text section

You can see an example of the HTML formatting for the SYSUT1 member in Figure 2-19 on page 47.

```

Menu Utilities Compilers Help
BROWSE      KINLAU.IASN.SDFSSWAN(DFSIAEML) - 01.01      Line 00000204 Col 001 080
Command ==>      Scroll ==> CSR
</li>
<a href="
#GENURL
">General search results for #GEN
</a></li>
%GEN
      </li>
<a href="
#NEWURL
">This is a new link.
</a></li>
      </ul></font>
      </td></tr></table>
      </td>
    </tr>
    <tr>
      <td width="375">
        <table border="0" cellspacing="0" cellpadding="5"><tr><td>
          <font size="-2" face="Verdana, Arial, Helvetica, sans-serif">
            <a href="
            #PSPURL
            ">Search the
            Preventative Maintenance Planning (PSP) Database</a>
            for specific installation tips, high impact or pervasive problems
            and service recommendations.</font><br>
          </td></tr></table>
        </td>
      </td>
      <td width="225" valign="center" align="center">

```

Figure 2-19 HTML formatting using the SYSUT1 member

For more information, see the *IMS Version 10: Release Planning Guide*, GC18-9717.

2.6.5 Summary

IMS ASN improves your self-sufficiency when an IMS abend or failure occurs by providing you with immediate notification through an e-mail or SMS text message. Messages contain links that take you directly to the specific IMS technical support Internet resources and databases that will assist you in troubleshooting the abend or failure. With one click, you will see the explanation of the abend or failure and know whether your IMS system failed due to a known problem. Some of the e-mail links have imbedded abend specific search keywords. This implies a significant problem determination simplification mainly for the following reasons:

- ▶ The generated e-mail will contain the pointer to the description of the specific abend experienced and, equally as important, the associated problem diagnosis procedure suggested if available (no manuals to find).
- ▶ The generated e-mail will have various search URLs with abend specific keywords imbedded.

This implies that the user does not need to find the keywords, know what keyword to use for the search, nor where on the Web this search should be performed.

2.7 Knowledge-Based Log Analysis (KBLA)

The Knowledge-Based Log Analysis (KBLA) utilities were originally introduced in IMS Version 9 and their capabilities are further enhanced in IMS Version 10.

KBLA is a collection of IMS utilities that select, format, and analyze log records. KBLA also provides an ISPF interface to create and run the jobs for various log-related utilities and to access other ISPF applications. KBLA uses an ISPF panel-driven user interface to simplify JCL job creation and to prevent JCL errors. KBLA generates the JCL and control statements necessary to run the supported utilities. This JCL preparation allows you to focus on the output of the utility used rather than on how to code JCL to extract information.

IMS Version 10 enhances this capability to support the log record changes introduced in this release. Additionally, enhancements to support multiple concurrent releases of IMS ensure that the correct log record DSECTs associated with an IMS release are used and that the parameters, which were defined for the prior release, are not invalidated or superseded by the installation of a new release.

2.7.1 Invoking KBLA

SDFSRESL contains the KBLA utility modules. You have to define the appropriate SDFSRESL library for an IMS version in the Default Parameter Maintenance panel. An audit in KBLA restricts the specifiable version number to either 9 or 10. Different SDFSRESL libraries for a version can be specified such as 9, 91, 10, 101, and so forth. KBLA attempts to find the best match of the IMS version specified on a processing panel to the SDFSRESL data set supplied on the Option 0.1 panel after KBLA is invoked. KBLA can be invoked in two ways:

- ▶ It can be invoked from the IMS Application Menu.
- ▶ It can be invoked directly.

To access KBLA from the IMS Application Menu:

- ▶ Access the IMS Application Menu by typing the following command:

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

hlq is the high-level qualifier that is associated with the installed IMS subsystem.

- ▶ From the IMS Application Menu, select option 3 for KBLA.

With IMS Version 9, if a value was not specified for the 'KBLA Loadlib' parameter in the KBLA ISPF Option 0.1 panel (IMS K.B.L.A. - Define KBLA Environment), KBLA defaulted to the specification associated with the IMS Version 9 resource library. KBLA always included this data set as the first DSN in the JOBLIB DD concatenation sequence.

IMS Version 10 renames the variable from *KBLA Loadlib*, which previously implied that a value was required, to *KBLA Test Loadlib*, which is intended to imply that a value is optional and perhaps only used for testing purposes. If specified, the associated data set will be included as the first DSN in the JOBLIB concatenation sequence.

Refer to 10.3.9, "Knowledge Based Log Analysis coexistence and migration" on page 270 for information how to migrate KBLA to IMS Version 10.

2.8 Large sequential data set support

z/OS V1.R7 Data Facilities Storage Management Subsystem (DFSMS) added support for large sequential data sets on DASD. These data sets can occupy more than 65 535 tracks on a single volume, enabling more data to be stored on fewer volumes. Without this support, physical sequential data sets (DSORG=PS) were restricted to 65 535 tracks on any volume. Multi-volume data sets could have up to 65 535 tracks on each volume. The architectural

(theoretical) limit with large data set support is 16 777 215 tracks per volume. This is far beyond the maximum number of tracks per volume that storage subsystems support.

Sequential data sets can get very large and can span many volumes of storage hardware. They are used in IMS primarily for GSAM/BSAM and OSAM data sets. These include:

- ▶ Logs (online log data set (OLDS) and system log data set (SLDS))
- ▶ Trace data sets
- ▶ Message queue data sets

IMS Version 10 adds support for large sequential data sets. IMS modules that process sequential data sets have been modified to accept large format data sets.

Migration to large format data sets can only be done at data set creation time. The following process shows one possible approach to using this enhancement:

- ▶ Determine which existing GSAM data sets that use the BSAM access method and which sequential OSAM data sets span more than one volume because they are larger than 65 535 tracks.
- ▶ Determine if you have hardware that supports more than 65 535 tracks.
- ▶ Allocate the data set with **DSNTYPE=LARGE** on the DD statement in the JCL that defines the data set or sets, TSO/E **ALLOCATE**, AMS **ALLOCATE** statement, dynamic allocation (SVC 99), or SMS data class.
- ▶ Bring the new data sets online.

If you are using the new data sets for an OLDS or a message queue, perform a cold start of IMS to bring the new data sets online. If you are using the new data sets for a database, use a database reorganization process (unload and reload) to bring the new data sets online.

No other actions are required. In Example 2-6, you see an example for the allocation of a large data set.

Example 2-6 Example JCL statement for the allocation of a large data set

```
//ABSDEF DD DSN=IMS.ABC.XYZ,DSNTYPE=LARGE,  
//          UNIT=SYSDA,SPACE=(CYL,(4500,100)),  
//          DISP=(NEW,CATLG),VOL=SER=LRGVOL1  
//*
```

In Table 2-2 on page 50, you see which volume types can contain large format data sets. The volume types 3390-1, 3390-2, and 3390-3 have fewer than 65 535 tracks per volume. Large format data sets can be defined on them; however, they cannot take advantage of the larger number of tracks per volume. The table also shows the approximate capacity of a volume with a variety of the most common block sizes.

Table 2-2 Tracks per volume for 3390s

Volume Type	Number of tracks per volume	Can exploit DSNTYPE=LARGE	Capacity with 4 K, 8 K, 12 K, and 24 K blocks
3390-1	16 695	No	0.76 G (1 GB = 1,073,741,824 bytes)
3390-2	33 390	No	1.53 G
3390-3	50 085	No	2.29 G
3390-9	150 255	Yes	6.88 G
3390-27	491 400	Yes	22.49 G
3390-54	982 800	Yes	44.99 G

Large format data sets can be used for any log, trace data set, message queue data set, GSAM/BSAM data set, or OSAM database data set. They can be used to avoid the requirement for multiple volumes for very large data sets. Because OLDS and message queue data sets must reside on only one volume, large format data sets allow OLDS and message queue data sets to be larger than previously possible.

Note: No specific hardware is required for this enhancement, but to take advantage of this support, you must use hardware that has more than 65 535 tracks.

OSAM database data sets are limited to a maximum of 8 gigabytes. The large format data set support does not change this limit. IMS Version 10 large sequential data sets cannot be shared with previous versions of IMS.

For more information, see *IMS Version 10: Release Planning Guide*, GC18-9717.

2.9 Online change enhancements

IMS Version 10 introduces two new enhancements to the online change process that improve IMS system availability. Both enhancements accomplish this by modifying the online change process so that a smaller amount of resource is quiesced. The ACBLIB member online change enhancement quiesces only the specific resources being changed. The online change enhancement for *work in progress* reduces what is considered work in progress for the resources being changed, thereby increasing the likelihood that the online change process will succeed. We now explore each of these enhancements in detail.

2.9.1 ACBLIB member online change

ACBLIB member online change (MOLC) is a new capability that allows you to add or change one or more members of your ACBLIB without the need to perform an online change on the entire library. MOLC does not support deletion of ACBLIB members. Only the resources that are affected by the MOLC are quiesced, allowing for more concurrent activity during the online change process than the full library switch type of online change. Its goal is to improve online change usability and improve resource availability. MOLC coexists with the full library switch capability. Let us take a look at how MOLC operates, beginning with some prerequisites that must be in place before using it.

Required environment and definitions

Before you can start to use the ACBLIB member online change function, you must ensure that the following conditions have been met:

- ▶ An OLCSTAT data set must be allocated and initialized. If your IMSplex is comprised of a single IMS system, you must use OLCSTAT instead of MODSTAT.
- ▶ Global online change must be enabled (OLC=GLOBAL) using either the DFSCGxxx or DFSDFxxx PROCLIB member data set.¹
- ▶ If your IMSplex is comprised of a single IMS system, a Common Service Layer (CSL) with the Operations Manager (OM) and Structured Call Interface (SCI) must be in place. In this case, a Resource Manager (RM) is not required. Therefore, specify RMENV=N in either the DFSCGxxx or DFSDFxxx PROCLIB member data set to indicate that RM is not required.
- ▶ If your IMSplex is comprised of multiple IMS systems, a CSL including an OM, SCI, and RM must be in place. A resource structure is recommended, although not required.
- ▶ All members in your IMSplex must be IMS Version 10.

In addition to these prerequisites, you must also define a dynamic allocation library for your staging ACBLIB. We now explore this step in more detail.

Dynamically allocating the staging ACBLIB

Prior to IMS Version 10, the staging ACBLIB was not used in the online change process and was not included in the IMS control region JCL. To use the MOLC capability, you must dynamically allocate the staging ACBLIB using the DFSMDA member TYPE=IMSACB. By using TYPE=IMSACB, you are only required to specify the data set name for the staging ACBLIB. A dynamic allocation member with the name of IMSACB is created for the staging ACBLIB.

Restriction: You cannot create database MDA members with a DDNAME=IMSACB.

IMS dynamically allocates the staging ACBLIB with DISP=SHR. To ensure that the staging ACBLIB can be allocated by your IMS control region, you can use either of the following methods:

- ▶ Create a DFSMDA member for the ACBLIB staging library. Here is sample JCL:

```
DFSMDA TYPE=INITIAL
DFSMDA TYPE=IMSACB,DSNAME=STAGING.LIBRARY
DFSMDA TYPE=FINAL
```

- ▶ Add an IMSACB DD statement to your IMS procedure. Sample JCL is shown:

```
//IMSACB DD DSN=STAGING.LIBRARY, DISP=SHR
```

If you use the IMSACB DD statement, it takes precedence over the DFSMDA member settings.

ACBGEN enhancement made for MOLC

In IMS Version 10, a new parameter of BLDPSB=YES or NO was added to the ACPGEN process. BLDPSB=YES ensures that all PSBs associated with a database (DBD) that is changing in a MOLC are automatically rebuilt. If you are changing a database using MOLC,

¹ If you have not yet enabled global online change, you can refer to a white paper entitled *Position for IMS Version 10's Member Online Change with Global Online Change Migration* for reference, which describes how to implement global online change. You can find it in the IBM *Techdocs* library at: <http://w3-03.ibm.com/support/techdocs/atsmastr.nsf/Web/TechDocs> under the search phrase "global online change."

you do not need to specify all of its associated PSBs or external referenced databases in order to rebuild them. Specifying (or defaulting) BLDPSB=YES in your ACBGEN will ensure that these PSBs and external referenced databases will get refreshed in your active ACBLIB. Previous IMS versions worked as though BLDPSB=NO was specified. BLDPSB=NO indicates that DFSUACB0 does not rebuild PSBs that reference the changed DBD if the changed DBD does not change the physical structure of the DBD. The DBD is rebuilt but PSBs are not necessarily rebuilt (for example, for an exit or randomizer change). You can still specify BLDPSB=NO if you are not planning to use ACB library member online change.

For further information about the ACBGEN process, see the *IMS Version 10: System Utilities Reference*, SC18-9968.

Member online change process

MOLC is based on IMS global online change (GOLC) logic, so there are similarities between these two capabilities. For example, MOLC and GOLC both:

- ▶ Use existing PSBLIB, DBDLIB, and ACBLIB IMS libraries
- ▶ Use existing IMS control block generation PSBGEN, DBDGEN, and ACBGEN processes
- ▶ Use the INITIATE OLC type-2 command
- ▶ Require the CSL
- ▶ Use the same online change-specific staging and active ACBLIBs

To use MOLC, you must first build the updated ACBLIB members. Next, you can create a backup copy of your active ACBLIB and, finally, bring the new members online using the INITIATE OLC command. We now explore each of these steps in more detail.

Building your updated ACBLIB members

You are most likely already familiar with the initial steps of the MOLC process, because they are the same as the full library switch type of online change. These steps are:

1. Update your PSB and DBD source to include your updated ACBLIB members.
2. Run your PSBGEN and DBDGEN to generate new PSB and DBD control blocks, which you must then store in your PSBLIB and DBDLIB respectively.
3. Run your ACBGEN process using your PSBLIB and DBDLIB as input. It creates DMB and PSB control blocks as output, which you must then store in your staging ACBLIB.

After completing these steps, all of your updated ACBLIB members are in your staging ACBLIB, because you have specified it as a target in your dynamic allocation library. You can see a diagram depicting this process in Figure 2-20 on page 53.

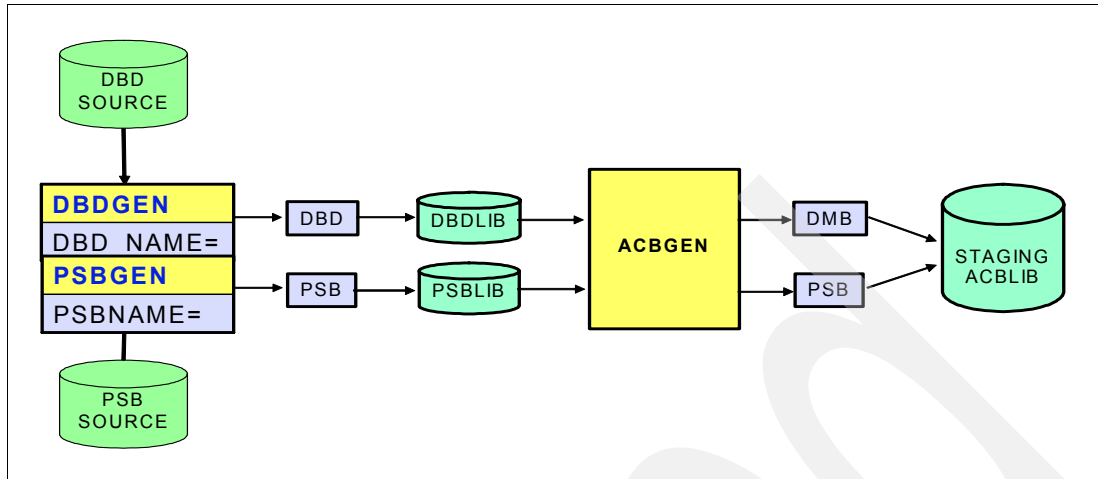


Figure 2-20 Process for building updated ACBLIB members and storing them in the staging ACBLIB

Creating a backup copy of your active ACBLIB

The next step of creating a backup copy of your active ACBLIB is optional, but we recommend it for fallback purposes. You are most likely already familiar with the Online Change Copy utility (DFSUOCU0), having used it with the full library switch type of online change. You can use this utility to copy your existing resource definitions in your active ACBLIB to another library of your choice, such as the inactive ACBLIB. You can see an example of the JCL that accomplishes this in Example 2-7.

Example 2-7 Sample JCL that copies the active ACBLIB to the inactive ACBLIB

```
//DFSUOCU0 JOB LINK,MSGLEVEL=1,REGION=640K,CLASS=N,
              USER=USRT001
//STEP1     EXEC OLCUTL,TYPE=ACTVACB,OUT=G,SOUT=*,
//           OLCLOCL='DUMMY,',OLCGLBL=,SYS=
//
```

Notice that OUT=G is specified with the TYPE=ACTVACB parameter. This signifies that the Online Change Copy utility copies the contents of the active ACBLIB to the inactive ACBLIB, which is determined by the utility using the OLCSTAT data set. The option OUT=O is also available for copying the ACBLIB on a dataset other than the inactive ACBLIB.

For more information about the Online Change Copy utility, see the *IMS Version 10: System Utilities Reference*, SC18-9968.

Bringing your updated ACBLIB members online

Now that you have satisfied the MOLC prerequisites and (optionally) created a backup copy of your active ACBLIB, you can now begin to use MOLC by issuing the INITIATE OLC command with the TYPE(ACBMBR) parameter to copy the updated or new members from the staging library to the active library. If you want to fall back to the previous version of the changed resource, perform a full online change process with a full library switch. We now explore the MOLC commands in more detail.

ACBLIB member online change commands

Just like the full library switch online change capability, you must issue two separate INITIATE OLC commands in order to complete a MOLC. One command prepares your

ACBLIB resources for the change and the other command commits the changes to the active ACBLIB. We now review the syntax and parameters for each of these commands.

Prepare phase command syntax and parameters

To begin the MOLC process, issue the INITIATE OLC command following the syntax shown in Example 2-8. Every parameter shown is required except for OPTION(FRCNRML).

Example 2-8 Command syntax for the prepare phase of member online change

```
INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(name1,name2,namen...)
OPTION(FRCNRML)
```

Here is a description of each parameter:

► **PHASE(PREPARE)**

Performs the member online change prepare phase on each IMS system listed in the OLCSTAT data set. PREPARE is not recovered across an emergency restart and must be reentered after restart if the COMMIT phase did not complete prior to IMS failure.

► **TYPE(ACBMBR)**

Specifies that an ACBLIB member online change process is to be performed for the specified ACBLIB member names. The command response lists all members that will be copied from the staging ACBLIB to the active ACBLIB during the COMMIT command processing.

► **NAME(name1,name2,namen...)**

Specifies that one or more ACBLIB members are to be copied from the staging ACBLIB to the active ACBLIB. The values for NAME can be DBDs or PSBs that are being changed or added. Wildcard characters are not supported in the member names.

If a DBD is being changed, you do not need to specify the associated PSBs on the command, because all of the PSBs that are associated with the changed DBD are copied automatically from the staging ACBLIB to the active ACBLIB².

If a DBD that is being changed or added has external references, the secondary index DBD does not need to be specified on the NAME parameter. The command processing copies all externally referenced members of the DBD from the staging ACBLIB to the active ACBLIB.

► **OPTION (FRCNRML)**

Forces the member online change prepare phase even if one or more required IMS systems in the IMSplex have terminated normally. An IMS is required to participate in the member online change if it is listed in the OLCSTAT data set. A successful member online change specified with the FRCNRML keyword removes an IMS that shut down normally from the list of required IMSs in the OLCSTAT data set. When issuing a MOLC, FRCNRML is the only valid value for the OPTION parameter.

The prepare phase is rejected if any IMS systems in the IMSplex current with the online change libraries are down, unless the FRCNRML option is used. The OLCSTAT data set lists the IMS systems that are current with the online change libraries. These IMS systems either participated in the last global or member online change, or they have cold-started since that point.

The prepare phase fails if any IMS participating in the member online change prepare phase fails, detects an error, or fails to respond in time. Zero, one, or more of the IMS systems can remain in an online change prepare complete state. Issue the QUERY

² Ensure that you have specified BLDPSB=YES (the default setting) in your ACBGEN in order to automatically include the PSBs.

MEMBER TYPE(IMS) SHOW(STATUS) command to display the online change state of all the IMS systems in the IMSplex. Evaluate the QUERY MEMBER TYPE(IMS) output to help you decide whether to issue the INITIATE OLC PHASE(PREPARE) command to try the prepare again or to issue the TERMINATE OLC command to cancel the online change. The IMS systems that are in an online change state remain in an online change state until you cancel or commit the online change. IMS does not automatically cancel online change because of a failure. IMS leaves the IMS systems in their online change states. You must issue the QUERY MEMBER TYPE(IMS) SHOW(STATUS) command and evaluate the output to decide whether to cancel the online change or attempt the online change phase again.

New output fields added for MOLC prepare phase

The output fields for MOLC are the same as for GOLC, but a few more output fields were added in IMS Version 10 that are specific to MOLC. Table 2-3 summarizes the new output fields that you can see after issuing an INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) command.

Table 2-3 New output fields added to the prepare phase of MOLC

Output field name	Meaning
ACBSHR	ACBSHR= value specified in DFSCGxxx or DFSDFxxx PROCLIB member
DBDName	DBD resource name
PSBName	PSB resource name
ADD	The member was found in the staging ACBLIB but not in the active ACBLIB and will be added to the active ACBLIB.
COPY	The member does not have a PDIR for a PSB or DDIR for a DBD in an IMS system. The member will be copied from the staging ACBLIB to the active ACBLIB and will not be available until the PDIR or DDIR is created.
CHNG	The member was found in both the staging and the active ACBLIBs and will be changed in the active ACBLIB.
RFSH	The PSB is refreshed in the active ACBLIB (found in staging and active ACBLIBs but is not changing).
RSCName	Resource name specified on NAME keyword that resulted in error (completion code returned with reason)

For information about the other existing output fields that were not introduced in IMS Version 10, see the *IMS Version 10: Command Reference, Volume 1*, SC18-9700. In Example 2-9, you can see an example of the INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(OLCDB105 OLCDX111) command output as it appears in the TSO SPOC, including some of the new output fields.

Example 2-9 Command output for the INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) command

MbrName	Member	CC	ACBSHR	DBDName	PSBName	ADD
IMS2	IMS1	0	Y			
IMS2	IMS2	0	Y			
IMS2	IMS1	0		OLCDB105		Y
IMS2	IMS2	0		OLCDB105		Y
IMS2	IMS1	0		OLCDB111		Y
IMS2	IMS2	0		OLCDB111		Y
IMS2	IMS1	0		OLCDI111		Y
IMS2	IMS2	0		OLCDI111		Y

IMS2	IMS1	0	OLCDX111	Y
IMS2	IMS2	0	OLCDX111	Y
IMS2	IMS1	0	OLCPB105	Y
IMS2	IMS2	0	OLCPB105	Y
IMS2	IMS1	0	OLCPB111	Y
IMS2	IMS2	0	OLCPB111	Y

Commit phase command syntax and parameters

After you have successfully prepared your ACBLIB resources with the INITIATE OLC PHASE(PREPARE) command, you must commit them to the active ACBLIB using the command syntax listed in Example 2-10.

Example 2-10 Command syntax for the commit phase of member online change

```
INITIATE OLC PHASE(COMMIT)
```

When you issue this command, the updated ACBLIB members are copied directly from the staging ACBLIB to the active ACBLIB. The active and inactive ACBLIB suffixes and the MODID are not updated in the OLCSTAT data set or in the IMS control blocks. Instead, an indicator is written to the OLCSTAT data set specifying either that an MOLC is either in progress or is complete. The indicator is an MOLCINPR literal if an MOLC is in progress and is an MOLCCMPT literal if an MOLC has completed. If the OLCSTAT data set has been updated with one of these indicators and one or more of the IMS systems in the IMSplex fails during MOLC processing, the online change will still succeed for the IMSplex. However in this case, the ACBLIB resources associated with any IMS that had a failure will not be available for use. This is the case regardless of whether the IMS systems are sharing ACBLIBs.

When one or more IMS system fails during MOLC command processing, issue the QUERY MEMBER TYPE(IMS) SHOW(STATUS) command to display the online change state of all the IMSs in the IMSplex. Evaluate the QUERY MEMBER TYPE(IMS) output to help you decide whether to issue the INITIATE OLC PHASE(COMMIT) command to try the commit again or to issue the TERMINATE OLC command to cancel the member online change. The IMSs that are in an online change state remain in an online change state until you cancel or commit the online change.

After you have successfully completed the MOLC commit process, your updated ACBLIB members will be in your active ACBLIB and available for use by your online IMS systems. For further information about using ACBLIB member online change, see the *IMS Version 10: IMSplex Administration Guide*, SC18-9709. You can see a diagram of the MOLC process in Figure 2-21 on page 57.

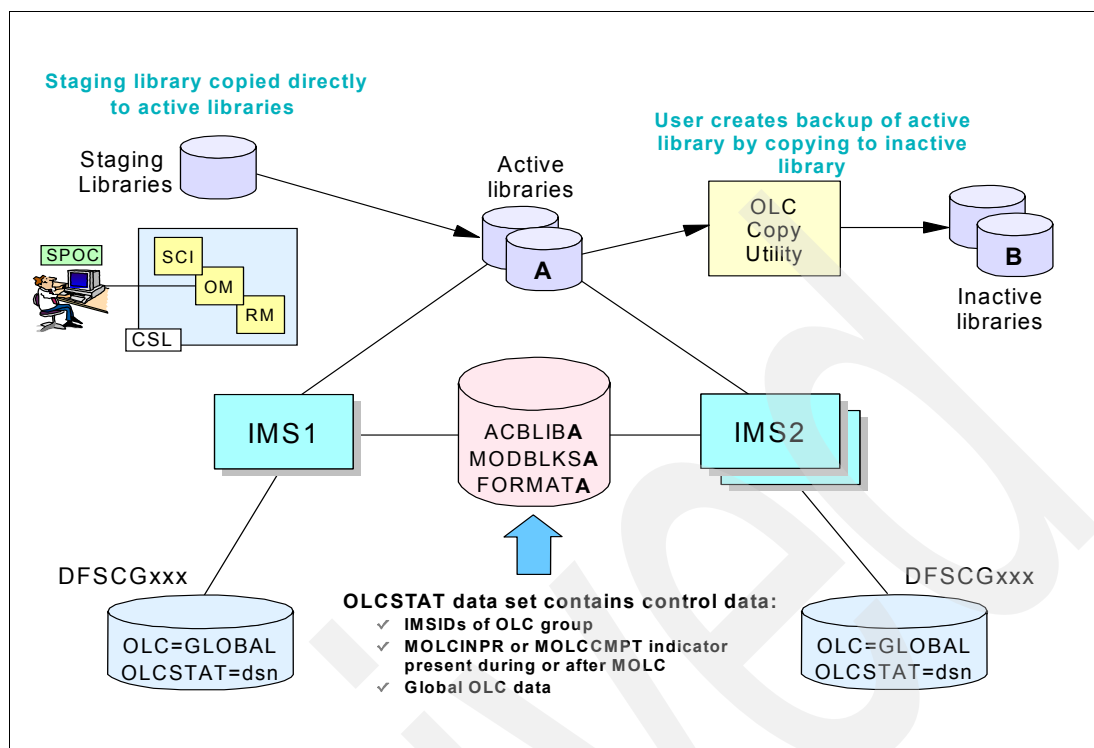


Figure 2-21 The ACBLIB member online change process

New completion and reason codes

IMS Version 10 also has new completion codes that are specific to MOLC. Review Table 2-4 to determine the possible outcomes of MOLC command processing.

Table 2-4 ACBLIB member online change completion codes

Completion code	Meaning
120	Staging library is empty
121	Resource in command has no change
122	Not all PSBs rebuilt for this DMB
123	Staging library level not compatible with current IMS
124	DOPT PSB will not be copied to active ACBLIB by member OLC
125	No PSB rebuilt for this changed DMB
126	ACBLIB member level OLC previous commit failed
127	ACBLIB member level OLC commit member failed
128	ACBLIB member level OLC refresh member failed
129	ACBSHR mismatch
130	Allocation of staging ACBLIB failed
131	Allocation of active ACBLIB failed
132	Open of staging ACBLIB failed
133	I/O error of active ACBLIB

Completion code	Meaning
134	Open active ACBLIB failed
135	I/O error of staging ACBLIB
136	Max concatenations for active ACBLIB

Next, you will see the new reason code of X'4303' if you attempt to issue an INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) command and one of your IMS systems is at a lower version than IMS Version 10. If you receive this reason code, it will be accompanied by a return code of X'10', signifying an *environment error*.

You can find these new completion and reason codes in the *IMS Version 10: Command Reference, Volume 1*, SC18-9700.

Enhanced QUERY OLC command

In IMS Version 10, the QUERY OLC command now shows the specific resources that are involved in an MOLC. A new SHOW parameter was added to support this function, and you can find the syntax for this command in Example 2-11.

Example 2-11 Command syntax for enhanced QUERY command as it pertains to MOLC

```
QUERY OLC SHOW(RSCLIST)
```

You can issue this command validly only when a MOLC is currently in progress, after an INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) has successfully completed. When you issue this command, you will see the ACBLIB members that are being added or copied in the active ACBLIB.

Note: When you specify SHOW(RSCLIST) as a QUERY command parameter, it is mutually exclusive with any other SHOW parameter, such as SHOW(ALL).

The QUERY command also had new output fields added to it in IMS Version 10, which are the same as the new output fields for the MOLC prepare phase, with the exception of the ACBSHR= output field. You can see an example of the QUERY OLC SHOW(RSCLIST) command output as it appears on the TSO SPOC in Example 2-12.

Example 2-12 Command output for the QUERY OLC SHOW(RSCLIST) command

MbrName	CC	DBDName	PSBName	ADD
IMS1	0	OLCDB105		Y
IMS2	0	OLCDB105		Y
IMS1	0	OLCDB111		Y
IMS2	0	OLCDB111		Y
IMS1	0	OLCDI111		Y
IMS2	0	OLCDI111		Y
IMS1	0	OLCDX111		Y
IMS2	0	OLCDX111		Y
IMS1	0		OLCPB105	Y
IMS2	0		OLCPB105	Y
IMS1	0		OLCPB111	Y
IMS2	0		OLCPB111	Y

Enhanced QUERY MEMBER command

Also in IMS Version 10, the QUERY MEMBER command was enhanced to show when a MOLC is in progress. If an INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) command has been completed successfully, the new status of OLCMACB will be output along with global online change status data when the QUERY MEMBER TYPE(IMS) SHOW(STATUS) is issued. Example 2-13 is an example of this command output as it appears on the TSO SPOC.

Example 2-13 Command output for the QUERY MEMBER TYPE(IMS) SHOW(STATUS) command

MbrName	CC	Status	LclStat
IMS1	0		OLCCMT1C,OLCCMT2I
IMS2	0		OLCCMT1C,OLCCMT2I
IMS2	0	OLCMSTR,OLCMACB,OLCCMT1C,OLCCMT2I	
IMS3	0		OLCCMT1C,OLCCMT2I

MOLC log records

New log records were created in IMS Version 10 for the MOLC process. They are summarized in Table 2-5, and you can also find them in the *IMS Version 10: Diagnosis Reference*, GC18-9707.

Table 2-5 ACBLIB member online change log records

Log record	Meaning
X'7002'	Begin write of the new versions of the updated members log record. The log records will contain the names of all the members that are affected by the member online change. If more member names exist than can fit in one X'7002' log record, multiple X'7002' log records are logged.
X'7003'	Write complete ACBLIB member online change log record
X'7004'	Commit start ACBLIB member online change log record
X'7005'	Commit complete ACBLIB member online change log record
X'7006'	Restart cancel ACBLIB member online change record
X'7010'	Terminate ACBLIB member online change process log record

Availability benefits

When you use MOLC to update your ACBLIB resources, only the resources affected by the change are quiesced. This results in higher availability for your other ACBLIB resources that are not affected by the change. These unaffected resources are no longer quiesced during online change processing like they are with the full library switch online change capability. This allows for more concurrent activity among your ACBLIB resources.

To leverage MOLC and maximize resource availability, use it with the Dynamic Resource Definition (DRD) capability to dynamically create runtime resource definitions. An online IMS system can use a new ACBLIB resource only if there is a corresponding resource definition available as well. You can use MOLC to create new ACBLIB resources while using DRD to create the corresponding runtime resource definitions, all without a system outage. To learn more about DRD, see Chapter 6, "Dynamic Resource Definition" on page 133.

2.9.2 Online change enhancement for work in progress

This section describes the IMS Version 10 online change enhancement for resources that have work in progress in a non-Shared Queues environment. This enhancement was completed for the Shared Queues environment prior to IMS Version 10 via APARs, which are summarized in Table 2-6.

Table 2-6 APARs containing the online change work in progress enhancement for Shared Queues

IMS Version	APAR number
IMS Version 7	PQ91164
IMS Version 8	PQ96042
IMS Version 9	PQ96043

Enhancement overview

In a non-Shared Queues environment, IMS no longer rejects the /MODIFY COMMIT or the INITIATE OLC PHASE(COMMIT) commands if an online change indirectly affects transactions that reference ACBLIB or MODBLKS resources that are being changed or deleted. These transactions themselves are not directly affected by the online change, because they do not have any attributes on the TRANSACT macro that are being changed by a MODBLKS online change. Instead, they reference a program or database that *is* directly affected by the online change, such as a:

- ▶ PSB in ACBLIB
- ▶ PSB in ACBLIB that references a changing database
- ▶ PDIR in MODBLKS
- ▶ PDIR in MODBLKS that references a changing database

This enhancement is especially useful for ACBLIB-only online change, because a transaction will never be directly affected by this type of online change. For example, a transaction that has messages queued to it will never cause an ACBLIB-only online change to fail.

Effect on command output

Because online change now has a higher likelihood of succeeding, you might not receive certain command output as before. If you are using local online change, the /MODIFY COMMIT command will no longer fail for indirectly affected transactions with the error message of DFS3452 WORK IN PROGRESS FOR RESOURCES TO BE CHANGED OR DELETED. If you are using global online change, the INITIATE OLC PHASE(COMMIT) command will no longer fail for indirectly affected transactions with the B1 completion code (resource state error) and error text of either QUEUEING or SUSPENDED. Lastly, when using local or global online change, the /DISPLAY MODIFY command will no longer report indirectly affected transactions as having the work in progress status of QUEUEING or SUSPENDED.

2.9.3 Summary of online change enhancements

Both MOLC and the online change enhancement for *work in progress* improve IMS system availability. MOLC accomplishes this by quiescing fewer resources than the full library switch online change capability, therefore allowing more concurrent activity. The online change enhancement for *work in progress* increases the likelihood that the commit phase of the online change will succeed.

2.10 Virtual storage constraint relief

IMS Version 10 provides virtual storage constraint relief (VSCR) in the following areas:

- Database resource adapter enhancement

IMS can now allocate the program communication block (PCB) list that the DRA passes to applications in 31-bit addressable storage.

- Fast Path buffer pool enhancement

The default allocation for Fast Path buffer pools used in FDBR regions is changed from extended common service area (ECSA) to extended private storage.

Database Resource Adapter

IMS can now allocate the PCB list that the DRA passes to applications in 31-bit addressable storage. Savings are dependent on the installation:

- DIRCA size x number of threads
- ACBGEN output message DFS689I assists in estimating DIRCA size

A more substantial impact is that of potential Dependent Region Inter-region Communication Area (DIRCA) storage movement above the 16 MB line for ODBA and CCTL address spaces. The savings are installation dependent and must be requested for the environment to take advantage of the capability. Each thread from the ODBA/CCTL address space has its own DIRCA, and each DIRCA contains the application's copy of the PCBs.

Note: This enhancement provides a benefit for those environments, such as CICS, that need to use larger PSBs and more concurrent threads without impacting storage usage below the 16 MB line.

The DRA, which is an interface to IMS DB from a coordinator controller (CCTL) or a z/OS application that uses the Open Database Access (ODBA) interface, is enhanced to allow IMS to allocate the PCB list that the DRA passes to applications in 31-bit addressable storage. Previously, the PCB list had to be in 24-bit storage, which could have been a contributing factor to private storage abends.

The DFSPRP macro provides the definition for the DRA parameters when building the DRA startup table. A new parameter, PCBLOC, provides the option of defining whether the DIRCA is to be built using 24-bit or 31-bit storage:

- New DFSPRP macro parameter:
 - PCBLOC = 24 | 31 (Default = 24)
 - Used when building the DRA startup tables:
 - DFSPZPxx for CCTL, DFSxxxx0 for ODBA
- Reassembly of DFSPZPxx:
 - Not required if defaulting to 24-bit storage for the DIRCA

Note: A specification of 31 also requires that the applications actually support AMODE 31 and 31-bit PCB addresses. With APAR PK54099, which covers CICS TS 2.2 and CICS TS 2.3, and PK54100, which covers CICS TS 3.1 and CICS TS 3.2, the application will abend with a new CICS abend code ADCF during schedule of the PSB if its AMODE 24 and PCBLOC=31 is in effect.

When a CCTL connector starts up, the DRA INIT call can be used to override the PCBLOC specification in the DRA startup table. A new parameter, PPLLPSO, allows specifications of 31 or 24. Other values are ignored, and the startup defaults to the value specified in the startup table.

Fast Path buffer pool

For IMS Version 10, the default allocation for Fast Path buffer pools used in FDBR regions is changed from extended common service area (ECSA) to extended private storage. The Fast Path buffer pools include the DEDB common buffer pool, as well as the Shared Virtual Storage Option (SVSO) buffer pools. FPBUFF=LOCAL is now the default in the DFSFDRxx member of the IMS.PROCLIB data set.

If your installation requires the Fast Path buffer pools to be allocated in ECSA (although we do not recommend this practice), you must specify FPBUFF=ECSA in the DFSFDRxx member of the IMS.PROCLIB data set.

The following values are valid for the FPBUFF= parameter:

- ▶ **FPBUFF=LOCAL**

Fast Path buffer pools are allocated from extended private storage. This is the default for IMS Version 10 and later.

- ▶ **FPBUFF=ECSA**

Fast Path buffer pools are allocated from ECSA storage.

For more information, refer to *IMS Version 10: Release Planning Guide*, GC18-9717.

2.11 XRF and FDBR enhancements

XRF and FDBR share many characteristics so that enhancements to one usually apply to the other. We detail the IMS Version 10 items here.

Fast Path SVSO

There is no IMS enforced limit on SVSO private pool expansion for XRF tracking in prior releases. The XRF alternate never compresses its SVSO private pool such that when a secondary allocation for the private pool storage is obtained, the storage will not be compressed in an XRF alternate environment. This unnecessarily consumes the CSA/ECSA storage and can lead to a lack of storage for other processes running in the same partition up to the next IPL.

The enhancement has been made so that the SVSO private pool will be compressed automatically during XRF or FDBR tracking.

Type-2 commands

Resource and descriptor definition changes that result from the CREATE, UPDATE, DELETE, or IMPORT commands are picked up by the XRF alternate from the X'22' log records in the log stream and are thence created or updated on the alternate. This also applies to a DBCTL warm standby, FDBR, and an RSR tracker.

MSC

MSLINK status can be set to turn tracing on and off during an XRF Takeover through the command:

```
TRACE SET ON|OFF LINK link TAKEOVER
```

This can also be done with the type-2 command UPDATE MSLINK.

Fast Path buffer pool allocation in FDBR region

FDBR enhancements also include the changed default for the Fast Path buffer pool allocation in FDBR regions. This is described under the title “Fast Path buffer pool” on page 62.

Archived

Database enhancements

This chapter describes the miscellaneous database enhancements in IMS Version 10 and contains:

- ▶ ACBGEN enhancements
- ▶ HALDB index/ILDS rebuild utility free space enhancement
- ▶ Image copy enhancements
- ▶ Sort enhancements of utilities

3.1 ACBGEN enhancements

There are two enhancements to the ACBGEN utility that we describe here:

- ▶ EPCB pool sizing information added to ACBGN output
- ▶ ACBGEN exploitation of storage above the 16 MB line

3.1.1 EPCB pool sizing information added to ACBGEN output

A new message, DFS0943I, has been generated in the ACBGEN output. This message tells you how much storage this PSB will require in the EPCB pool to help you size the pool:

```
DFS0943I PSB PsbName REQUIRES A MINIMUM OF xxxxxxxx AND A MAXIMUM OF  
xxxxxxx BYTES OF STORAGE IN THE EPCB POOL
```

This message is issued for all ACBGENs with Fast Path DEDBs defined. If you use HSSP with DEDBs, we suggest you plan to use the maximum required EPCB size, because the amount needed for the HSSP work area, at its maximum, is not very large.

3.1.2 ACBGEN exploitation of storage above the 16 MB line

The previous limit of program communication blocks (PCBs) in a PSB was determined by the use of *below the line* storage in ACBGEN, so that PSBs with more than approximately 500 PCBs could result in an S80A abend when running ACBGEN. The count includes database and alternate destination PCBs.

IMS Version 10 ACBGEN allocates most of its working storage above the 16 MB line, which eliminates these out-of-storage abends. The benefit to you is that you are now allowed up to 2 500 database and alternate destination PCBs per PSB.

The practical limit for your installation will depend on available pool sizes and, for Batch DL/I, the amount of addressable storage available below the 16 MB line.

The limit of 2 500 PCBs is reduced by the count of secondary indexes, logical databases, physical databases, segments, and fields within the PSB. You can calculate the PSB size, but it is very complex, so the best approach is to run ACBGEN and examine the output.

Note: At the time of writing, there is a limit on the number of objects that can have outstanding updates within a BMP checkpoint interval at the time a system checkpoint is taken. The objects are HALDB partitions, other Full Function databases, and secondary indexes, but not DEDBs or Areas. If there are more than about 300 such objects, the results of the system checkpoint will be unpredictable. This condition is most likely to occur when you are updating many HALDB partitions in a BMP.

Amelioration of this condition is best attempted by minimizing the number of HALDB partitions that are updated in any one BMP checkpoint interval. Alternatively, you can try to reduce the duration between BMP checkpoints or reduce the system checkpoint frequency, while the BMP is running. System checkpoint frequency is changed by using the /CHANGE CPLOG command to increase the number of system log records between system-generated checkpoints; although, this latter choice will require coordination.

3.2 HALDB index/ILDS rebuild utility free space enhancement

IMS HALDB Partition data set recovery often requires that the indirect list data set (ILDS) is rebuilt using the recovered source data. Standard access to the ILDS is in VSAM update mode so the process is slowed by the many CI and CA splits due to the random writes of the ILDS entries (ILEs).

The new option for the HALDB Index/ILDS Rebuild utility (DFSPREC0) enables the ILEs to be written to the ILDS sequentially in load mode. This avoids CI/CA splits during the rebuild process and creates free space according to your data set definition (VSAM DEFINE CLUSTER). The ILDS entries are stored in data spaces, because they are created and sorted before they are written to the ILDS.

In IMS Version 10 without the new option (and IMS Version 9), the utility writes the ILDS entries in update mode, which can cause CI/CA splits in the KSDS, because the entries (ILEs) are written randomly. Any free space request that you specified in VSAM DEFINE is not honored.

You invoke the new “free space” option with one of two new parameters on the utility control statement illustrated in Figure 3-1. The “F” suffix (for Free space) requests that the ILE entries are sorted in data spaces and that VSAM load mode is used for the ILDS.

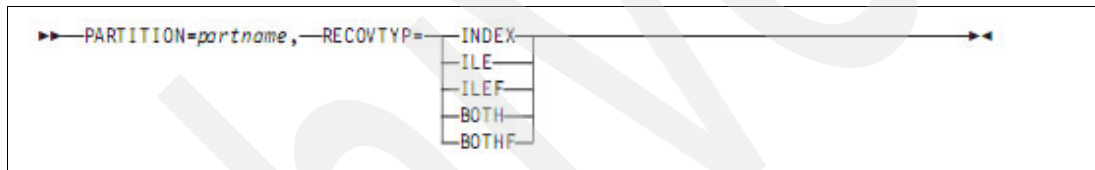


Figure 3-1 DFSPREC0 Utility control statement

Choosing the Free Space option has consequences:

- ▶ It uses up to five data spaces of 2 GB each:
 - Your z/OS IEFUSI exit can limit data space use so that you might require changes to your installation’s IEFUSI exit to be able to exploit this new option.
- ▶ Free space creation can result in larger ILDSs.

Whether you choose the free space option or not will depend on the circumstances.

You need to probably use VSAM update mode when either of these apply:

- ▶ There is only a minimal number of updates to an ILDS.
- ▶ Your system does not allow the utility to use the required number of data spaces.

You need to probably use the VSAM load mode option when any of these apply:

- ▶ You want to add free space to the ILDS for improved performance.
- ▶ You need to rebuild an ILDS that requires a significant number of updates.
- ▶ You want to eliminate CI and CA splits in the ILDS.
- ▶ You want to purge the ILDS of ILEs for segments that no longer exist in the active partition.

If you use the free space option, it will usually result in faster execution of DFSPREC0 when rebuilding an ILDS and might improve the performance of subsequent reorganizations.

You must provide an empty VSAM ILDS when running the DFSPREC0 utility with the free space option. You need to run a job deleting and redefining the ILDS prior to executing the DFSPREC0 utility.

3.3 Image copy enhancements

This section discusses three Image Copy enhancements:

- ▶ Fuzzy user image copy support
- ▶ Flash copy support for Image Copy 2 (IC2)
- ▶ DFSMSdss SET PATCH support for Image Copy 2

3.3.1 Fuzzy user image copy support

Although under the heading of Image Copy enhancements, this is equally a DBRC enhancement.

If you take fuzzy image copies of database data sets outside of the control of DBRC, for example with pack dumps or DFSMSdss DUMP/COPY without using the IMS Image Copy 2 utility, you will have no record of the image copy in DBRC. More to the point, you will have no DBRC support for recovering the data set from that fuzzy copy.

The fuzzy User Image Copy (UIC) support enables you to notify DBRC of the fuzzy copy so that GENJCL RECOV will generate the correct job stream to use the logs after the copy has been restored. DBRC does not know which tool has been used to take the copy.

You can use GENJCL USER to provide the appropriate steps for the copy, restore, and then the recovery functions.

Note that pack dumps and restores can back-level other data sets on the volume.

You identify the type of image copy on the NOTIFY.UIC statement by specifying BATCH if it is a clean (static) copy and CIC plus a STOPTIME if it is fuzzy.

Your procedure for a non-fuzzy (static, clean) user image copy and recovery is as before:

```
Taking clean user IC
  Create clean user IC
  NOTIFY.UIC ... RUNTIME(time1)
```

```
Recovery using clean user IC
  Restore clean user IC
  NOTIFY.RECOV ... RCVTIME
  GENJCL.RECOV ... USEDBDS
```

To exploit the fuzzy copy support, you use a slightly different procedure and use the new keywords:

```
Taking fuzzy user IC
  Create fuzzy user IC
  NOTIFY.UIC ... CIC RUNTIME(time1) STOPTIME(time2)
```

```
Recovery using fuzzy user IC
  Restore fuzzy user IC
  GENJCL.RECOV ... USERIC(time1) or GENJCL.RECOV ... LASTUIC
```

The USERIC(*time1*) parameter in the GENJCL.RECOV command is used to indicate the time of the fuzzy user image copy that was restored. If the restored copy was the last user fuzzy image copy recorded in the RECONs, the LASTUIC parameter can be used instead of the USERIC(*time1*) parameter.

You must restore the user image copy before executing the Database Recovery utility (DFSURDB0). You do not need to NOTIFY.RECOV for a fuzzy copy (however, you must do so after restoring the clean user image copy).

Because the DBDS is not usable before the recovery is performed, we suggest that you use CHANGE.DBDS ... RECOV to set “recovery needed” prior to restoring the user image copy.

The STOPTIME tells DBRC the stop time of the concurrent image copy. GENJCL.RECOV creates the proper JCL for recovery and includes all of the required logs.

You must specify the use of a fuzzy user image copy on the DB Recovery utility (DFSURDB0) control statement illustrated in Example 3-1. The “M” is required in column 63 of the S control statement to indicate that a fuzzy user image copy has been restored and that an M control statement follows. The M is in column 1 with the runtime of the user fuzzy image copy in columns 2-32.

Example 3-1 Database Recovery utility (DFSURDB0) control statement

```
|...+...1...+...2...+...3...+...4...+...5...+...6...
//SYSIN DD *
S DHVNTZ02 VNT0A001 M
M061531233296-0600
```

To illustrate the effects of the commands, we show a LIST.RECON sample here in Example 3-2 and the results of various GENJCL.RECOV executions following that in Example 3-3 on page 70.

Example 3-2 LIST.RECON DBDS

```
DBDS
DSN=IMSTESTL.DHVNTZ02.FKXXI01E TYPE=IMS
DBD=DHVNTZ02 DDN=HIDAM DSID=001 DBORG=HIDAM DSORG=VSAM
-----
IMAGE
RUN = 07.231 10:26:56.7 * RECORD COUNT =49
STOP = 00.000 00:00:00.0 BATCH USID=0000000001
IC1
DSN=IMSVS.DHVNTZ02.HIDAM.IC.IC102620 FILE SEQ=0001
UNIT=SYSDA VOLS DEF=0001 VOLS USED=0001
VOLSER=222222
ALLOC
ALLOC =07.231 10:28:07.5 * ALLOC LRID =0000000000000000
DSSN=0000000001 USID=0000000002 START = 07.231 10:18:09.0
DEALLOC =07.231 10:29:01.1 DEALLOC LRID =0000000000000000
IMAGE
RUN = 07.231 10:28:17.3 * USER CIC USID=0000000000
STOP = 07.231 10:28:17.6
USERDATA= 1ST UCIC TEST OF KSDS
ALLOC
ALLOC =07.231 10:35:31.6 * ALLOC LRID =0000000000000000
DSSN=0000000002 USID=0000000003 START = 07.231 10:18:09.0
DEALLOC =07.231 10:36:47.4 DEALLOC LRID =0000000000000000
```

```

IMAGE
  RUN      = 07.231 10:35:41.5          *  USER CIC   USID=0000000000
  STOP     = 07.231 10:35:41.8
  USERDATA= 2ND UCIC TEST OF KSDS IMAGE
IMAGE
  RUN      = 07.231 10:37:59.1          *  RECORD COUNT =49
  STOP     = 00.000 00:00:00.0          BATCH  USID=0000000003
  IC1
    DSN=IMSVS.DHVNTZ02.HIDAM.IC.IC202620      FILE SEQ=0001
    UNIT=SYSDA                                VOLS DEF=0001 VOLS USED=0001
                                              VOLSER=222222

```

Example 3-3 shows the results of various GENJCL.RECOV commands.

Example 3-3 GENJCL.RECOV results

GENJCL.RECOV LASTUIC generates a recovery using:

Runtime: 07.231 10:35:41.5

GENJCL.RECOV USERIC(072311035415) generates recovery using:

Runtime: 07.231 10:35:41.5

GENJCL.RECOV USERIC(072311028173) generates recovery using:

Runtime: 07.231 10:28:17.3

GENJCL.RECOV USERIC(072311037591) - a BATCH IC

DSP0815I NO USABLE IMAGE COPY AVAILABLE FOR RECOVERY

3.3.2 Flash copy support for Image Copy 2 (IC2)

Flash copy support for Image Copy 2 capability expands your options for database data set image copies. The utility uses the DFSMSdss COPY command with the FASTREP(REQ) parameter to exploit the Fast Replication capability of various devices. It invokes FlashCopy on Enterprise Storage Server® (ESS) and DS8000™ or SnapShot technology on RAMAC Virtual Array (RVA).

The process copies database data sets to other data sets on the same storage system and supports both fuzzy and clean image copies. You do not need to supply DD statements for the output data sets.

You use the Database Recovery utility as usual for these image copies, because it restores the data set with a DFSMSdss COPY command with the FASTREP(PREF) parameter. Here, we contrast concurrent copy and Fast Replication.

Concurrent copy

The concurrent copy uses the DFSMSdss DUMP command with the CONCURRENT parameter. The output is in “dump” format and can be written to any device, including tape or disk.

The copy is done in two phases:

1. Logical copy sets up the copy.

It issues the DFS3121A message when the clean image copy logically completes. This message can be used to indicate that you can start the database in online systems.

2. Physical copy creates the output data sets using the System Data Mover.

Up to four copies of dump data sets can be created, but only two copies can be recorded in the RECONS. The dump data sets can be compressed.

Fast Replication

Fast Replication uses the DFSMSdss COPY command with the FASTREP parameter. The copy is done in a single phase. The output is in the same format as the database data set and must be written to same storage system. The duration is comparable to the logical copy time for concurrent copy.

The utility issues the DFS3141A message when the clean image copy completes, and the message can be used to indicate that you can start the database in online systems.

Only one image copy per database data set can be created, and the same data set option is not allowed. There is no compression option for output data sets, and the DBRC REUSE option is not allowed.

Concurrent copy and Fast Replication

You must register the database in DBRC and also have BWO(TYPEIMS) for fuzzy copies of KSDSs for both Fast Replication and concurrent copy.

KSDSs require SMS management for fuzzy image copies but a CI/CA split causes failure of the process. If you are copying only one data set, the copy is retried up to nine times; however, if you are copying multiple data sets, the copy is not retried.

You use the Image Copy 2 DBDS Control Statement to specify the database data set to be copied. An "F" in position 62 invokes Fast Replications shown in Figure 3-2 on page 72, Figure 3-3 on page 72, and Figure 3-4 on page 72. A blank in position 63 indicates that your control stream includes a DD statement for output data set. The output DD name is in positions 22-29, and the DD statement for output must be in the JCL, whereas an "H" specifies that the output data set is dynamically allocated. You supply the high-level qualifier for the output data set name in the HLQ control statement. A DD statement is not needed for SMS-managed data sets. A DD statement can be supplied for non-SMS managed data sets for the sole purpose of specifying volumes. You can only specify one output data set.

You use the Image Copy 2 HLQ Control Statement to specify the data set name of the image copy data sets that we show in Figure 3-3 on page 72 and Figure 3-4 on page 72. An "H" in position 1 indicates that this is an HLQ control statement, and you set position 3 to "Y" so that a timestamp trailer is appended to the output data set name or "N" if you do not want a timestamp trailer appended to the output data set name. You place the high-level qualifier for output data set name in positions 4-29 and the optional SMS storage class in positions 31-38, along with the optional SMS management class in positions 40-47.

The format of the output data set name when you use dynamic allocation, with and without the timestamp trailer, is illustrated in Example 3-4 on page 72.

Example 3-4 Dynamic Allocation DSName with and without timestamp

If timestamp is not chosen in the HLQ control statement:

hlq.dbdname.ddname

hlq=MY.HLQ, database=DBNAME1, and DD=DDNAME1

MY.HLQ.DBNAME1.DDNAME1

If timestamp is chosen in the HLQ control statement:

hlq.dbdname.ddname.Dyyddd.Thhmmss

For noon on August 30, 2007:

MY.HLQ.DBNAME1.DDNAME1.D07242.T120000

Figure 3-2 shows statements to copy the data set for DD DDMNOPRS in the DBABCDEF database to the output data set specified on the ICABC001 DD using Fast Replication and producing a clean copy.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+  
//SYSIN    DD *  
1 DBABCDEF DDMNOPRS ICABC001                                X   F
```

Figure 3-2 Clean copy, Fast Replication

Figure 3-3 is similar to Figure 3-2 except that the image copy is fuzzy and the output data set is dynamically allocated. The output data set name will be MYHLQ01.TEST.DBABCDEF.DABCDEF1.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+  
//SYSIN    DD *  
1 DBABCDEF DABCDEF1                                         S   FH  
H NMYHLQ01.TEST
```

Figure 3-3 Fuzzy copy, dynamic allocation

Figure 3-4 is similar to Figure 3-3 except that the dynamically created data set name will include the timestamp. It might be MY.TEST.DBABCDEF.DABCDEF1.D07242.T120000.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+  
1 DBABCDEF DABCDEF1                                         S   FH  
H YMY.TEST
```

Figure 3-4 Fuzzy copy, dynamic allocation, timestamp

In Figure 3-5 on page 73, we show the use of a group statement. Fuzzy copies using Fast Replication are made of the three data sets in the DBABCDEF database. The G statement specifies a group name and fuzzy copies using Fast Replication.

	1	2	3	4	5	6

//SYSIN	DD	*				
G	GROUP123				S	F
1	DBABCDEF	DABCDEF1				H
H	YMYH.TEST					
1	DBABCDEF	DABCDEF2				H
H	YMYH.TEST					
1	DBABCDEF	DABCDEF3				H
H	YMYH.TEST					

Figure 3-5 Using the group statement

DBRC records new image copy types for Fast Replication image copies:

- ▶ SMSONLC - fuzzy Fast Replication image copy (also used by IMS High Performance Image Copy, product number 5655-N45)
- ▶ SMSOFFLC - clean Fast Replication image copy (also used by IMS High Performance Image Copy)

There are new GENJCL.IC parameters that you can use:

- ▶ SMSONLC - generates IC2 JCL for fuzzy IC with Fast Replication
- ▶ SMSOFFLC - generates IC2 JCL for clean IC with Fast Replication
- ▶ HLQ(*hlq*) - specifies high-level qualifier for the output data set name
- ▶ DSNISUF | NODSNISUF - date and time stamp as suffix in output data set name
- ▶ STORCLAS(*name*) - specifies Storage Class use by ACS routines for allocating image copy data set
- ▶ MGMTCLAS(*name*) - specifies Management Class use by ACS routines for allocating image copy data set

Database Recovery

The Database Recovery Utility (DFSURDB0) reads Fast Replication image copies when you invoke recovery by using the DBRC information to determine that the IC is a Fast Replication IC. You do not specify it.

It uses COPY with FASTREPLICATION(PREFERRED). If the Fast Replication copy is on the same storage system as the target data set and the storage system supports Fast Replication, Fast Replication is used to restore the database data set from the image copy. Otherwise, a traditional copy is used.

The DBDS DD statement is optional when using Fast Replication IC input: a 'D' in position 64 of the control statement indicates that a DD statement is not used. The DBRC information is used to allocate the data set.

To match the new GENJCL.IC parameters, you use new GENJCL.RECOV parameters DBDSDD(default) or NODBSDD:

- ▶ If DBDSDD is specified (or defaulted), the output DBDS DD statement is created in the JCL and a blank is placed in position 64 of the DB Recovery utility control statement. The DB Recovery utility gets data set information from the DD statement.
- ▶ If NODBSDD is specified, the output DBDS DD statement is not created and a 'D' is placed in position 64 of the DB Recovery utility control statement. The DB Recovery utility

gets data set information from the RECONS. This is only valid with Fast Replication IC input.

In either case, positions 13-20 of the utility control statement contain the DD name for the output DBDS as we show in Figure 3-6.

```
-----1-----2-----3-----4-----5-----6-----7--
S  DBABCDEF DBDSABCD                                     D
/*
```

Figure 3-6 Database Recovery control statement

3.3.3 DFSMSdss SET PATCH support for Image Copy 2

DFSMSdss has SET PATCH capabilities to change the default actions. You can use them to change wait/retry values or change actions when orphaned CAs are found. Image Copy 2 has added control statements for SET PATCH as shown in Figure 3-7 whereas previous IMS releases required a module zap to use SET PATCH. GENJCL.IC now includes support for generating the SET PATCH control statement to provide a user friendly interface for changing DFSMSdss defaults. How you use Patch Bytes is described in the Syntax - Auxiliary Commands section of *z/OS V1R9.0 DFSMS Storage Administration Reference*, SC26-7402. Patch bytes and values are documented in the *z/OS V1R9.0 DFSMSdss Storage Administration Guide*, SC35-0423.

```
-----1-----2-----3-----4-----5-----6-----7--
P 111=aa 222=bb 333=cc 444=dd 555=ee
/*
```

Figure 3-7 SETPATCH control statement

In Figure 3-7, we show the layout of the patch control statement with five patches, each separated by one blank. You can supply multiple patch statements. The 'P' in position 1 indicates a SET PATCH control statement and positions 3 to 71 contain up to 10 patch entries. The numerics identify the patch byte of interest and the alpha characters represent the byte values to be set. IMS does not check that the target is a valid patch entry for DFSMSdss. A missing entry terminates the input for that statement so you must supply the first entry on every statement.

3.4 Sort enhancements of utilities

The two processes that dynamically invoke your installation's sort/merge program, namely Prefix Resolution (DFSURG10) and Change Accumulation (DFSUCUM0), have had their default for the sort/merge storage changed from 200 K to MAX.

If you do not override the CORE= parameter of the utilities' EXEC statement, IMS will use the installation's maximum Sort/Merge storage allocation.

This eases the limit on work data sets for some vendors' sort/merge utility where the Unit Control Blocks (UCBs) are below the 16 MB line.

Transaction Manager enhancements

IMS Version 10 provides many Transaction Manager (TM) enhancements in the areas of connectivity and integration, manageability, and scalability. This chapter concentrates on the following topics:

- ▶ Advanced Program-to-Program Communication (APPC) enhancements
- ▶ Removal of Basic Telecommunications Access Method (BTAM) support
- ▶ Multiple Systems Coupling (MSC) enhancements
- ▶ Sysplex serialized program management
- ▶ Transaction level statistics enhancements
- ▶ TM-related Fast Path enhancements

Connectivity and integration-related enhancements for IMS TM are described later in this book in Chapter 8, “Connectivity enhancements” on page 203 and Chapter 9, “Integration enhancements” on page 235, and manageability enhancements are described in Chapter 5, “Manageability enhancements” on page 105.

4.1 APPC enhancements

APPC/IMS has been enhanced in the following three areas:

- ▶ Timeout granularity improvement with support for the APPC/z/OS timeout in seconds
- ▶ Capability for APPC and Open Transaction Manager Access (OTMA) clients to issue the /LOCK and /UNLOCK commands
- ▶ Local logical unit (LU) support

4.1.1 APPC timeout granularity

The APPC/MVS capability for timeout has been supported by IMS since IMS Version 6. With the introduction of z/OS V1R7, APPC/z/OS enhanced its timeout support to specify a lower level of granularity. Prior z/OS releases supported timeout in minutes with the shortest value being one minute. Even one minute, however, can be too long of a time when resources are held. With IMS Version 10 and z/OS V1R7, the timeout capability as provided in the first parameter of the APPCIOT keyword and in the /CHANGE APPC TIMEOUT command allows a value that can be specified in seconds.

You can specify the APPCIOT value with a seconds value in the DFSDCxxx member of IMS.PROCLIB as follows:

```
APPCIOT=(mmmmA:ssA,mmmmB)
```

where *mmmmA* is the timeout value for APPC/z/OS in minutes (0-1440) and refers to an IMS wait for a requested APPC/z/OS service to complete, and *ss*= 0-59. The second parameter, *mmmmB*, is the timeout value for APPC/IMS in minutes (0-1440), which was introduced in IMS Version 9. This refers to an APPC/z/OS wait for an IMS process to complete. For example, the timer begins when IMS receives a message from APPC/z/OS and this IMS timeout (also called *Timeout 2*) is for the whole life of a transaction. For conversational transactions, it is reset for every iteration, for example, the send of a reply, error message, or deallocate. This second value is similar to putting a timeout on a queue to queue response.

For the /CHAnge command, you specify:

```
/CHANGE APPC TIMEOUT mmmm:ss
```

The ability to set the timeout in seconds is particularly helpful when slowdowns in the network occur or when APPC clients that are unable to respond in a timely manner cause IMS dependent regions to hang. Another benefit is one that can affect command processing when a slow or non-responding client impacts the IMS command task DFSCMT10.

For example, when an APPC node sends input for a transaction that is stopped, IMS replies to this condition by sending a DFS065 message. This error message is sent under the IMS command task, which waits for a response from the APPC node. When this node fails to respond, the task hangs until the wait is broken by the APPC/z/OS timeout facility. This task, however, is quite vital in the IMS environment and plays a major role in handling most commands. When the condition that we just described happens, a slowdown in IMS throughput can result. An APPC timeout value of 1 minute, therefore, can be too long when suspending the command task in a production environment.

When a transaction is terminated because of a timeout, messages DFS1965E and DFS1995E are sent to the MTO terminal and the z/OS console:

```
DFS1965E APPC/MVS CALL FAILURE,FUNCTION=aaaaaaa,REASON CODE=xyyy,  
RETURN CODE=xxxxxxxx,LUNAME=nnnnnnnn
```

Message DFS1995E follows message DFS1965E and explains the return code and reason code of the previous DFS1965E message. DFS1995E includes the APPC/MVS error message provided by the APPC/MVS Error Extract Service. This APPC/MVS message, starting with prefix ATB, is described in *z/OS MVS Programming: Writing Transaction Programs for APPC/MVS*, SA22-7621.

4.1.2 /LOCK and /UNLOCK command support

IMS Version 10 lifts the restriction of issuing the /LOCK and /UNLOCK commands from APPC and OTMA clients. The support allows the DATABASE, PROGRAM, and TRANSACTION keywords, but there is no support for LTERM, PTERM, or NODE. Note that the commands themselves are not changed.

We show an example of an APPC client that specifies the /LOCK command as a TPNAME on the ALLOCATE request in Example 4-1. The remainder of the command is in the subsequent SEND_DATA verb. IMS responds with a DFS058 LOCK COMMAND COMPLETED, which simply means that the command was accepted and processed by IMS. Note that the capability to send IMS commands from an APPC client is not new.

Example 4-1 /LOCK command with APPC

```
ALLOCATE LUNAME=IMSLU, TPN=/LOCK. ...
SEND_DATA DB database1
RCV_AND_WAIT                                <-----DFS058 LOCK COMMAND COMPLETED
DEALLOCATE
```

Likewise, your OTMA clients, such as IMS Connect and MQ, can also send in the command requests. Each of your OTMA clients provides its own interface for remote applications. We show a request coming in from a remote application through IMS Connect in Example 4-2.

Example 4-2 /LOCK command through IMS Connect

```
CONNECT
WRITE LLLL 11zz IRM (specify /LOCK in IRM_TRNCOD) 11zz /LOCK DB database1 EOM
READ                                     <-----DFS058 LOCK COMMAND
COMPLETED
DEALLOCATE
```

4.1.3 Local LU support

APPC conversations are held between two Transaction Programs (TPs) executing on LUs in one or more execution environments. The SNA 6.2 LU in z/OS is defined as the base LU in the APPCPMxx member of SYS1.PROCLIB, which contains the APPC/z/OS definitions. You also specify local LUs in the same member as being associated with APPC/IMS. Note that an LU must be unique to an IMS system in an IMSplex.

IMS supports three modes of APPC operation:

- ▶ Implicit

IMS manages the conversation so that a traditional IMS message program can process the message text.
- ▶ Explicit

IMS schedules the application program, but the program must manage the conversation with the appropriate APPC verbs.

► Modified

Modified is a misnomer for mixed, where the conversation is managed by IMS and a message program is scheduled as with Implicit, but the program can also initiate asynchronous outbound conversations by using the relevant APPC verbs.

IMS Version 10 enables your IMS applications to control where and how the message is sent for alternate TP program communication block (ALTPCB) APPC asynchronous outbound requests when you specify which LU name to use on the outbound ALLOCATE request through a new keyword in the DFS62DTx descriptor. We illustrate this in Example 4-3 where the ALTPCB destination name is the descriptor name. If you do not provide the OUTBND parameter, the default base LU name is used. Additionally, you have a new startup parameter, APPCLLU, to specify whether the incoming LU name (if different from the base LU) is to be used for any associated asynchronous IOPCB outbound conversations.

Example 4-3 New OUTBND keyword in the LU62 descriptor DFS62DTx

```
U L62A LUNAME=L62RMT1 TPNAME=MYTRAN MODE=L62MODE1
U L62A SYNCLEVEL=C OUTBND=IMS62A
```

Your partner APPC application sees the same name when it ACCEPTs the conversation. If the OUTBND keyword is not available, the default base LU name is used.

Note: The DLI API has not been changed to add the OUTBND keyword to the CHNG call's LU 6.2 options.

Your LU 6.2 Edit Exit Routine (DFSLUEE0), if it exists, is always called for inbound and outbound conversations managed by IMS. Word 13 in the exit interface points to the local LU name and can be examined, as well as modified, by the exit code.

You need to remember that asynchronous APPC conversations are those conversations that you send without waiting for a reply. The verb set you use is:

- Allocate
- Send_data
- Deallocate

Synchronous conversations, however, are those conversations that send a message and wait for a reply using the following verb set:

- Allocate
- Send_data
- Receive_and_wait
- Deallocate

The local LU support that we describe here applies to asynchronous conversation requests only.

The example that we show in Figure 4-1 on page 79 illustrates how if you specify different local LUs in the OUTBND parameter of two descriptors, the remote partner sees the different names. If the remote partner is IMS, the target application, PGM1 in this example, sees a different LTERM name for each message.

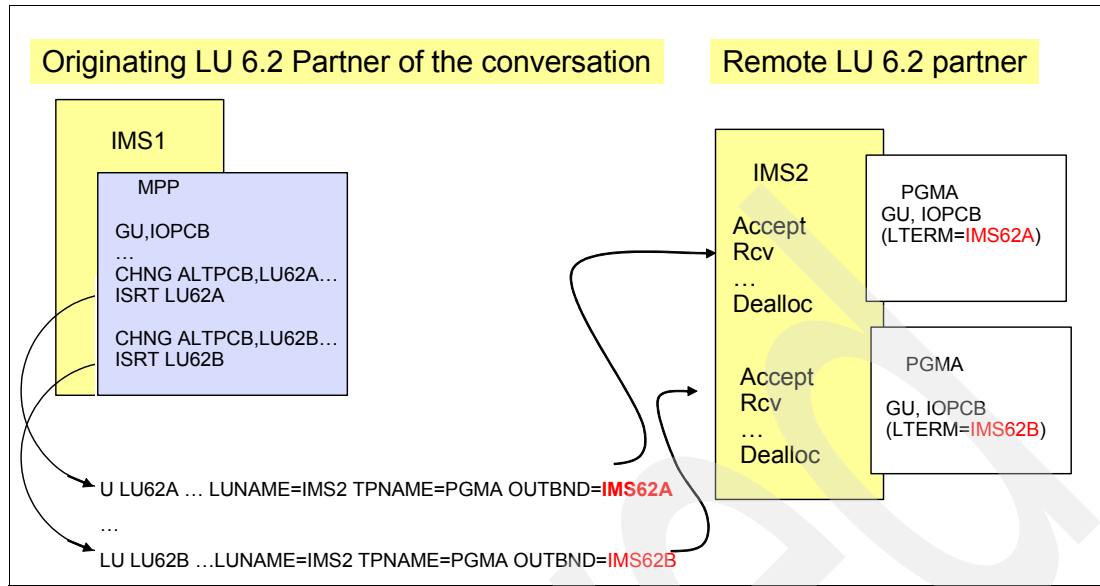


Figure 4-1 OUTBND example using IMS systems

Note that an APPC/IMS implicit environment is one where IMS functions as the LU 6.2 partner on behalf of the IMS application program. In this environment, IMS places the Partner LU name in the LTERM field of the IOPCB, and you do not define LU 6.2 devices to IMS as LTERMS. A possible benefit for you of using the OUTBND capability is that your receiving IMS application can branch to different logic based on the LTERM name. The remote partner can, of course, be any LU 6.2 partner.

IMS Version 10 also provides a new startup parameter in the DFSDCxxx member of IMS.PROCLIB. APPCLLU affects asynchronous outbound messages that are inserted to the IOPCB. For this situation to occur, the message that the IMS application processed with a GU IOPCB had to originate in an APPC partner as an asynchronous inbound message.

We show two examples to illustrate the difference between the APPCLLU specifications of N and Y. In Figure 4-2, where APPCLLU=N, the IOPCB asynchronous output reply is sent using the base LU name associated with IMS2 regardless of which LU name was used for the inbound asynchronous conversation.

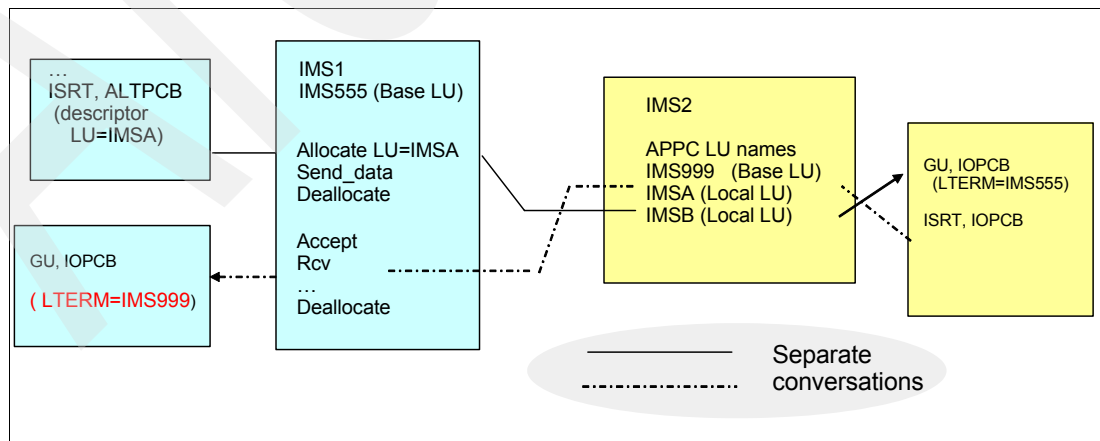


Figure 4-2 Local LU APPCLLU=N

In Figure 4-3, where APPCLLU=Y, IMS2 uses the local LU name of IMSA which is the same name used on the inbound asynchronous request. This capability is of value when the remote application is designed to expect a response from a specific partner LU name.

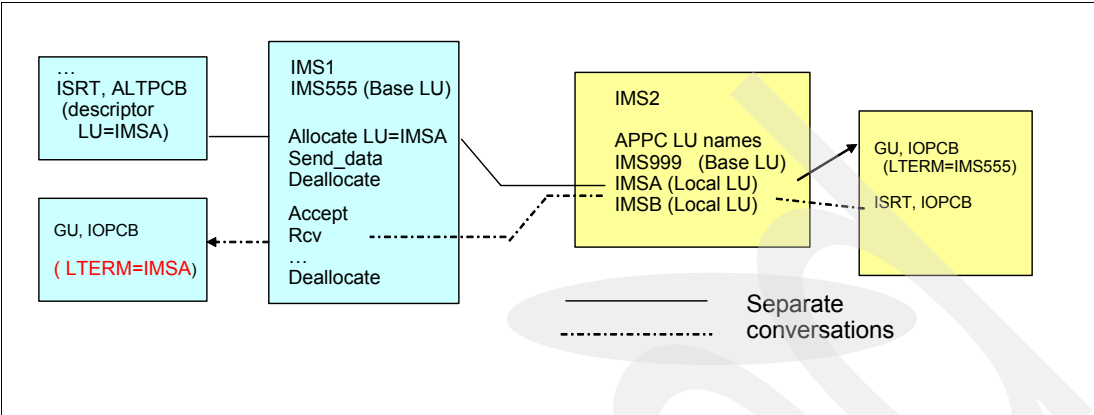


Figure 4-3 Local LU APPCLLU=Y

The /CHANGE DESC and /DISPLAY DESC provide support for the local LU (locallu) capability as we show in Example 4-4:

- ▶ The /CHANGE DESC descriptor OUTBND locallu command modifies the descriptor to use the specified locallu name for any new outbound asynchronous conversations associated with the specified descriptor. Note that this impacts messages that use the descriptor after the command has been issued. Messages that are already on the queue but have not yet been sent are not affected by the change.
- ▶ The /DISPLAY DESC command has also been enhanced to support the local LU. If the OUTBND keyword exists in a descriptor, its value is displayed under the OUTBNDLU column. If the value for a descriptor displays as blanks, the base LU will be used.

Example 4-4 DISPLAY DESC output includes local LU names

DESC	LUNAME	MODE	SIDE	SYNLEVEL	TYPE	OUTBNDLU
L62A	IMS2	L62MDE02		CONFIRM	MAPPED	IMS62A
	TPNAME : PGMA					
L62B	IMS2	L62MDE02		CONFIRM	MAPPED	IMS62B
	TPNAME : PGMA					
L62TM	L62RMT1	L62MDE09		CONFIRM	MAPPED	
	TPNAME : RMTPGM					

Remember, as we said earlier, remote partners that receive an asynchronous conversation request from IMS have the ability to determine the LU name that IMS uses when sending a request. The local LU capability provides the opportunity for different names other than the base LU to be provided. Also, note that LU names are specific to IMS systems and cannot be shared within IMSplex members. Finally, any modifications that occur as a result of you issuing a /CHA command affect new messages that are created after the change, not those already on the message queue.

Local LU security

A security check is done using the target LU name for IMS (Base or Local) when an inbound conversation is allocated with APPC/z/OS in a secured environment. If authorization is granted, APPC/z/OS passes the RACF object to IMS. The RACF object, similar to an ACEE, is used by the IMS security call to check if the user ID is authorized for the transaction before scheduling the request. In the situation where the message was sent to IMS using a local LU,

the local LU name is used. After this is done, the RACF object is deleted. If RACF=FULL has been specified for APPC/IMS, the ACEE must again be built for the dependent region. Prior to IMS Version 10, the ACEE is always built using the base LU. If a mismatch occurs because a secured message was sent in using a local LU, but the dependent region ACEE using the base LU does not authorize the user, the queued transaction will fail authorization. IMS Version 10 addresses this situation by using the applicable LU (Base or Local) that was used for the inbound message when building the dependent region ACEE.

4.2 Disable support of BTAM

IMS continued to support the BTAM macros through IMS Version 9 even though IBM withdrew marketing and service of BTAM products several years ago. IMS Version 10 removes this BTAM support and relates to the terminal types that we list in Table 4-1. Devices, such as Spool, Reader, Printer, Punch, Tape, and Disk are not affected.

If you include a macro statement operand that has an unsupported BTAM terminal specification in your IMS STAGE 1 system definition input, you will receive the following warning message in the SYSOUT:

```
G411  MACRO  STATEMENT  ASSOCIATED  WITH  AN  UNSUPPORTED  BTAM  TERMINAL
```

A severity code of 2 will be issued to allow system definition to continue.

Table 4-1 Unsupported BTAM Terminal Types

BTAM device type	Comments or other specifications
1050	Switched Terminal
2740	Non-Station-Control
2740	Non-switched, model 1
2740	Non-switched, model 2
2740	Switched Terminal, model 1
2741	Non-switched
2741	Switched Terminal
2260	Local
2780	N/A
3270	Remote, Non-switched
3270	Local
3270	Switched Terminal
3275	Switched Terminal
3741	Switched Terminal
SYSTEM/3	N/A
SYSTEM/7	BSC, BSC and Contention
SYSTEM/7	Start/Stop, Start/Stop and Contention

IMS system definition macros that previously supported BTAM are shown in Table 4-2.

Table 4-2 IMS system macros that previously supported BTAM

IMS system macros				
CONFIG	CTLUNIT	DCLIST	IDLIST	LINE
LINEGRP	MSPLINK	NAME	POOL	STATION
SUBPOOL	TERMINAL	TYPE		

4.3 Multiple Systems Coupling (MSC) enhancements

Several aspects of Multiple Systems Coupling (MSC) have been enhanced in IMS Version 10, including:

- ▶ Removal of exits replaced by DFSMSCE0
- ▶ MSC bandwidth capability
- ▶ Link statistics through a command
- ▶ Support of VTAM Generic Resources on MSC links

IMS Version 10 removes the routing exit invocations that were replaced by the TM and MSC Message Routing and Control User Exit (DFSMSCE0) beginning with IMS Version 7. DFSMSCE0 consolidated and replaced the following exits:

- ▶ MSC Terminal Routing Exit (DFSCMTR0)
- ▶ MSC Input Message Routing Exit (DFSMPRT0)
- ▶ MSC Link Receive Routing Exit (DFSCMLR0)
- ▶ MSC Program Routing Exit (DFSCMPR0).

Although the older exit routines were supported concurrently with DFSMSCE0 in IMS Version 7, 8, and 9, they are now being removed in IMS Version 10 and will no longer be supported or called. DFSMSCE0 must be used.

Note: DFSMSCE0 can be used in both MSC and non-MSC environments, although not all routing options will apply to non-MSC systems.

To improve the performance and bandwidth requirements of high-volume MSC systems, many installations use a large number of parallel MSC links between pairs of IMS systems today. This scheme can complicate the operations of the IMS systems and adds to the complexity of the system configurations, as well as to load balancing schemes. IMS Version 10 introduces several enhancements to increase MSC bandwidth. These include:

- ▶ Blocking multiple messages and responses into a single buffer when sending messages across the MSC links. In previous releases, IMS only sent one message or response at a time.
- ▶ Reducing the logger I/O operations by reducing the number of CHECK WRITE calls.
- ▶ Expanding the maximum link buffer size to 64 K (it was 32 K) so that more messages and responses can fit into a buffer. Additionally, the initial link buffer sizes set during the IMS system definition process can be increased or decreased and displayed dynamically with IMS commands.

A secondary issue associated with MSC bandwidth is the difficulty that your IMS operators and system programmers can encounter when MSC link performance is inadequate due to a message backup or poor performance. Inadequate performance data in previous IMS

releases often resulted in days or weeks of analyzing log records and traces to resolve the issue. In IMS Version 10, this issue is resolved with an enhanced link query statistics command.

4.3.1 Removal of exits replaced by DFSMSCEO

IMS Version 10 only supports the TM and MSC Message Routing and Control User Exit Routine (DFSMSCEO) for routing. DFSMSCEO was introduced in IMS Version 7 to provide an opportunity to migrate over several releases. There is no longer any support for the following older routing exit routines:

- ▶ DFSCMTR0
- ▶ DFSNPRT0
- ▶ DFSCMLR0
- ▶ DFSCMPR0

If you specify NOMSPEX/MSPEXIT and NOMSLEX/MSLEXIT parameters on the COMM macro during system definition, they will be ignored but you get a G092 warning message. Previous releases of IMS allowed specification of the parameters NOMSPEX/MSPEXIT to exclude/include the MSC Program routing exit (DFSCMPR0) and NOMSLEX/MSLEXIT to exclude/include the MSC Link Receive routing exit (DFSCMLR0). IMS no longer recognizes these routines even if they are included in the IMS resource library.

Note: If the older exit routines still exist in the IMS release from which the migration is being done, migration to DFSMSCEO becomes a required action for the IMS Version 10 migration. The manual *IMS Version 10: Exit Routine Reference*, SC18-9708 documents a sample DFSMSCEO exit along with the user edit parameter list macro (DFSMSCEP), which provides information about using and customizing the exit. The sample itself is in the IMS.SDFSSMPL library. To include the DFSMSCEO replacement exit in the IMS system, it must be bound (link-edited) into IMS SDFSRESL (or a concatenated library).

4.3.2 MSC bandwidth capability

The definition of MSC physical links provides a specification for a BUFSIZE on the MSPLINK system definition macro. The buffer sizes specified are fixed at system definition. When the link is started, an input and output buffer is acquired and held for the duration of the link restart (that is, link active), at the specified size to send and receive data (messages) across the link. Because each side (partner) of the link has a send and receive buffer, a message or response can be simultaneously sent each way. However, only one message or response is sent per buffer, even if the buffer is large enough to hold multiple messages/responses. Another message is not sent until the partner IMS responds that it has received, queued, and logged the message. For high-volume systems, the wait associated for a freed buffer can be unacceptable. To get around this issue, high-volume systems oftentimes are defined with a large number of parallel links to support the concurrent traffic from one IMS to another.

The MSC bandwidth mode in IMS Version 10 is a mechanism to determine whether to send multiple messages in one buffer. In non-bandwidth mode, MSC sends a maximum of one message or response per I/O operation (that is, send or write). In bandwidth mode, IMS attempts to maximize the capacity of a link by sending as many messages that are queued and ready to go, and responses that are owed for messages received, in the same buffer. By increasing the link buffer size, more and more messages and responses can be sent simultaneously.

Note that BANDWIDTH mode is not a system definition option. It can only be set ON with a command. The default is non-bandwidth mode, which allows MSC to function as in previous releases. In Example 4-5, we provide an example of the MSC setup.

During system definition, three macros provide the mechanism to define the connection between two IMS systems. The MSPLINK macro has not changed, but the range in values that you can specify for BUFSIZE reflects the new minimum and maximum limits. The MSLINK has changed to provide a label. In this example, the name is LNK12V02 and is important, because it will be used in our UPDATE command.

After the system definition process is complete and IMS is initialized, the parameters on the MSLINK can be modified. Note that in this example, our UPD command identifies the specific MSLINK named LNK12V02. Two changes are made: the first is the setting of BANDWIDTH Mode to ON and the second is the increase of the BUFSIZE to 4096. Alternatively, we could issue the command as a type-1 command using the /UPD form.

The update command has been enhanced to provide a mechanism to change the characteristics of the link associated with the name provided. The name is either the label value on the MSLINK definition or the default DFSLKxxx value. Possible actions that control the bandwidth capacity of the link include increasing and decreasing link buffer sizes, setting a higher limit for buffer size, and setting the bandwidth mode off and on. The command can be entered as either a type-2 (UPDATE) command, which uses the Operations Manager (OM) or a type-1 (/UPD) command, which does not require OM. Any accepted changes are kept across a warm start.

Example 4-5 Bandwidth mode setup

System Definition

```
PLNK12V  MSPLINK TYPE=VTAM,NAME=L6APPL1,SESSION=3,BUFSIZE=1024
LNK12V02 MSLINK PARTNER=AK,MSPLINK=PLNK12V, OPTIONS=FORCSESS
LINK12V1 MSNAME  SYSID=(20,10)
```

Update

```
UPD MSLINK NAME(LNK12V02) SET(BANDWIDTH(ON), BUFSIZE(4096))
```

Response for: UPD MSLINK NAME(LNK12V02) SET(BANDWIDTH(ON),BUFS...

MSLink	MSLink#	MbrName	CC
LNK12V02	10	IMS1	0

OR

```
23/UPD MSLINK NAME(LNK12V02) SET(BANDWIDTH(ON) BUFSIZE(4096))
DFS3476I UPDATE MSLINK COMMAND COMPLETED, CC= 0
```

After we have made the changes to the link, we can use either the QRY command or the /DIS command to display the link and ensure that the changes have actually been made. See Example 4-6 for our illustration.

Example 4-6 Confirm link changes with QRY or /DIS

Query

```
QRY MSLINK NAME(LNK12V02) SHOW(ALL)
```

Response for: QRY MSLINK NAME(LNK12V02) SHOW(ALL)

More:

MSLink	MSLink#	MbrName	CC	MSPLink	CID	PID	RecdCnt	SentCnt
LNK12V02	10	IMS1	0	PLNK12V	00000000	AK	0	0

Bufsize Bandwidth Lc1Stat

4096 ON STOCOMM, IDLE, COLDSESS, FORCSESS, TRACE

Display

87/DIS LINK 10

DFS000I	LINK	PARTNER	RECD	ENQCT	DEQCT	QCT	SENT
DFS000I	10	AK	0	0	0	0	0
DFS000I		PSTOPPED	IDLE	COLD	FORCE	TRA	

88/DIS LINK 10 OPTION BUFSIZE

DFS000I	LINK	LINKNAME	BUFSIZE	STATUS
DFS000I	10	LNK12V02	4096	PSTOPPED IDLE COLD TRACE
DFS000I		FORCE	BANDWIDTH	

4.3.3 MSC performance results

The figures in Example 4-7 show the time in seconds to send 1 000 queued messages, where the data length = 100 bytes + PFX. These results are from a test comparing the use of Bandwidth Mode against a variety of buffer sizes. Note that the test results vary based on buffer size with a controlled message size and queue depth. It was a defined and controlled environment and the results are intended as an example and *not* to provide exact percentages. Results vary based on:

- Buffer sizes
- Traffic (number of messages that are queued and ready to be sent)

Example 4-7 MSC performance results

CTC Shared Queues

non-bandwidth mode 4k Time	= 8.074355 sec
bandwidth mode 4k Time	= 5.279757 sec = 34% less
bandwidth mode 8k Time	= 5.036957 sec = 37% less
bandwidth mode 16k Time	= 4.588605 sec = 43% less
bandwidth mode 32k Time	= 4.47057 sec = 44% less

VTAM Shared Queues

non-bandwidth mode 4k Time	= 7.78207 sec
bandwidth mode 4k Time	= 6.97356 sec = 10% less
bandwidth mode 8k Time	= 5.252516 sec = 32% less
bandwidth mode 16k Time	= 4.829623 sec = 37% less
bandwidth mode 32k Time	= 4.446261 sec = 42% less

MSC bandwidth mode is only available on IMS Version 10 to IMS Version 10 connections.

Turning bandwidth mode ON can also impact logger I/O. As a general rule, there are two CHKWs in the path of a recoverable message that is sent across an MSC link: one on the send side when the last part of a message is sent and one on the receive side when the received message is enqueued. There are no CHKWs issued for non-recoverable messages. Note, however that even though a message that is sent from a front-end IMS to a back-end system is non-recoverable (therefore, no CHKWs), its response is always considered recoverable and will incur CHKWs. Therefore, recoverable remote transactions incur four CHKWs: two for the message going to the back-end and two for the response. Non-recoverable remote transactions incur two CHKWs: none for the non-recoverable message that is sent to the back-end and two for the reply, which IMS always sets to recoverable.

The choice of bandwidth mode, however, does impact the number of CHKWs that can be issued in an IMS Version 10 environment. Bandwidth mode potentially reduces the number of CHKWs issued, because only one CHKW is written per send buffer regardless of how many messages are contained in the buffer. For example, if there are five messages in the buffer, then only one CHKW is written rather than five. Note that on the receive side, the back-end IMS continues to issue one CHKW per message. When the response messages are ready, the back-end IMS will attempt to buffer as many messages as can fit in a buffer and issue one CHKW for the buffer. The front-end IMS, which is the receiving system for the responses, will issue one CHKW per message.

Non-bandwidth mode provides the same processing as pre-IMS Version 10 systems. For example, if the front-end IMS has five messages to send, these messages are all sent in separate buffers and five CHKWs will occur, one for each message. On the receive side, IMS will again issue one CHKW per message. The same processing and number of CHKWs occur for the responses from the back-end IMS to the front-end system.

IMS Version 10 increases the range of buffer size specifications. The new MSPLINK BUFSIZE range is from 1 024 to 65 536 and specifies the input and output buffer sizes for each logical link defined for a physical link. The range is standardized and applicable to CTC, MTM, and VTAM links. Prior to IMS Version 10, the size for CTC and MTM links ranged from 160 to 32 689 and from 208 to 30 720 for VTAM links. Note that when bandwidth mode is used, a BUFSIZE of 1 024 might be too small to take advantage of the ability to send multiple messages with one buffer especially if the messages include some of the extended headers that capabilities, such as OTMA, impose. A value of 4 096 is more realistic.

Additionally, in prior releases IMS system definition added 78 bytes to the MSPLINK BUFSIZE for CTC and MTM links, and 288 bytes to VTAM links. IMS initialization added another 28 bytes. This overhead has been removed in IMS Version 10. IMS initializes the MSC buffers to the actual BUFSIZE specified. For IMS Version 10 to be compatible with IMS Version 9 and earlier, however, IMS will continue to add the overhead (that is, 78 + 28 for CTC and MTM links, 288 + 8 for VTAM links) at link restart if the partner IMS is at an earlier level than IMS Version 10 level. This action insures that the buffer sizes are compatible. If the partner is at IMS Version 10 or higher, the overhead is not added.

Also note that in prior IMS releases, VTAM required buffer sizes to reflect a formula of X times 2 to the power of Y , where X had to be a value of 8 through 15 and Y has to be a value from 3 to 13. This restricted the MSC VTAM buffer sizes to certain values that were documented in a VTAM table under the MSPLINK macro in the system definition manuals for previous versions. For IMS Version 10, these restrictions are removed, but remain for compatibility when communicating to IMS Version 9 or an earlier release of IMS. Any BUFSIZE from 1 024 to 65 536 is acceptable when communicating between IMS Version 10 systems.

A label field has been added to the MSLINK macro to allow you to assign a name to the logical link. The name, if you provide it, can be used for the new command capabilities that are provided to affect specific links as we show in Example 4-8. If not specified, IMS will generate the name DFSLxxxx where xxxx is the link number. Link numbers, as in previous releases, continue to be assigned automatically by IMS based on the relative position of the MSLINK macro in the SYSGEN.

Example 4-8 Logical link name

Type 1 command

```
/UPD MSLINK NAME (name) SET((BANDWIDTH(ON | OFF) BUFSIZE(xxxxx))
```

Type 2 command

```
UPDATE MSLINK NAME (name) SET((BANDWIDTH(ON | OFF) BUFSIZE(xxxxx))
```

To support the increased bandwidth enhancements, IMS Version 10 introduces several new command capabilities.

If the characteristics of the link need to be changed, the link must first be stopped and idle. The UPDATE command, when issued, must also be issued on both IMS systems on either side of the link to reflect identical specifications for bandwidth mode and buffer sizes.

The /DISPLAY and QUERY commands show the links with their current buffer size and whether bandwidth Mode is on or off. We show the command syntax in Example 4-9.

Example 4-9 /DIS LINK and QUERY commands

Display Command - Type 1

/DISPLAY LINK number

Existing command that displays link values and BANDWIDTH status if bandwidth mode is set on

/DISPLAY LINK number OPTION BUFSIZE

New format of the command that displays the link parameters such as link#, linkname, bandwidth, bufsize, and status

Query Command - Type 2

QUERY MSLINK NAME(*name*) SHOW(ALL | BUFSIZE | BANDWIDTH)

4.3.4 MSC link statistics

IMS Version 10 also provides a new capability to gather and show statistics on the MSC logical links.

A new log type x'4513' is available to provide information about each logical link. The QUERY command provides options for you to request statistics on the MSC links. This allows for quick and easy access to link performance. The information can be used to determine the efficiency of the link and assist you in deciding on an optimum buffer size. Individual statistics for each MSLINK are collected in the DFSMSCWA work area.

Note that link statistics are not displayed with the SHOW(ALL) keyword. SHOW(STATISTICS) must be used. This keyword also displays the statistics reset mode, RESET,CHKPT or NORESET,CHKPT.

There are three categories of statistics:

- ▶ General, such as statistics start time, ITASK dispatch counts, ITASK processing times, and the rate and number of logger check writes
- ▶ Send, such as messages sent, byte count sent, send message sizes, queue manager get counts and times, and send I/O times
- ▶ Receive, such as messages received, byte count received, receive message sizes, QMGR insert counts and times, and receive I/O times

You can invoke the UPDATE MSLINK command to reset the statistics counters and control how the resetting is to be done as we show in Example 4-10 on page 88.

Example 4-10 Resetting statistics

```
UPD MSLINK NAME(name) START(STATISTICS) OPTION(RESET,CHKPT)
```

```
UPD MSLINK NAME(name) START(STATISTICS) OPTION(NORESET,CHKPT)
```

```
UPD MSLINK NAME(name) START(STATISTICS) OPTION(RESET)
```

The first two forms of the command that control the RESET mode are:

- ▶ RESET,CHKPT mode causes the link statistics to be reset at each IMS checkpoint, after the statistics are logged. The recording interval, therefore, is from IMS checkpoint to checkpoint. This is the default mode and provides a reasonable interval for the statistics to be gathered. A longer interval might not be as useful in determining problems.
- ▶ NORESET,CHKPT mode does not reset the statistics at IMS checkpoint. To be reset, the operator must do so manually. The recording interval, therefore, is from IMS restart or when the last manual reset was issued until the time the command is issued. This mode is useful when running a benchmark or gathering statistics for a longer interval than between IMS checkpoints.

The third form of the command manually resets the link statistics, sets the start time to the current time, and begins a new recording interval.

You can reset link statistics after each IMS checkpoint using the following command. The recording interval is from checkpoint to checkpoint:

```
UPD MSLINK NAME(name) START(STATISTICS) OPTION(RESET,CHKPT)
```

You can avoid resetting the statistics at checkpoint with the following command:

```
UPD MSLINK NAME(name) START(STATISTICS) OPTION(NORESET,CHKPT)
```

The recording interval is from restart or from last manual reset. There is a new option that you can use to reset the statistics and set the start time to the current time:

```
UPD MSLINK NAME(name) START(STATISTICS) OPTION(RESET)
```

We list the general statistics here:

Start_Time	Start time for statistics
Tot_DISP_CT	Number of times link ITASK was dispatched by IMS
Tot_Proc_Time	Total link ITASK processing time for all dispatches
Hi_Proc_Time	Highest link processing time per dispatch
Low_Proc_Time	Lowest link ITASK processing time per dispatch
Avg_Proc_Time	Average link ITASK processing time per all dispatches
chkw_CT	Number of logger check writes
chkwio_CT	Number of logger check writes that caused an I/O
chkw_Rate	Logger check write rate calculated by calculating chkw_ct divided by the statistics recording time

The type of information that you can analyze by reviewing the Send and Receive statistics is shown in Table 4-3 on page 89 and Table 4-4 on page 89.

The Send statistics reference QGET calls. These calls are used to get messages off the queue for send processing.

Table 4-3 Send statistics

Tot_Send_CT	Total # of send I/O
Tot_MSG_Send_CT	Total # of msgs sent
Tot_MsgByte_Send_CT	Total # of msg bytes sent
Tot_Byte_Send_CT	Total bytes of data sent
Hi_MSG_Send_SZ	Largest msg size sent
Low_MSG_Send_SZ	Smallest msg size sent
Avg_MSG_Send_SZ	Average msg size sent
Send_Msg_Time	Interval between 1st & last msg
Tot_Qget_CT	Total # of QMGR gets
Tot_Qget_TM	Total tm to get msg off q
Hi_Qget_Time	Longest QMGR tm to send
Low_Qget_Time	Shortest QMGR tm to send
Avg_Qget_Time	Average QMGR tm to send
Tot_SendIO_Time	Total I/O time to send msgs
Hi_SendIO_Time	Longest send I/O time
Low_SendIO_Time	Shortest send I/O time
SendIO_Req_Rate	Send I/O requests per second
SendIO_Byte_Rate	Rate in bytes/sec sent

The Receive statistics shown in Table 4-4 reference QPUT calls. These calls are used to put messages on the queue as part of receive processing.

Table 4-4 Receive statistics

Tot_Rec_CT	Total # of receive I/O
Tot_Msg_Rec_CT	Total # of messages received
Tot_MsgByte_Rec_CT	Total # of msg bytes received
Tot_Byte_Rec_CT	Total bytes of data received
Hi_Msg_Rec_SZ	Largest msg size received
Low_Msg_Rec_SZ	Smallest msg size received
Avg_Msg_Rec_SZ	Average msg size received
Rec_Msg_Time	Interval between 1st & last msg
Tot_Qput_CT	Total # of QMGR put calls
Tot_Qput_TM	Total proc tm for QMGR puts
Hi_Qput_Time	Longest QMGR tm to rcv
Low_Qput_Time	Shortest QMGR tm to rcv
Avg_Qput_Time	Average QMGR tm to rcv

Tot_RecIO_Time	Total I/O tm to rcv msgs
Hi_RecIO_Time	Longest I/O tm to rcv msgs
Low_RecIO_Time	Shortest tm to rcv msgs
RecIO_Req_Rate	Rcv I/O requests/sec
RecIO_Byte_Rate	Rate in bytes/sec received

When you issue the QUERY MSLINK NAME(*name*) SHOW(STATISTICS) command, all the statistics information on the link is displayed as we illustrate in Example 4-11.

Example 4-11 MSC Statistics Report example - first side

```

IMS1 side
QUERY MSLINK NAME(LNK12V02) SHOW(STATISTICS)

MSLink   MSLink#  MbrName  CC   Option          Start_Time
LNK12V02  10        IMS1     0    NORESET,CHKPT   2007.231 19:03:58.77

Tot_Dispatch CT  Tot_Proc_Time  Hi_Proc_Time  Low_Proc_Time  Avg_Proc_Time  Chkw_CT
98              0.176661      0.059530      0.000003      0.000853      33

ChkwIO_CT  Chkw_Rate  Tot_Send_CT  Tot_Msg_Send_CT  Tot_MsgByte_Send_CT
33          0.165016  26          20              10,461

Tot_Byte_Send_CT  Hi_Msg_Send_SZ  Low_Msg_Send_SZ  Avg_Msg_Send_SZ  Send_Msg_Time
22,403            578            502            523            2.408548

Tot_Qget_CT  Tot_Qget_Time  Hi_Qget_Time  Low_Qget_Time  Avg_Qget_Time  Tot_SendIO_Time
51           0.002054      0.001070      0.000004      0.000040      0.505401

Hi_SendIO_Time  Low_SendIO_Time  SendIO_Req_Rate  SendIO_Byte_Rate  Tot_Rec_CT
0.193435        0.000708        51.444298        44,327            26

Tot_Msg_Rec_CT  Tot_MsgByte_Rec_CT  Tot_Byte_Rec_CT  Hi_Msg_Rec_SZ  Low_Msg_Rec_SZ
20              10,503            22,445          584            502

Avg_Msg_Rec_SZ  Rec_Msg_Time  Tot_Qput_CT  Tot_Qput_Time  Hi_Qput_Time  Low_Qput_Time
525             2.320062      40           0.037326      0.015957      0.000008

Avg_Qput_Time  Tot_RecIO_Time  Hi_RecIO_Time
0.000933      0.070947      0.025376

Low_RecIO_Time  RecIO_Req_Rate  RecIO_Byte_Rate
0.000155      316,363      148,040.09

```

You can issue the command on both sides of the link, as we show in Example 4-12.

Example 4-12 MSC Statistics Report example - second side

```

IMS2 side
QUERY MSLINK NAME(LNK21V02) SHOW(STATISTICS)

MSLink   MSLink#  MbrName  CC   Option          Start_Time
LNK21V02  13        IMS2     0    NORESET,CHKPT   2007.231 19:03:08.42

Tot_Dispatch CT  Tot_Proc_Time  Hi_Proc_Time  Low_Proc_Time  Avg_Proc_Time  Chkw_CT
97              0.150468      0.060122      0.000003      0.000696      33

```

ChkwIO_CT	Chkw_Rate	Tot_Send_CT	Tot_Msg_Send_CT	Tot_MsgByte_Send_CT	
33	0.544554	26	20	10,503	
Tot_Byte_Send_CT	Hi_Msg_Send_SZ	Low_Msg_Send_SZ	Avg_Msg_Send_SZ	Send_Msg_Time	
22,445	584	502	525	2.395508	
Tot_Qget_CT	Tot_Qget_Time	Hi_Qget_Time	Low_Qget_Time	Avg_Qget_Time	Tot_SendIO_Time
51	0.003884	0.001169	0.000004	0.000076	1.304330
Hi_SendIO_Time	Low_SendIO_Time	SendIO_Req_Rate	SendIO_Byte_Rate	Tot_Rec_CT	
0.217838	0.000821	19.933605	17,208	26	
Tot_Msg_Rec_CT	Tot_MsgByte_Rec_CT	Tot_Byte_Rec_CT	Hi_Msg_Rec_SZ	Low_Msg_Rec_SZ	
20	10,461	22,403	578	502	
Avg_Msg_Rec_SZ	Rec_Msg_Time	Tot_Qput_CT	Tot_Qput_Time	Hi_Qput_Time	Low_Qput_Time
523	2.344878	40	0.054777	0.023559	0.000007
Avg_Qput_Time	Tot_RecIO_Time	Hi_RecIO_Time			
0.001369	0.210252	0.036846			
Low_RecIO_Time	RecIO_Req_Rate	RecIO_Byte_Rate			
0.000155	123.661130	106,553			

We list here some suggestions of how you can use the statistics information:

- ▶ Create a benchmark of normal or high message activity by gathering the statistics and keep them for comparison in case a performance problem arises. This base will also be good when you are looking for potential capacity limitations or bottlenecks and when you are determining when more logical links or network capacity might be needed.
- ▶ Estimate the base profile for a unit of work, which in this case, is a send/receive flow. You can do this in a controlled environment where the message send and receive pairs can easily be detected. You need to gather the statistics for consecutive messages and subtract the like statistics (for example, time) to determine the delta, which can then be used to establish a profile for a unit of work.
- ▶ Determine what the message sizes are for the MSC link and use the information to adjust the link and message queue buffer sizes. The goal is for most messages to be processed with a single I/O operation and two QGET (1 GU and 1 DEQ) calls.
- ▶ Determine the efficiency of bandwidth mode if it is enabled, that is, multiple messages with a single send or receive. If bandwidth mode is actually occurring, the total of sends and receives will be less than the Total_message_send and Total-message_receive counts. Although effective for processing efficiency, the value might also indicate that more logical link capacity is needed.
- ▶ Detect problems such as message backups on a link. If a message backup occurs, unusually high values will be seen on the statistics, such as Hi_Proc_Time, Hi_MSG_Send_SZ, Hi_QGET_Time, Hi_SendIO_Time, Hi_MSG_Rec_SZ, Hi_QPUT_Time, and Hi_SRECIO_Time. If high values are observed, then you can take action. A Hi_SendIO_Time, for example, might indicate that an outage in the network connecting the two IMS systems has occurred.

Detailed documentation about the new MSC link statistics capability is provided in several manuals: *IMS Version 10: Command Reference, Volume 1*, SC18-9700, *IMS Version 10: Communications and Connections Guide*, SC18-9703, *IMS Version 10: Diagnosis Reference*, GC18-9707, and *IMS Version 10: Operations and Automation Guide*, SC18-9716.

4.3.5 VTAM Generic Resources support on MSC links

MSC support for VTAM Generic Resources (VGR) is added in IMS Version 10 to provide a single MSC system image for the IMS instances in an IMSplex. That is, remote MSC links outside an IMSplex view the IMSplex with a generic APPLID name, instead of a specific IMS APPLID name, by specifying the generic APPLID name on the MSPLINK NAME parameter. IMS already supports VGR for other terminal types, including ISC, but this function adds the MSC environment as a local mode VGR with IMS-managed affinities. An *affinity* is a mapping of a node to an IMS system and, in the case of MSC, all the parallel sessions associated with a link. MSC VGR support provides the ability to grow the capacity and capabilities of the IMS MSC environment.

Coupling between two (often remote) IMS systems can be through multiple parallel VTAM links as shown in Figure 4-4. The number of links determines the capacity between the IMSs and enhances availability across the links.

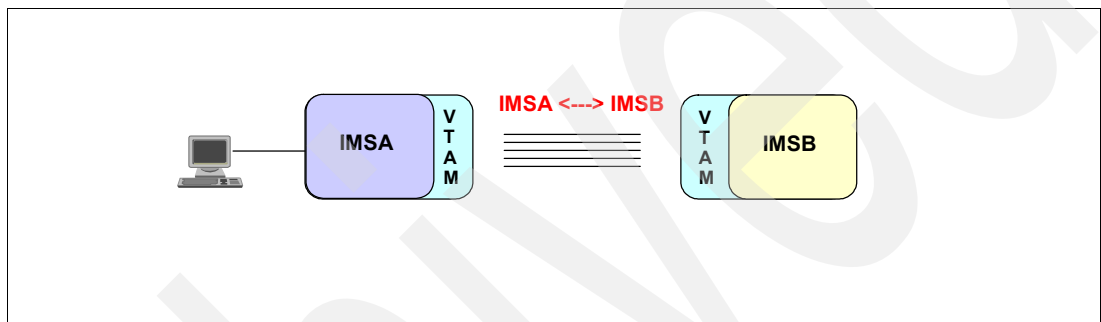


Figure 4-4 Multiple Systems Coupling between two IMSs

When one of the IMSs is turned into a Block Level Data Sharing group, the MSC links can be replicated between the group and the remote IMS or the MSC links within the data sharing group can route messages around the group. Both capabilities are illustrated in Figure 4-5.

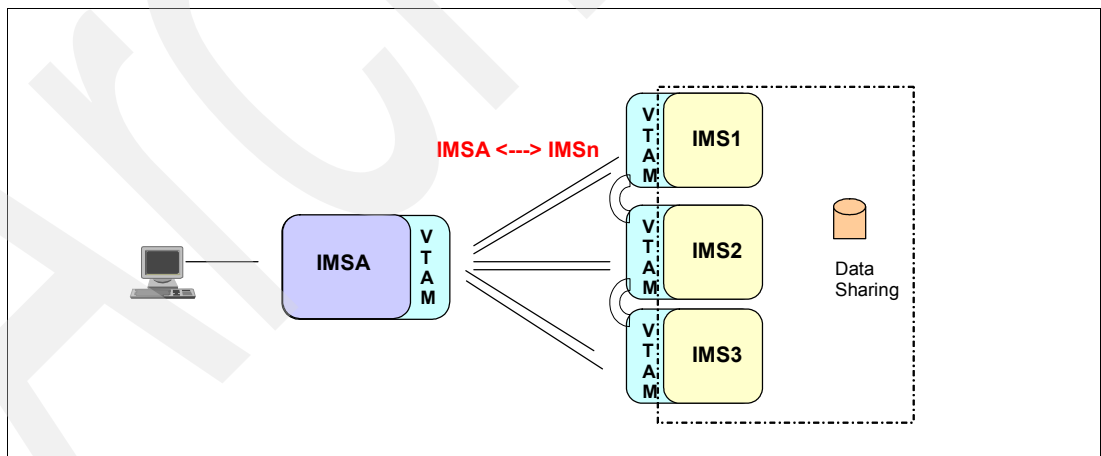


Figure 4-5 Multiple Systems Coupling between two IMS groups

After IMS Shared Queues is implemented, the MSC links between the IMSs in the SQ group are subsumed by SQ even though the definitions might be retained for compatibility. MSC links to remote IMSs can be defined for each member of the IMS SQ group, but only the first set to be started is connected to the remote IMS, as shown in Figure 4-6 on page 93.

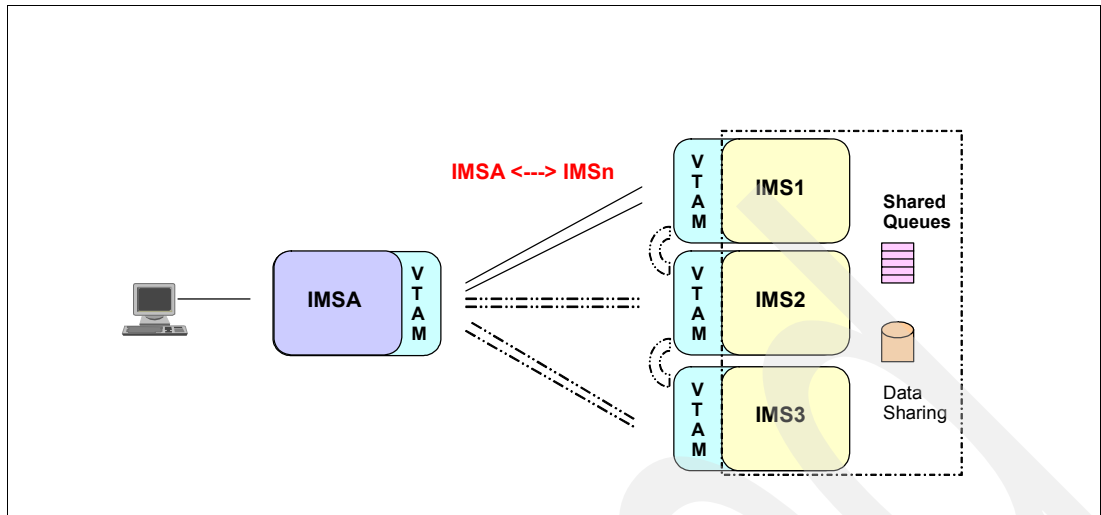


Figure 4-6 Multiple Systems Coupling with two IMS groups and Shared Queues

The use of MSC VGR eases the requirements on the definitions that are needed. A remote IMS can use a single set of MSC link definitions to access any of the IMS systems in the VGR group. Likewise, IMS systems that are part of the local VGR group can clone definitions for MSC access to the single remote IMS. Note that when VGR is used, all parallel sessions associated with the link are routed to the same IMS as shown in Figure 4-7.

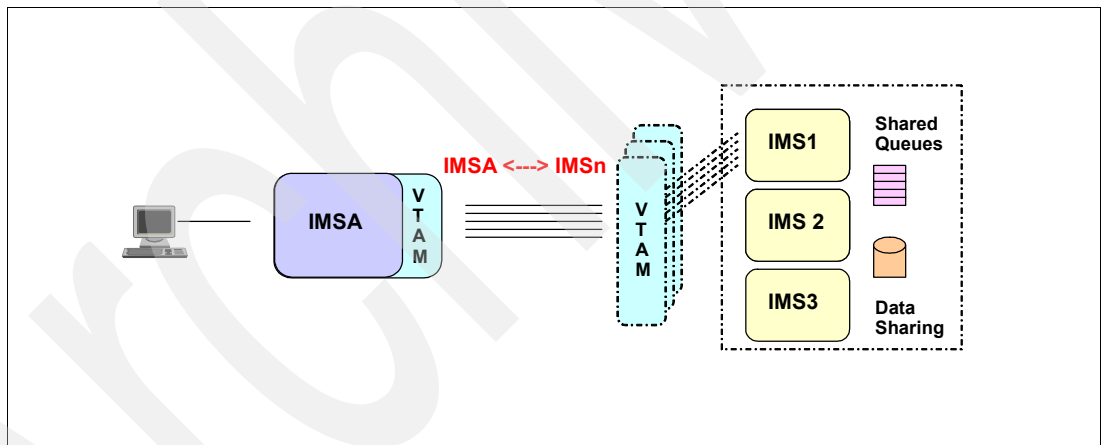


Figure 4-7 MSC between two IMS groups with Shared Queues and VGR

MSC VGR support uses IMS-managed affinity, which causes IMS to delete the affinity when the link session is terminated. This allows you to restart the link on the same or different IMSs. MSC commands are available to terminate and restart the MSC links. The information that MSC keeps about its sessions includes the link status, active or stopped, and the sequence numbers. This is a local mode only characteristic, that is, no link status information is stored in the Coupling Facility or made available to the link sessions when they are restarted on another IMS in the IMSplex.

Local mode restricts usage to planned restarts, such as normal or cold IMS restarts, or IMS ERE restarts where the link status is reset to COLDSESS by either the /CHANGE LINK or UPDATE MSLINK commands. If you plan an outage, you must stop the parallel sessions on both sides of the link and then restart using the new link partner. To allow the information about sequence numbers to be cleared, the sessions need to be in cold or NRE status on both sides of the link.

Alternatively, if an IMS failure occurs where one IMS system terminates, you must stop the parallel sessions on the surviving IMS. Note that if you change the session status from ERE to cold mode, messages can be duplicated or lost.

If you need to reestablish the links between the same two instances of IMS, then a way for you to accomplish this is to:

- ▶ Start the links on the instance of the IMS in the VGR group where the connection is desired. This action establishes affinity for all the parallel sessions, which remains in effect through subsequent restarts and until the last parallel session is terminated.
- ▶ After the affinities are established, you can start the links on the remote IMS system.

4.3.6 Migration considerations for MSC

When migrating to IMS Version 10, ensure that your older routing exit routines, if any, have been converted to DFSMSCE0. This migration can be accomplished in any of the supported previous releases prior to the actual IMS Version 10 migration. If the exit routines still exist while the IMS Version 10 migration is in progress, IMS Version 10 migration tasks need to include the upgrade of the previous releases to DFSMSCE0.

A like-for-like migration from a previous release to IMS Version 10 allows MSC to be initialized in non-bandwidth mode. This mode is compatible with MSC operations in previous releases.

The ability to turn on bandwidth mode is provided via the UPD command and requires both partners of the link to be at an IMS Version 10 level. In this mode, the minimum size of the buffers must be 4 096. To be more effective, the buffer sizes must be defined to accommodate multiple messages.

The bandwidth mode can be set ON/OFF by the IMS UPDATE command on a link by link basis for all the CTC, MTM, or VTAM links.

The specified mode must be the same on both sides of the link, or the link restart will be rejected with a DFS3218 message. IMS Version 10 MSC links that are connected to IMS Version 8 or IMS Version 9 are not be able to set bandwidth mode on the IMS Version 8 and IMS Version 9 side of the link, so the link, in effect, will not be able to operate in bandwidth mode.

4.4 Sysplex serialized program management

Over the years, IMS has been designed to be the best high volume transaction processing system. The design is based on running many independent programs and transactions simultaneously to get this optimum performance. The intent is that each transaction is a distinct unit of work, with no interdependencies on other transactions; therefore, IMS can use its enhanced scheduling techniques to run transactions wherever resources are available.

However, applications and their transactions are not always completely independent. There can be application reasons why certain transactions must run in a certain sequence or order rather than randomly. IMS has provided two techniques to help handle these situations, at the program level and at the transaction level. However, these techniques provide serialization within a single IMS system; they are not designed to manage serialization across multiple IMS systems, such as in a Parallel Sysplex Shared Queues environment.

The first technique you can employ for handling serialization of programs is implemented via the SCHDTYP=SERIAL parameter on the APPLCTN macro where you define a particular program. By specifying SCHDTYP=SERIAL, you tell IMS that this application program cannot be scheduled into more than one IMS dependent region (MPR, BMP, or JMP) within a single IMS system; this also applies to CICS and ODBA users.

The second technique that you can use for handling serialization of transactions is implemented via the SERIAL=YES parameter on the TRANSACT macro where you define the transactions that are associated with a particular program (APPLCTN macro). In most cases, IMS will process transactions in their arrival order; however, if an unusual condition occurs, such as a U3303 abend, the queue might not be maintained in arrival order. Using SERIAL=YES on the TRANSACT macro will guarantee that transactions getting U3303 abends will be placed at the beginning of the queue to maintain that arrival order. Again, this technique only works within a single IMS system.

Because these serialization techniques only apply within a single IMS system, if you have transactions that need serialization within an IMSplex rather than just in a single IMS system, you have to set up special operational procedures to handle these cases.

IMS Version 10 enhances the support for serialization of programs that are defined as SCHDTYP=SERIAL across all the members of an IMSplex, rather than just within a single IMS when using Shared Queues.

Now, IMS Version 10 will ensure that your SCHDTYP=SERIAL programs will be scheduled in only one IMS dependent region at a time across an entire IMSplex with Shared Queues. CICS and ODBA are not supported for Sysplex serialized program management (SSPM). The dependent region types are:

- ▶ MPR
- ▶ JMP
- ▶ IFP
- ▶ Message-driven BMP
- ▶ Non-message driven BMP
- ▶ JBP

Note: SCHDTYP=SERIAL is the default if not specified on the APPLCTN macro. When you are using DRD, SCHDTYPE(PARALLEL) is the default.

IMS uses the IMS Resource Manager (RM) component of the IMS Common Services Layer (CSL) to provide the serialization. IMS Version 10 determines if a SCHDTYP=SERIAL program is already scheduled by interacting with the IMS Resource Manager and its resource structure. IMS uses RM to attempt to create the program resource in the resource structure. If the program resource does not exist in the resource structure, then no other copy of this program is currently scheduled; therefore, this scheduling can continue. A program record for this program will be created in the resource structure. For MPRs, JMPs, and IFPs, the program resource remains active in the resource structure until the PSB is unscheduled. For BMPs and JBPs, the program resource will be deleted when the batch message processing program terminates.

If the serial program is already scheduled on another IMS, the program resource already exists in RM and the attempt to create the program resource in RM fails; therefore, this attempt to schedule what would be the second copy of the serial program will fail. For MPRs and JMPs, the dependent region will be available to schedule the next eligible program. The message that got the scheduling failure will remain on the Shared Queues (was never taken off) and will hopefully be picked up by the currently scheduled program. BMPs, JBPs, and

IFPs will get a U0457 abend when scheduling finds that the program is already scheduled, as they did before in previous versions within a single IMS.

The RM program record is released at normal termination of the program/PSB; abnormal termination will be handled by backout. There is then a notification to retry scheduling if the queue has gone from empty to non-empty in a Shared Queues IMSplex. This is achieved by sending an SCI notification to all IMSs when the serial program is terminated. Naturally, if there are serial programs with multiple transactions defined, scheduling will lock out other transactions for that program

Because SSPM applies only to IMS Version 10 subsystems and is not enforced for pre-IMS Version 10 subsystems, you must continue with any current procedures to prevent parallel scheduling in these subsystems.

SSPM requires an IMSplex environment with Shared Queues; that is, you must supply CQS and System Logger address spaces and structures, Shared Queues structures, and a Common Service Layer consisting of SCI, OM, and RM address spaces together with the resource structure. You do not need to make any program changes or definition changes.

4.5 Transaction level statistics enhancement

IMS Version 10 provides enhanced performance and accounting information in its log records. This can provide you with more granular information for accounting and charge-out purposes. It also provides more detailed information for understanding the performance of individual transactions and programs. Some of the information in the log records was previously available only when the IMS Monitor trace was on for IMS TM environments. You then processed the trace data with the IMS Monitor report program or the IMS Performance Analysis tool. IMS Version 10 provides new options for obtaining this data.

Enhanced statistics are written in several IMS log records: the application start accounting log record (x'08'), the application termination accounting log record (x'07'), and the CPI-C driven transaction termination log record (x'0A07'). There is a new log record that records statistics for individual transactions, not just for executions of programs or schedulings of PSBs. The enhancements provide:

- ▶ Improved performance data by transaction or commit
- ▶ Improved accounting data by transaction or commit
- ▶ Additional accounting and performance data for IMS TM, whereas some data was only available with DBCTL in previous releases

Transaction level statistics are available in a new optional log record, the x'56FA' log record. When you choose it, it is written for each commit for transactions defined as MODE=SNGL and for non-message driven BMPs and JBPs. For transactions defined as MODE=MULT, the log record is written for each message. Most of the fields in the x'56FA' log record are like those in the x'07' log record except they are for individual transactions, not the execution of the entire program or scheduling of a PSB.

The additional fields in the x'56FA' log record include read counts and write counts for both VSAM and OSAM database data sets. There are counts of External Subsystem Attach Facility calls. There are typically calls to DB2, but they also might be for MQ or another subsystem. The CPU time for the transaction is written in time of day (TOD) format. The statistics include the total elapsed time for database I/Os and the total elapsed time for waits for locks. These two elapsed times were previously available in the x'07' log records for

DBCTL threads. They were not available for IMS TM except in IMS Monitor trace records. The x'56FA' log record is mapped by the DFSETPCP macro.

The transaction level statistics log records are optional. You specify TRANSTAT in the diagnostic statistics section of the DFSDFxxx PROCLIB member as illustrated in Example 4-13 to determine the default setting for all transactions and non-message driven programs. TRANSTAT=N is the default whereas TRANSTAT=Y causes the log records to be written. The setting in this member can be overridden for individual transactions and non-message driven programs by using commands. You can also request TRANSTAT by specifying it in the TRANSACT and APPLCTN macros.

Example 4-13 Setting transaction statistics request

```
<SECTION=DIAGNOSTIC_STATISTICS>
TRANSTAT=Y|N
```

There are three type-2 commands that you can use to display and modify transaction level statistics for transactions and non-message driven programs:

- ▶ The QUERY command to show the current setting for a transaction (TRAN), transaction descriptor (TRANDESC), program (PGM), or program descriptor (PGMDESC). SHOW(TRANSTAT) causes the setting to be returned by the command. PGM applies only to non-message driven programs, ODBA, and DRA (CICS). The syntax and an example are shown in Example 4-14.

Example 4-14 QUERY syntax and example

```
QUERY TRAN|TRANDESC|PGM|PGMDESC SHOW(TRANSTAT)

QRY TRAN NAME(ABC) SHOW(TRANSTAT)
```

- ▶ The UPDATE command, which we illustrate in Example 4-15, changes the current setting of transaction level statistics for any of the entities.

Example 4-15 UPDATE syntax and example

```
UPDATE TRAN|TRANDESC|PGM|PGMDESC SET(TRANSTAT(Y|N))

UPD TRAN NAME(ABC) SET(TRANSTAT(Y))
```

- ▶ The CREATE command that we show in Example 4-16 includes the SET(TRANSTAT(Y|N)) parameter for you to set the transaction level statistics on or off when creating a new transaction, transaction descriptor, program, or program descriptor. It defaults to the setting in the descriptor or the transaction or program specified in the LIKE parameter.

Example 4-16 CREATE syntax and example

```
CREATE TRAN|TRANDESC|PGM|PGMDESC SET(TRANSTAT(Y|N))

CRE TRAN NAME(ABX) LIKE(RSC(ABC)) SET(TRANSTAT(N))
```

When you implement IMS Version 10, you choose whether you want transaction level statistics. If you want transaction level statistics for most of your transactions and programs, you must specify TRANSTAT=Y in the DFSDFxxx member of PROCLIB. If you do not want transaction level statistics for most of your transactions and programs, you must specify TRANSTAT=N. Use the UPD command for exception transactions and programs, or you can request TRANSTAT on individual APPLCTN and TRANSACT macros in the system definition. If you use Dynamic Resource Definition (DRD), you can also create transaction

descriptors and program descriptors with default settings for TRANSTAT so that newly created transactions and programs using these descriptors will default to the setting that you want.

The x'56FA' log record is approximately 528 bytes and is written for every transaction and for every commit for non-message driven programs for those transactions and programs with the option. This logging can be voluminous so you must plan for the increased logging before using this option in general.

Application start accounting log record (x'08')

The statistics in the application start accounting log record (x'08'), written for each scheduling of a program, have been enhanced. Statistics about the scheduling of the program were written for DBCTL threads in previous releases. IMS Version 10 writes these statistics for all programs, not just DBCTL threads:

- ▶ LINTMINT - Wait time for intent conflict
- ▶ LINTMPOL - Wait time for pool space
- ▶ LINTMSCH - Elapsed time of scheduling process

These statistics show the time that scheduling waited due to intent conflicts, such as conflicts that can occur when a PROCOPT includes E, the time that scheduling waited due to a lack of pool space, such as in the PSB pool, and the elapsed time of the scheduling process. Two new fields have been added. They are LINTPGM, which contains the program name, and LINTPSB, which contains the PSB name. The program name and PSB name were previously available in the LINTSY1 and LINTSY2 fields; however, there were complicated rules governing the contents of these two fields. LINTPGM and LINTPSB greatly simplify the discovery of the program name and PSB name. LINTSY1 and LINTSY2 remain in IMS Version 10 and contain the same contents as in previous releases. The log record is mapped by the DFSLOG08 macro.

Application termination accounting log record (x'07')

The application termination accounting log record (x'07') mapped by the DFSLOG07 macro has been enhanced. It has always contained many fields with counts of DLI calls and other statistics for the execution of a program or PSB.

THE DLRFLAG2 byte has a new bit definition. The DLRFALSE bit indicates that the scheduling of the transaction was false. This can occur in a Shared Queues environment when multiple IMS systems react to the arrival of a transaction message on the queue by scheduling the same transaction. If a system attempts to retrieve a message but it has already been retrieved by another system, the system that processes the no input message has gone through a false schedule.

The DLREXTIM field has been added. It contains the CPU time for the program in time of day (TOD) format. Previously, this time was only available in timer units in the DLRTIME field in the log record. DLRTIME also depended on the setting of the STIMER parameter. DLREXTIM does not depend on the STIMER setting.

The DLRTMEIO (total DB I/O time) and DLRTMEPL (total lock wait time) fields are now populated in all environments. Previously, only DBCTL threads caused these fields to be populated. DLRTMEIO contains the total I/O time for database calls. DLRTMEPL contains the total wait time for database locks.

Fields have been added. These include counts for VSAM and OSAM database data set reads and writes, ESAF calls, such as those to DB2, and additional Fast Path call counts for FLD calls and POS calls. When an abend occurs, the abend reason code is included when available.

The DLI call counts for GU and GN. ISRT, and so on now include both full function and Fast Path database calls.

CPI-C driven transaction termination log record x'0A07'

The x'0A07' log record is written when a CPI-C driven transaction terminates and includes accounting data similar to that in the x'07' log records. IMS Version 10 has added the LOAEXT field, which contains the CPU time in time of day (TOD) format, to the log record. Previously, this time was only available in timer units in the LOAETIM field in the log record. The log record is mapped by the DFSLOG0A macro.

4.6 Fast Path enhancements

This section discusses the following topics:

- ▶ Command enhancements, which provide you an option to start all areas with UPD DB ... START(ACCESS), an option to keep the DEDB randomizer in memory when stopping access to a database and QUERY DB output with different status for /DBR as compared to /STOP.
- ▶ Shared VSO Lookaside specification on the private buffer pool statement for shared VSO is not required, and compression of shared VSO private pool by XRF is described in Chapter 2, "System enhancements" on page 13.
- ▶ Capacity enhancements
- ▶ An EMH enhancement to allow you to reset Fast Path response mode

Starting all areas with UPDATE DB command

This enhancement allows you to start all of the areas in a database when you start the database through OM. The only valid specification that you can give for the AREA parameter is AREA(*). If an area cannot be started, a separate line is provided for it in the response as we show in Example 4-17 where area ARPS64 was not started.

Example 4-17 TSO SPOC UPDATE DB command and response

UPDATE DB NAME(DEDBPS) START(ACCESS) AREA(*)

DBName	AreaName	MbrName	CC
DEDBPS	*	IMSA	8
DEDBPS	ARPS64	IMSA	A5

The 8 completion code for the database indicates that the command completed with errors for one or more areas of the database. The A5 completion code for area ARPS64 indicates that the DBRC "Prohibit Further Authorization" flag was on for this area.

Keeping the randomizer resident when stopping a DB

In previous releases, if you made a DEDB inaccessible, its randomizer was unloaded from memory if the randomizer was not being used by any other DEDB. You make DEDBs inaccessible with either a /DBR DB command or an UPDATE DB STOP(ACCESS) command.

IMS Version 10 adds an OPTION(NORAND) parameter to the UPDATE DB STOP(ACCESS) command as follows:

```
UPDATE DB NAME(DEBPS) STOP(ACCESS) OPTION(NORAND)
```

This causes IMS to keep the randomizer module in memory, which helps to avoid the fragmentation of ECSA that can occur when modules are unloaded and reloaded. The /DBR DB command does not have a parameter that is equivalent to OPTION(NORAND) on the UPD DB STOP(ACCESS) command.

A new status is available with the QUERY DB command in conjunction with the OPTION(NORAND) parameter on the UPDATE DB STOP(ACCESS) command. In previous releases, you could tell that a database was inaccessible, because the RNL status, which indicates that the randomizer module is not loaded, was associated with the database. The absence of the RNL status does not indicate whether the database is accessible. The new status, STOACC status, in the response to a QUERY DB command, indicates that the database is inaccessible as we show in Example 4-18. That is, the database has been /DBRed or an equivalent UPDATE DB STOP(ACCESS) command has been processed for the database.

Example 4-18 Stop database access and unload randomizer (equivalent to /DBR DB)

```
UPDATE DB NAME(DEBPS) STOP(ACCESS)
```

DBName	AreaName	MbrName	CC
DEDBPS	*	IMS1	0

```
QRY DB NAME(DEBPS) SHOW(STATUS)
```

DBName	AreaName	MbrName	CC	TYPE	LclStat
DEDBPS		IMS1	0	DEDB	NOTOPEN,STOACC,RNL
DEDBPS	ARPS10	IMS1	0	AREA	NOTOPEN,STOPPED
DEDBPS	ARPS20	IMS1	0	AREA	NOTOPEN,STOPPED

The RNL status is now used only to indicate that the randomizer module is not loaded, whereas the STOACC status is used to indicate that the database is inaccessible. The STOACC status is not returned by the /DISPLAY DB command, but is only returned by QUERY DB commands as we show in Example 4-19. The response to the QUERY command after the UPDATE DB STOP(ACCESS) OPTION(NORAND) command has "STOACC" in the status field for the database. This indicates that the database is not accessible. The response does not include RNL in the status field, which indicates that the randomizer is still loaded, even though the database is not accessible.

Example 4-19 Stop access to database and randomizer not unloaded

```
UPDATE DB NAME(DEBPS) STOP(ACCESS) OPTION(NORAND)
```

DBName	AreaName	MbrName	CC
DEDBPS	*	IMS1	0

```
QRY DB NAME(DEBPS) SHOW(STATUS)
```

DBName	AreaName	MbrName	CC	TYPE	LclStat
DEDBPS		IMS1	0	DEDB	NOTOPEN,STOACC
DEDBPS	ARPS10	IMS1	0	AREA	NOTOPEN,STOPPED
DEDBPS	ARPS20	IMS1	0	AREA	NOTOPEN,STOPPED

Shared VSO enhancements

If you had single area structures specified for shared VSO in previous releases, you needed to specify in two places whether lookaside buffering was used for an area. You specified it in

DBRC with an INIT.DBDS or CHANGE.DBDS command and also in the DEDB statement in the DFSVSMxx member as part of the private pool definition. If the specifications conflicted, the DBRC value was used; only the DBRC value mattered. Nevertheless, some value had to be specified in the DEDB statement. The requirement to specify a lookaside value in the DEDB statement has been eliminated in IMS Version 10. You do not have to specify any value there.

The specification of lookaside for multiple area structures, which were introduced in IMS Version 9, has not changed in IMS Version 10. IMS continues to ignore any lookaside specification in DBRC for areas using multiple area structures. Only the specification in the DEDBMAS statement of the DFSVSMxx member is used.

Capacity enhancements

The Fast Path buffer pool DBBF has had the limit on the number of buffers raised from 65 535 to a theoretical value of 4 294 967 295. This is theoretical, because a system is unlikely to have the storage for so many buffers. Any value up to the theoretical limit can be specified on the DBBF= execution parameter for the online system. Previously, you could specify the number of Fast Path buffers on the FPCTRL macro, but this macro is not used in IMS Version 10. The DBBF= execution parameter must be used to specify these buffers in IMS Version 10.

Certain log records have been modified to accommodate the new maximum value; IMS now maintains the number of buffers in full word (4 bytes). The log records that have been modified include the following that are mapped by the DBFLSRT macro: 5950, 5951, 5952, 5953, 5956, and 5957. Other modified records are mapped by the DBFLGRSD macro. They include 5955 and 5958.

The maximum number of Fast Path output threads in previous releases was the smaller of the MAXPST value or 255. In IMS Version 10, the maximum is 32 767. This allows you to have many more output threads. This can be desirable as processor capacities increase.

A single module, DBFCONT0, in previous releases contained most Fast Path control blocks and buffers. The module resides in ECSA and it can be very large. IMS Version 10 places these control blocks and buffers in five modules to allow you to use ECSA storage more efficiently, because smaller contiguous areas of ECSA can be used.

The major new modules and their contents are:

- ▶ DBFCONT1: ECNTs, MSDBs, and MSDB blocks
- ▶ DBFCONT3: buffer headers (DHMRs) and buffers
- ▶ DBFCONT4: DEDB blocks
- ▶ DBFCONT5: output threads
- ▶ DBFCONT6: BALGs (balancing groups)

FDBR FPBUFF= parameter default

APARs PQ97745 for IMS Version 9 and PQ80264 and PQ97043 for IMS Version 8 added a new parameter in the DFSFDRxx PROCLIB member for FDBR. FPBUFF=LOCAL could be specified to place module DFSCONT0 and HSSP and HSRE buffers in extended private. There was no other valid value for FPBUFF=. If the parameter was not specified, the module and buffers were placed in ECSA.

As described in section “Fast Path buffer pool” on page 62, IMS Version 10 changes the default. If the FPBUFF= parameter is not specified, DBFCONT1, DBFCONT3, DBFCONT4, DBFCONT5, DBFCONT6, and HSSP and HSRE buffers are placed in extended private for FDBR. A specification of FPBUFF=ECSA places these modules and buffers in ECSA. We

recommend a specification of FPBUFF= LOCAL and allowing the parameter to default. There is no advantage in placing the modules and buffers in ECSA.

EMH enhancement to reset Fast Path response mode

This EMH enhancement addresses a rarely occurring situation; however, when it does occur, you were likely to need an IMS outage to correct the problem in previous releases.

As a full function user, you have been able to reset response mode for hung terminals via the /RST, /STO, and /STA commands; this enhancement extends the support to Fast Path users.

Note: You cannot use the /RST command to reset response mode for Fast Path nodes/users as you can for full function users, because Fast Path requires a session to be in a quiesced state and /RST does not have this requirement.

You can issue the following commands at any time to reset the response mode:

- ▶ /STO NODE and /STA NODE for static nodes
- ▶ /STO USER and /STA USER for dynamic users

The enhancement applies to input response mode with local/shared EMH and output response mode with local/shared EMH. The reset is performed during /STA processing.

The DISPLAY command output has been enhanced with a new status, RESP-INP-FP, to show Fast Path input response mode, for the commands that we show in Example 4-20. Hung users are typically in “input” response mode, not “output” response mode.

Example 4-20 RESP-INP-FP status added to /DIS command output

```
/DIS NODE, /DIS NODE RESPINP  
/DIS USER, /DIS USER RESPINP  
/DIS STATUS, /DIS STATUS NODE, /DIS STATUS USER
```

The status RESP-INP-FP (Fast Path input response mode) means that a terminal or user has entered an FP transaction and that transaction has not yet processed. It might be something that will resolve itself quickly so the wait would be short. Or it might be something where the wait can be very long; there is no guarantee that an output will ever come back for this input transaction. This is the case where your use of this new command support will help.

When you see the status RESP for Fast Path output response mode, it means that the terminal is in response mode waiting for available output to be delivered. In most cases, seeing an FP terminal/user hung in output response mode will be a transient event that is resolved through normal procedures. The user will eventually get their output. For example, if the session is lost before the output is sent, you just press PA2 on a 3270 to get that message after you have reestablished your session. So having to reset FP output response mode is a rare occurrence. However, if you reset FP output response mode, the output message will be deleted.

The status RESP for full function response mode means that the terminal is in response mode waiting for available output to be delivered. For full function, if you reset output response mode, the output message stays on the LTERM queue.

To determine which of your Fast Path static nodes or dynamic users are hung and need to be reset, you can use one of the forms of the /DIS command that we show in Example 4-20 and look for nodes/users that have the status of RESP-INP-FP, Fast Path input response mode. Hung users will typically be hung on input response mode, not output response mode.

The keyword RESPINP on the /DIS command will list both Fast Path input response mode (RESP-INP-FP) and full function input response mode (RESP-INP).

You use the /STOp USER for dynamic terminals and /STOp NODE for statically defined terminals and the appropriate /STArT USER or NODE to reset the response mode. The detailed actions taken by IMS depend on the circumstances:

- ▶ Local EMH message with output not available
 - Write a X'67D0' subtype 11 trace record
 - Reset Fast Path response mode
 - Discard output when it becomes available
 - Write a X'5936' log record (FLDQ1STA)
- ▶ Local EMH message with output available
 - Write a X'5936' log record (FLDQ1STA)
 - Write a X'67D0' subtype 11 trace record
 - Discard output
 - Reset Fast Path response mode
- ▶ Shared EMH message with output not available
 - Write a X'5936' log record (FLDQ1STA)
 - Write a X'67D0' subtype 11 trace record
 - Reset Fast Path response mode
 - Discard output when it becomes available as an unsolicited message
 - Write a X'67D0' subtype 11 trace record
- ▶ Shared EMH message with output available
 - Write a X'5936' log record (FLDQ1STA)
 - Write a X'67D0' subtype 11 trace record
 - Discard output
 - Reset Fast Path response mode
 - Write a X'67D0' subtype 11 trace record

For operational purposes, you will see “Input” response mode now indicated by two states:

- ▶ RESP-INP-FP for Fast Path
- ▶ RESP-INP for Full Function

If you have automation dependent on the RESP-INP status and have Fast Path EMH, your automation needs to be changed to handle both input response mode states.

Archived



Manageability enhancements

With IMS Version 10, manageability continues to evolve and be enhanced. New enhancements for IMS TM and IMS DB that focus on both operations management and systems management enable you to manage your IMS environments more effectively. New installation options and administration features help you easily administer, deploy, and manage operations from a single point of control to improve visibility and reduce costs.

5.1 Operations Manager enhancements

Operations Manager, a major component of the IMS Common Service Layer (CSL), controls the operations of an IMSplex. IMS Version 10 has two enhancements for Operations Manager that provide the auditing capabilities needed for your IMSplex environment, as well as improved access to all messages processed by OM.

5.1.1 OM audit trail

Operations Manager provides an audit trail for IMS commands that are processed through the OM interface. This audit trail includes command input, command responses, and unsolicited output messages. This audit trail can be used as an information source for all commands and responses going through Operations Manager, addressing client requirements for auditing capabilities in the OM environment. This enhancement makes the OM environment a more comprehensive solution for operations management with improved output and auditability for commands entered through the SPOC.

OM uses the z/OS System Logger to log commands, responses, and unsolicited output to an audit trail log stream. The name of this audit trail log stream is specified in the CSLOIxxx OM initialization PROCLIB member in the AUDITLOG sub-parameter of the IMSPLEX parameter as follows:

```
IMSPLEX(NAME=plexname,AUDITLOG=log stream name)
```

This *log stream name* must match the NAME parameter specified in the z/OS LOGR policy on the DEFINE LOGSTREAM statement. You can configure the OM audit trail so that each OM uses its own log, or all OMs use the same log for all SPOC activity. However, in an IMSplex environment, we recommend that no more than two OMs share the same audit trail log data stream because every OM in the IMSplex receives a copy of all unsolicited output messages. For example, if you have four OMs in your IMSplex and all four OMs connect to (or share) the same audit trail log stream, you will have four copies of each unsolicited output message on the audit trail log stream.

The OM audit trail can use either a Coupling Facility (CF) log stream or a DASD-only log stream. In an IMSplex environment with OM, you typically use a CF log stream to facilitate sharing among all OMs. However with a single IMS system in an OM environment, you might want to use a DASD-only log stream.

Example 5-1 shows the z/OS CFRM policy changes needed for the OM audit trail.

Example 5-1 Example of updates needed in the z/OS CFRM policy for the OM audit trail

```
STRUCTURE NAME(MVSLOGOM2Q01)
          SIZE(8192)
          PREFLIST(CF01,CF02)
```

Example 5-2 shows the z/OS LOGR policy changes needed for the OM audit trail.

Example 5-2 Example of updates needed in the z/OS LOGR policy for the OM audit trail

```
DEFINE STRUCTURE NAME(MSVLOGOM2Q01)
          LOGSNUM(1)
          AVGBUFSIZE(4000)
          MAXBUFSIZE(32760)
DEFINE LOGSTREAM NAME(SYSLOG.OMAUDIT.LOG)
```

```
STRUCTNAME(MVSLOGOM2Q01)
LOWOFFLOAD(20)
HIGHOFFLOAD(50)
LS_SIZE(nnn)
```

Example 5-3 shows the parameters in the CSLOIxxx PROCLIB member to be used with the z/OS logger definitions shown in Example 5-1 on page 106 and Example 5-2 on page 106.

Example 5-3 Example of parameters needed in the CSLOIxxx PROCLIB member

```
IMSPLEX(NAME=plexname,AUDITLOG=SYSLOG.OMAUDIT.LOG)
```

Sizing information for the z/OS logger structure for the OM audit trail is available at the CFSIZER (Coupling Facility Sizer Tool) Web site at:

<http://www.ibm.com/servers/eserver/zseries/cfsizer/>

The LS_SIZE parameter in the log stream definition determines the size of the offload DASD log data sets. The z/OS logger supports a maximum of 168 offload DASD log data sets unless you specify data set directory extent records in the LOGR couple data set and make it the active primary LOGR couple data set. IBM recommends that you size the data sets as large as your installation can afford to make them. This will minimize the number of log data sets required to represent a log stream. It will also minimize the number of times that system logger must reallocate and switch to using a new log data set when an old one becomes full. Because allocating and switching to a new log data set incur overhead, do this as little as possible. We recommend that you use SMS to manage these data sets; DFHSM can also be used to migrate these data sets.

Log data in the OM audit trail is not marked for deletion by OM, because each installation will have its own requirements for keeping this data for auditability purposes. You can set up procedures based on the RETPD and AUTODELETE options in the DEFINE LOGSTREAM definition in the z/OS LOGR policy to determine how long you keep data in the log stream.

The z/OS Logger provides several methods for duplexing Coupling Facility log data for recovery purposes. These methods ensure against data loss due to Coupling Facility, system, or System Logger failure. You can use local buffers, DASD staging data sets, or system-managed duplexing. We recommend that you analyze your availability requirements for access to the OM audit trail data and then choose the duplexing method that best addresses your needs.

You can read more information about using the z/OS logger in *z/OS MVS Setting up a Sysplex*, SA22-7625, for the level of z/OS that you use.

Optionally, an OM input exit routine and an OM output exit routine can be used to examine and manipulate message data that is logged to the OM audit trail. Details of these two OM exits can be found in *IMS Version 10: Exit Routine Reference*, SC18-9708.

If an individual command or command response plus its associated control data is longer than 32 760 bytes, it is split across multiple log records. These log records include control data that log formatting utilities can use to reassemble the log records into a single command or command response. We recommend that you set the log stream maximum buffer size (MAXBUFSIZE) to 32 760 bytes. This setting ensures that all command and command response data is written to the audit trail. A lower setting will result in the truncation of some OM audit trail log records.

There are several record types used for logging the OM audit trail messages. For command input, logging is done before an optional OM input exit is called; if the exit modifies this data,

another log record will be written. For command output and for unsolicited output messages, logging is done before an optional OM output exit is called; if the exit modifies this data, another log record will be written. Table 5-1 lists the OM audit trail log record types.

Table 5-1 OM audit trail log record types

Log record type	Subtype	Mapping macro	Conditions for writing log record
06	01	CSLLOGCM	Command input, prior to calling the OM input exit routine
06	02	CSLLOGCM	Command input, after being modified by the OM input exit routine
08	01	CSLLOGCR	Command response output, prior to calling the OM output exit routine
08	02	CSLLOGCR	Command response output, after being modified by the OM output exit routine
09	01	CSLLOGOU	Unsolicited output message to an AOP client before calling the OM output exit routine
09	02	CSLLOGOU	Unsolicited output message to an AOP client after calling the OM output exit routine and OM output exit routine modified the unsolicited output message

You can print or view the OM audit trail in two ways:

- ▶ By using the IMS File Select and Formatting Utility (DFSERA10) with exit routine CSLULALE for formatted output or exit routine CSLOERA3 for dump output
- ▶ By using the TSO SPOC support for viewing and printing the OM audit trail or writing a REXX™ SPOC application for accessing the OM audit trail. Details of using the TSO SPOC to view and print the OM audit trail can be found in topic 5.5.1, “Support for viewing the OM audit trail” on page 120. Details for using a REXX SPOC application for accessing the OM audit trail can be found in topic 5.6.1, “Support for OM unsolicited output messages” on page 128.

The following examples show the usage of DFSERA10 to print the OM audit trail with either formatted or dump output.

Example 5-4 on page 109 shows sample JCL for using DFSERA10 to print the OM audit trail. The EXITR= parameter on the OPTION PRINT statement determines whether formatted output will be produced (exit CSLULALE) or dump output will be produced (exit CSLOERA3).

Example 5-4 Sample JCL using DFSERA10 to print the OM audit trail

```
//CSLERA10      JOB      MSGLEVEL=1,MSGCLASS=A,CLASS=K
//STEP1        EXEC     PGM=DFSERA10
//STEPLIB      DD       DSN=IMS.SDFSRESL,DISP=SHR
//SYSPRINT     DD       SYSOUT=A
//SYSUT1       DD       DSN=SYSLOG.OMAUDIT.LOG,
//              SUBSYS=(LOGR,IXGSEXIT),
//              DCB=(BLKSIZE=32760)
//SYSIN        DD       *
CONTROL        CNTL     H=EOF
OPTION         PRINT    EXTR=CSLULALE | CSLOERA3
END
//
```

Example 5-5 shows output produced with the formatted option for printing the OM audit trail with DFSERA10 exit CSLULALE.

Example 5-5 Sample output for printing the OM audit trail with the formatted option

```
*-----*
*      CONTROL STATEMENT : SPECIFIED      *
*          INPUT = SYSUT1                 *
*          OUTPUT = SYSPRINT              *
*  SELECTION QUALIFIERS :                 *
*      1. DEFAULT = ALL INPUT RECORDS     *
*-----*
```

```
CONTROL      CNTL H=EOF DDNAME=SYSUT1
OPTION       PRINT EXTR=CSLULALE
END

OM1OM  2007.242 19:21:52.47 CSL0020I OM  READY OM1OM
RM1RM  2007.242 19:22:43.91 CSL0020I RM  READY RM1RM
IMS1   2007.242 19:23:20.43 DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB MEM
IMS1   2007.242 19:23:20.86 DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB MEM
IMS1   2007.242 19:23:20.86 DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB MEM
IMS1   2007.242 19:23:23.45 DFS3613I - LUM TCB INITIALIZATION COMPLETE IMS1
IMS1   2007.242 19:23:23.61 DFS3613I - RLM TCB INITIALIZATION COMPLETE IMS1
IMS1   2007.242 19:23:23.61 DFS3613I - XCF TCB INITIALIZATION COMPLETE IMS1
IMS1   2007.242 19:23:23.61 DFS3613I - RLM TCB INITIALIZATION COMPLETE IMS1
IMS1   2007.242 19:23:23.62 DFS3613I - ALC TCB INITIALIZATION COMPLETE IMS1
IMS1   2007.242 19:23:23.62 DFS3613I - ALM TCB INITIALIZATION COMPLETE IMS1
IMS1   2007.242 19:23:23.62 DFS2088I APPC/OTMA SMQ Enablement inactive. Reason
IMS1   2007.242 19:23:23.62 DFS2088I APPC/OTMA SMQ Enablement inactive. Reason
IMS1   2007.242 19:23:23.65 DFS3613I - RRS TCB INITIALIZATION COMPLETE IMS1
IMS1   2007.242 19:23:23.66 DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB MEM
IMS1   2007.242 19:23:23.66 DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB MEM
```

Example 5-6 on page 110 shows output produced with the dump option for printing the OM audit trail with DFSERA10 exit CSLOERA3. This example shows OM audit trail log record types 0601 and 0801 for command input and command output for the QRY TRAN command where the OM input and OM output exit did not modify the messages.

Example 5-6 Printing the OM audit trail containing command input and output in dump format

0601 RECORD - 2007-08-30 22:34:01.721325 UTC - CSLOMCMD UNMODIFIED BY OM INPUT EXIT

```
000000 00000187 06010000 BF695E7B 429ED597 .... *...g.....;#..NpCSLPLEX10130....*
000020 0B9153B8 F6F78001 0A9A5288 BF695E7B .... *.j..67.....h...;#....USRT011 ....*
000040 00084040 40404040 4040E4E2 D9E3F0F1 .... *.          USRT011 .....USRT011 *
000060 00000000 00000000 00000000 00000000 .... *.
000080 00000000 00000000 00000000 00000000 .... *.
LINES 000080 TO 0000BF SAME AS ABOVE
0000C0 00FD4B98 00000000 7FFFF000 7FFFF000 .... *...q....".0.".0.....;#*
0000E0 429ED597 40404040 40404040 00000000 .... *.Np .....*
000100 00000000 00000000 00000000 00000027 .... *.
000120 D6D35C5D 40E2C8D6 E64DC3D3 C1E2E26B .... *.....QRY TRAN NAME(AP*
000140 00000000 00000000 00000000 00000000 .... *OL*) SHOW(CLASS,STATUS).....*
000160 00000000 00000000 00000000 00000000 .... *.
LINES 000160 TO 00017F SAME AS ABOVE
000180 00000000 000000 .... *.....*
```

0801 RECORD - 2007-08-30 22:34:01.721325 UTC - CSLOMRSP UNMODIFIED BY OM OUTPUT EXIT

```
000000 00000A9C 08010000 BF695E7B 429ED597 .... *.
000020 0B9153B8 F6F70001 0A9D3A08 BF695DF3 .... *.j..67.....)3....USRT011 ....*
000040 00084040 40404040 40400000 00000000 .... *.
000060 E4E2D9E3 F0F1F140 F1F0F1F5 F3F4F0F1 .... *USRT011 10153401..)3|...OM10M *
000080 00000000 00000000 00000000 00000000 .... *.
LINES 0000A0 TO 0000BF SAME AS ABOVE
0000C0 00000000 00000000 00000000 00000000 .... *.
0000E0 40404040 40404040 00000000 00000000 .... * .....*
000100 00000000 00000000 00000990 4C6FA794 .... *.
000120 6E4C5AC4 D6C3E3E8 D7C54089 94A296A4 .... *><!DOCTYPE imsout SYSTEM "imsout*
000140 4B84A384 7F6E4C89 94A296A4 A36E4C83 .... *.dtd"><imsout><ctl><omname>OM10M*
000160 4040404C 61969495 8194856E 4C9694A5 .... * </omname><omvsn>1.3.0</omvsn>*
000180 4CA79493 A5A2956E F2F04040 4C61A794 .... *<xmlvsn>20 </xmlvsn><statime>20*
0001A0 F0F64BF2 F5F840F2 F27AF3F4 7AF0F14B .... *07.242 22:34:01.721325</statime>*
0001C0 4CA2A396 A3899485 6EF2F0F0 F64BF2F5 .... *<stotime>2007.242 22:34:01.72226*
```

More details about printing OM audit trail records using DFSERA10 can be found in *IMS Version 10: System Utilities Reference*, SC18-9968.

5.1.2 Unsolicited output support

A message is defined as unsolicited when it is not generated as a response to a command. For example, system informational messages can be unsolicited messages. Prior to IMS

Version 10, unsolicited messages were sent only to the master terminal operator (MTO) for the particular IMS system where that message was generated. Users were unable to monitor unsolicited messages across an IMSplex.

With IMS Version 10, all unsolicited output messages, except those issued during IMS initialization before OM is available, are routed to Operations Manager (OM). Therefore, they are available to single-point-of-control (SPOC) applications, such as a user-written application using the REXX SPOC API. Prior to IMS Version 10, these unsolicited messages were not available to SPOC applications. With this new support, SPOCs can *subscribe* to Operations Manager to have these unsolicited messages routed to them. The subscribed SPOC will only receive unsolicited messages that are sent while the SPOC application is running; it does not receive unsolicited messages sent before it was not active. This SPOC application can then *unsubscribe* to OM when it no longer needs to receive unsolicited output messages.

OM will send unsolicited messages only to subscribed clients. The subscribed client must have an SCI input exit to receive the unsolicited output message. Messages go to the OM output user exit (if present) first, which can edit the message or indicate not to send the unsolicited message.

SPOC applications will use the following new requests:

- ▶ Assembler programs will use :
 - CSLOMSUB to subscribe
 - Their own SCI input exit to receive messages
 - CSLOMUSB to unsubscribe
- ▶ Rexx programs will use:
 - CSLULSUB to subscribe
 - CSLULGUM to receive messages
 - CSLULUSB to unsubscribe

For details of these new OM requests, refer to *IMS Version 10: System Programming API Reference*, SC18-9967.

The TSO SPOC application does not use these new OM unsolicited output message services; rather unsolicited messages can be accessed from the TSO SPOC by viewing the OM audit trail. See topic 5.5, “TSO SPOC enhancements” on page 120 for details of the TSO SPOC OM audit trail support.

There are three types of unsolicited output messages with this new support:

- ▶ Unsolicited messages created within the CSL environment using the OM CSLOMOUT request to create an unsolicited output message
These unsolicited output messages are logged to the OM audit trail.
- ▶ Unsolicited messages sent by IMS address spaces that did not go to the OM environment prior to IMS Version 10

In IMS Version 10, these now use the OM CSLOMOUT request to create an unsolicited output message. These unsolicited output messages are logged to the OM audit trail. In an IMSplex environment, these unsolicited output messages will be sent to all OMs in the IMSplex; therefore, the OM audit trail can contain multiple copies of an unsolicited output message.

Note that unsolicited messages created by IMS initialization before the OM environment is available will not use the OM CSLOMOUT request and those messages will not be logged.

- Command response for commands issued from a SPOC

These output messages are sent to all unsolicited message subscribers. The input commands and command responses are also logged to the OM audit trail.

There are two methods for controlling or tailoring which unsolicited messages are sent to OM and, therefore, are available or unavailable to subscribed SPOC applications:

- By using user modifiable tables for messages from CQS, CSL, and IMS.
- By using a new UOM= parameter in the DFSDFxxx or DFSCGxxx members of IMS.PROCLIB to control messages from IMS (DFS™-messages) only.

Using user modifiable tables

There are three optional user modifiable tables that can be used to indicate whether to send a message to OM as unsolicited output. The default is to send all messages. The three tables are:

- CSLZUMTU – CSL Unsolicited Output Messages
- CQSUOMTU – CQS Unsolicited Output Messages
- DFSUOMTU – IMS Unsolicited Output Messages

The tables are generated by using the following macros to populate the just mentioned tables with messages that should not be routed as unsolicited messages:

- CSLZUMT
- CQSUOMT
- DFSUOMT

You can also specify MESSAGE=SUPPRESS in the CSLZUMTU, CQSUOMTU, and DFSUOMTU message tables. If you code the following, shown in Example 5-7 for the message table DFSUOMTU, messages DFSxxxxx and DFSyyyyy will be sent to OM, message DFSzzzzz will not be sent to OM, and any other messages will not be sent to OM because of MESSAGE=SUPPRESS.

Example 5-7 Sample of using the user modifiable message table DFSUOMTU

```
DFSUOMT MESSAGE=DFSxxxxx,SEND=YES
DFSUOMT MESSAGE=DFSyyyyy,SEND=YES
DFSUOMT MESSAGE=DFSzzzzz,SEND=NO
DFSUOMT MESSAGE=SUPPRESS
```

Using the new UOM= IMS PROCLIB parameter

You can control which unsolicited output messages are sent to OM from IMS by using the new UOM= parameter that is available in either the DFSDFxxx or DFSCGxxx IMS PROCLIB members:

- With UOM=MTO specified, IMS unsolicited output messages that are destined for the MTO only or system console only (or both) will be sent to OM. IMS unsolicited output messages being sent to other destinations, such as user terminals, will not be sent to OM. UOM=MTO is the default.
- With UOM=NONE specified, no IMS unsolicited output messages will be sent to OM.
- With UOM=ALL specified, all IMS unsolicited output messages will be sent to OM.

5.2 IMS systems management enhancements

IMS Version 10 provides several system management enhancements that can improve the single system image view of your IMSplex. There are enhancements to type-2 commands for the IMSplex environment and additional capabilities for the secondary master terminal.

5.2.1 Global status

Global status is a new IMS Version 10 enhancement that focuses on improving the single system image in an IMSplex environment. This function maintains global command status for database, area, or transaction resources in the RM structure that can be applied during IMS restart. This allows the command status to be synchronized across the IMSplex even if an IMS system is down when the command was issued. Global status requires a Common Service Layer (CSL) environment including RM and a resource structure. A HALDB master database and its partitions are treated independently for global status.

Global status for databases, areas, and transactions is created in the following ways:

- ▶ By using a type-1 command with the GLOBAL parameter, such as /START, /STOP, /DBR, or /DBD
- ▶ By using a type-2 command with SCOPE(ALL), such as UPDATE

Global status exists only if set by one of these commands. Global status is never set by events in IMS.

Resources have both a global and a local status. Global status commands are processed by all active systems in an IMSplex. The local status is set in each IMS; local and global status are set in the RM structure. If global status is set while an IMS is down, when that IMS is restarted it will assume the global status set while it was down.

A new initialization parameter, PLEXPARM, specifies how global status is kept in the IMSplex. The next topic, 5.2.2, “IMSplex-wide parameter support” on page 113 discusses this new parameter.

5.2.2 IMSplex-wide parameter support

IMS Version 10 provides a new initialization parameter, PLEXPARM, which can be used to define the initial setting for global parameters (PLEXPARM values) to be maintained in an IMSplex for database status, DEDB area status, and transaction status. This parameter is available in both the DFSDFxxx and the DFSCGxxx IMS PROCLIB members and is used in conjunction with the new Global Status feature of IMS Version 10 described in topic 5.2.1, “Global status” on page 113.

The PLEXPARM parameter can be specified either in the DFSCGxxx PROCLIB member or the common service layer section of the DFSDFxxx PROCLIB member as follows:

PLEXPARM=(GSTSDB=Y | N,GSTAREA=Y | N,GSTSTRAN=Y | N)

Parameters are:

- ▶ GSTSDB=
Specifies how global status for databases is kept in the IMSplex
- ▶ GSTSAREA=
Specifies how global status for DEDB areas is kept in the IMSplex

► GSTSTRAN=

Specifies how global status for transactions is kept in the IMSplex

5.2.3 New and enhanced type-2 commands

IMS Version 10 has two new type-2 commands (CREATE and QUEUE), as well as enhancements to many existing type-2 commands.

QUEUE command

You can use the new QUEUE command to queue or dequeue messages to local or shared LTERMs or transactions. For queuing a message, you provide the message data in the command. All queued messages are single segment. For dequeuing, you can dequeue either a single message or all messages on the LTERM or transaction queue.

The format of the QUEUE command to enqueue a message is shown in Example 5-8.

Example 5-8 Format of the type-2 QUEUE command to enqueue a message

```
QUEUE TRAN | LTERM NAME(XXXX) OPTION(ENQ) DATA(message-data)
```

The format of the QUEUE command to dequeue a message is shown in Example 5-9.

Example 5-9 Format of the type-2 QUEUE command to dequeue a message

```
QUEUE TRAN | LTERM NAME(XXXX) OPTION(DEQ1 | DEQALL)
```

The QUEUE command is defined to OM as ROUTE=ALL. If no routing is provided at command entry, the QUEUE command will be sent to all IMSs registered for the QUEUE command in the IMSplex.

The QUEUE command with OPTION(ENQ) will be rejected if the TRAN or LTERM is a remote destination.

Security is provided by OM for the QUEUE command either via the OM security exit or RACF authorization.

CREATE command

You can use the new CREATE command to create resources and descriptors for dynamic definition of databases, programs, transactions, and routing codes with the new Dynamic Resource Definition (DRD) function in IMS Version 10. Refer to Chapter 6, “Dynamic Resource Definition” on page 133 for further details.

UPDATE command

Enhancements have been made to the UPDATE command in IMS Version 10 to support Dynamic Resource Definition (DRD) to update the attributes of a resource or descriptor and to change the status of a resource. There is new support for updating PGM (program) and RTC (routing code) attributes and for updating descriptor attributes. There is enhanced support for updating DB (database) and TRAN (transaction) attributes and for updating TRAN (transaction) status. Refer to Chapter 6, “Dynamic Resource Definition” on page 133 for further details.

DELETE command

Enhancements have been made to the DELETE command in IMS Version 10 to support Dynamic Resource Definition (DRD) to be able to delete resources and descriptors for

databases, programs, transactions, and routing codes. Refer to Chapter 6, “Dynamic Resource Definition” on page 133 for further details.

QUERY command

Several enhancements have been made to the QUERY command in IMS Version 10. It now supports new resource types for DRD, such as PGM (program), RTC (routing codes), and descriptors. You can also display information about PLEXPARM values. Enhancements have been made for querying databases, areas, transactions, the IMSplex, and IMSplex members.

QUERY resource for work:

► **QUERY *rsc-type* NAME(*names*) SHOW (WORK)**

For databases, programs, transactions, and routing codes, the output from this command will tell you the reasons that prevent an online change or some DRD commands from completing successfully.

Example 5-10 shows sample output from the QUERY ... SHOW(WORK) command.

Example 5-10 Sample output for QUERY TRAN NAME() SHOW(WORK)*

Trancode	MbrName	CC	Work
ADDACCT	IMS1	0	IN CONVERSATION
ADDCUST	IMS1	0	IN USE
DELACCT	IMS1	0	QUEUING
UPDACCT	IMS1	0	SCHEDULED
UPDCUST	IMS1	0	SUSPENDED
UPDINV	IMS1	0	ANOTHER CMD IN PROGRESS

QUERY resource for related work:

► **QUERY DB NAME(*names*) SHOW (PGM)**

The output from this command will return the names of programs that reference a database in their PSBs.

Example 5-11 shows sample output from the QUERY DB ... SHOW(PGM) command.

Example 5-11 Sample output for QUERY DB NAME(ACCTMSTR) SHOW(PGM)

DBname	MbrName	CC	PgmName
ACCTMSTR	IMS1	0	UPDACCT
ACCTMSTR	IMS1	0	DELACCT

► **QUERY TRAN NAME(*names*) SHOW(PGM)**

The output from this command will return the name of the program that the transaction invokes.

Example 5-12 shows sample output from the QUERY TRAN...SHOW(PGM) command.

Example 5-12 Sample output for QUERY TRAN NAME(ADDACCT) SHOW(PGM)

Trancode	MbrName	CC	PgmName
ADDACCT	IMS1	0	ACCOUNT

► **QUERY RTC NAME(*names*) SHOW(PGM)**

The output from this command will return the program associated with the routing code. Example 5-13 shows sample output from the QUERY RTC ... SHOW(PGM) command.

Example 5-13 Sample output for QUERY RTC NAME(GFP) SHOW(PGM)*

Rtcode	MbrName	CC	PgmName
GFP1	IMS1	0	FPACCT
GFP2	IMS1	0	FPACCT2

► **QUERY PGM NAME(*names*) SHOW(TRAN | RTC | DB)**

TRAN returns the names of the transactions that invoke the program.

RTC returns the names of the routing codes associated with the program.

DB returns the names of the databases referenced by the PSB.

Example 5-14 shows sample output from the QUERY PGM ... SHOW(DB) command.

Example 5-14 Sample output for QUERY PGM NAME(ADDACCT) SHOW(DB)

PgmName	MbrName	CC	RgnType	Database
ADDACCT	IMS1	0	MPP	ACCTMSTR
ADDACCT	IMS1	0	MPP	CUSTOMSTR

QUERY resource for timestamps:

► **QUERY *rsc-type* NAME(*names*) SHOW (TIMESTAMP)**

The output from this command will return four timestamps for DB (database), PGM (program), TRAN (transaction), and RTC (routing code) resources and descriptors:

- TIMECREATE - Time resource or descriptor was created.
- TIMEUPDATE - Time resource or descriptor was last updated.
- TIMEACCESS - Time resource or descriptor was last accessed or used.
- TIMEIMPORT - Time resource or descriptor was last imported from an RDDS.

Example 5-15 shows sample output from the QUERY TRAN ... SHOW(TIMESTAMP) command.

Example 5-15 Sample output from the QUERY TRAN(INVTRA) SHOW(TIMESTAMP) command*

Trancode	MbrName	CC	TimeCreate	TimeUpdate	TimeAccess
INVTRAR	IMS1	0	2007.248 12:33:15:16	2007.250 09:15:26.17	2007.250 10:
INVTRA7	IMS1	0	2007.250 08:15:12:20	2007.253 13:25:17:10	2007.253 15:

QUERY IMS for global PLEXPARM information:

► **QUERY IMS TYPE(PLEXPARM) SHOW (ALL)**

The output from this command will show the global PLEXPARM values for databases, areas, and transactions.

Example 5-16 on page 117 shows sample output from the QUERY IMS command.

Example 5-16 Sample output from the QUERY IMS TYPE(PLEXPARM) SHOW(ALL) command

MbrName	GLOBAL	GSTAREA	GSTSDDB	GSTSTRAN
IMS1	Y	Y	Y	Y
IMS1		Y	Y	Y
DBC1		Y	Y	NULL
DCC1		NULL	NULL	Y

QUERY member for IMSplex member information:

► **QUERY MEMBER TYPE(IMS) SHOW (ALL)**

The output from this command will show IMS member information such as DRD enablement and secondary master settings.

Example 5-17 shows sample output from the QUERY MEMBER TYPE(IMS) SHOW(STATUS) command that lists the secondary master settings.

Example 5-17 Sample output from the QUERY MEMBER TYPE(IMS) SHOW(STATUS) command

MbrName	Type	Status	LclStat
IMS1	IMS		SECCMD,SECCMDT,SECMSG
IMS2	IMS		SECMSG

INIT command

The INIT command has been enhanced for use with the new ACBLIB Member Online Change function in IMS Version 10. Refer to 2.9, “Online change enhancements” on page 50 for further details.

TERM command

The TERM command has been enhanced for use with the new ACBLIB Member Online Change function in IMS Version 10. Refer to 2.9, “Online change enhancements” on page 50 for further details.

Further details of enhancements for type-2 commands in IMS Version 10 can be found in *IMS Version 10: Command Reference, Volume 1*, SC18-9700 and *IMS Version 10: Command Reference, Volume 2*, SC18-9701.

5.2.4 Secondary master terminal enhancements

There are three enhancements for the secondary master terminal in IMS Version 10.

/SMCOPY command enhancement

The /SMCOPY command is used by the master terminal operator to control the printing of certain output to the secondary master terminal. In IMS Version 10, there is a new keyword, MSG, that controls the logging of system messages to the secondary master. Prior to this enhancement, system messages were always sent to the secondary master. Now, you can eliminate system messages from the secondary master with **/smcopy msg off**.

The purpose of this enhancement is to provide more user control over which messages are logged to the secondary master. Example 5-18 on page 118 shows examples of using the /SMCOPY command with the new MSG keyword.

Example 5-18 Examples of the /SMCOPY command with the new MSG keyword

```
/smcopy msg on
```

```
DFS058I 17:48:17 SMCOPY COMMAND COMPLETED
```

```
/smcopy msg off
```

```
DFS058I 17:49:26 SMCOPY COMMAND COMPLETED
```

Type-2 AOI exit routine (DFSABE00) enhancement

The Type-2 Automated Operator (AOI) exit routine, DFSABE00, can now specify if messages can be logged to the IMS secondary master terminal on a message-by-message basis. Previously, this exit could only control messages sent to the primary master terminal.

The exit decides during IMS initialization if it wants to be called for secondary master messages; if it is called for secondary master messages, it can reply to IMS to process the message normally or cancel the message to the secondary master. A new function code and two new origin values are passed to the exit, and a new return code is added.

The details of this new change for the DFSABE00 exit can be found in the *IMS Version 10: Exit Routine Reference*, SC18-9708.

QRY MEMBER command enhancement for secondary master settings

The QUERY MEMBER command has been enhanced to show the secondary master settings for each IMS in an IMSplex. Refer to topic 5.2.3, “New and enhanced type-2 commands” on page 114 for more details.

5.3 /DIS SHUTDOWN STATUS command enhancement

The /DISPLAY SHUTDOWN STATUS command has been enhanced to show active or in doubt ODBA threads that could prevent IMS from shutting down completely if IMS is not shutting down as expected.

The following additional line is added to the output of this command that contains the count of active and in doubt ODBA threads:

```
xxx ODBA THREADS, yyy INDOUBT ODBA THREADS active
```

where xxx is the number of active ODBA threads and yyy is the number of in doubt ODBA threads active in the system.

Example 5-19 shows the response to a /DISPLAY SHUTDOWN STATUS command with active or in doubt ODBA threads.

Example 5-19 Sample output of a /DISPLAY SHUTDOWN STATUS command

```
TERMINAL  USER      STATUS
1-      1          INPUT IN PROCESS
TERMINAL  USER      STATUS
NO OUTPUTTING LINES
MSG-IN    1      MSG-OUT    0
PRIMARY MSTR SECOND MSTR ACTIVE, MESSAGES WAITING
IMSLU=N/A.N/A          #APPC-CONV=          0  DISABLED
```

```
OTMA PHASE=0
  0 ODBA THREADS,      1 INDOUBT ODBA THREADS ACTIVE
*07242/135629*
```

5.4 Enhanced display of system parameters

Prior to IMS Version 10, we display the IMS system parameters during initialization in the DFS1929I message. However, the actual values that are used during online execution can be different because values, such as those for Advanced Program to Program Communication (APPC), Fast Path, and Shared Queues, might be changed when processing log records during restart. To address this possible inaccuracy, IMS Version 10 will issue this message twice: one time when IMS control region initialization completes and a second time when IMS control region restart completes. The first DFS1929I message will list the values supplied by the user either via system generation, execution parameters, or DFSPBxxx. The second DFS1929I message will display the actual values that are used during online execution.

The DFS1929I message is also enhanced to include the IMS version number and control region type in the message header and to suppress Fast Path parameters if Fast Path is not active. The message will also include the keyword INITIAL for the first message and the keyword ACTIVE for the second message. The format is:

```
DFS1929I * IMS SYSTEM PARAMETERS INITIAL | ACTIVE FOR THIS V nn.n crt EXECUTION
      where nn.n is the IMS version number and crt is the IMS control region type, such as
      DBDC, DBCTL, DCCTL, or FDBR.
```

This enhancement now allows the DFS1929I message to be used for diagnostic tasks, because it more accurately displays system parameter values.

Example 5-20 shows the initial IMS DFS1929I message.

Example 5-20 Sample of the first DFS1929I INITIAL message issued

```
DFS1929I * IMS SYSTEM PARAMETERS INITIAL FOR THIS V10.1 DBDC
EXECUTION: *IMSL
DFS1929I *      ALOT      = 60                                *IMSL
DFS1929I *      AOIP      = 2147483647                        *IMSL
DFS1929I *      AOIS      = N                                *IMSL
DFS1929I *      AOI1      = R                                *IMSL
DFS1929I *      APPC      = N                                *IMSL
DFS1929I *
DFS1929I *
DFS1929I *      USERVAR   =                                *IMSL
DFS1929I *      VAUT      = 1                                *IMSL
DFS1929I *      VSPEC     = DC                                *IMSL
DFS1929I *      WADS      = D                                *IMSL
DFS1929I *      WKAP      = 49152                            *IMSL
DFS1929I *      YEAR4     = N                                *IMSL
```

Example 5-21 on page 120 shows the second IMS DFS1929I message.

Example 5-21 Sample of the second DFS1929I ACTIVE message issued

```
DFS1929I * IMS SYSTEM PARAMETERS ACTIVE FOR THIS V10.1 DBDC
EXECUTION: *IMSL
DFS1929I *      ALOT      = 60                                *IMSL
DFS1929I *      AOIP      = 2147483647                        *IMSL
DFS1929I *      AOIS      = N                                *IMSL
DFS1929I *      AOI1      = R                                *IMSL
DFS1929I *      APPC      = N                                *IMSL
      .
      .
DFS1929I *      VAUT      = 1                                *IMSL
DFS1929I *      VSPEC     = DC                                *IMSL
DFS1929I *      WADS      = D                                *IMSL
DFS1929I *      WKAP      = 49152                            *IMSL
DFS1929I *      YEAR4     = N                                *IMSL
```

5.5 TSO SPOC enhancements

In IMS Version 10, the TSO SPOC enhancements focus on supporting new features in IMS Version 10, such as the OM audit trail, as well as enhancing options when using the SPOC, including a new batch capability.

5.5.1 Support for viewing the OM audit trail

The TSO SPOC has been enhanced in IMS Version 10 to view the Operations Manager (OM) audit trail. See topic 5.1.1, “OM audit trail” on page 106 for a description of the new OM audit trail.

There is one new TSO SPOC preference that needs to be set for viewing the OM audit trail:

- Specify a list of IMSplex member names or IMSplex member types from which to display audit trail information.

There are several TSO SPOC OM audit trail panels that can be used. From the TSO SPOC action bar, select SPOC, then choose option 6, Audit Trail to bring up the OM audit trail filter panel. This is where you provide the log stream name of the audit trail (specified on the IMSPLEXNAME parameter in the AUDITLOG parameter in the CSLOIxxx PROCLIB member) and the start and end date and time for the audit log records that you want to view. You can specify an asterisk (*) to use the current date and time.

Example 5-22 on page 121 shows the TSO SPOC OM audit trail filter panel.

Example 5-22 Example of the TSO SPOC OM audit trail filter panel

```

File   Action   Manage resources   SPOC   View   Options   Help
-----
PLEX1 IMSplex Audit Trail
Command ==> _____

Datastream name . .SYSLOG.OMAUDIT.LOG
Start date . . . . 2007/242
Start time . . . . 14:00:00
End date . . . . . 2007/242
End time . . . . . *

F1=Help F12=Cancel

```

Example 5-23 shows the TSO SPOC OM audit trail output panel for Example 5-22.

Example 5-23 Example of output returned for OM audit trail display

```

-----
File   Action   Manage resources   SPOC   View   Options   Help
-----
PLEX1 IMSplex Audit Trail
Command ==> _____

```

		Members. .	Type . .
MbrName	Time	Message	
IMS3	2007.242	16:20:52.61	DFSxxxxI message text
IMS4	2007.242	16:20:52.93	DFSxxxxW message text
IMS8	2007.242	16:20:52.94	DFSxxxxE message text
IMS3	2007.242	16:20:57.63	DFSxxxxI message text
IMS3	2007.242	16:20:57.66	DFSxxxxI message text
USRT002	2007.242	16:20:57.75	response for: QRY TRAN
IMS4	2007.242	16:20:58.75	DFSxxxxI message text
IMS8	2007.242	16:21:09.46	DFSxxxxI message text
IMS4	2007.242	16:21:11.54	DFSxxxxW message text
USRT003	2007.242	16:21:12.75	response for: QRY IMSPLEX
IMS8	2007.242	16:21:13.89	DFSxxxxI message text
IMS8	2007.242	16:21:13.58	DFSxxxxI message text
IMS8	2007.242	16:21:13.80	DFSxxxxI message text

```

F1=Help F7=Up F8=Down F12=Cancel

```

You must press Enter to see new messages (the panel is not updated until you press Enter). If the panel is full, you must press the Down arrow key to see more messages.

Some lines on the display represent the responses from type-2 commands. The command response is more complex and instead of appearing inline with other messages, a hotlink is provided. Position the cursor on the "Response for" line and press Enter to view the response from a type-2 command.

5.5.2 Batch SPOC utility

The batch SPOC is a utility program that uses OM services to submit IMS operator commands to members of an IMSplex. It will accept any commands supported by the OM API

in either type-1 or type-2 format. The IMSplex environment is specified as execution parameters to the utility. Multiple commands are supported and they are executed serially. The output is similar in format to the output of the TSO SPOC.

The batch SPOC can be used to add additional steps to a batch job, for example, to add /DBR and /STA commands before and after a database reorganization job.

The batch SPOC utility is invoked by the use of JCL statements. Example 5-24 shows the JCL necessary to run the batch SPOC utility.

Example 5-24 Sample JCL for invoking the batch SPOC utility

```
//SPOCKPB JOB ,  
// MSGCLASS=H,NOTIFY=&SYSUID,USER=&SYSUID  
//SPOC EXEC PGM=CSLUSPOC,  
// PARM=('IMSPLEX=PLEXB,ROUTE=IM1B,WAIT=30')  
//STEPLIB DD DSN=IMSPSA.IMOB.SDFSRESL,DISP=SHR  
//SYSPRINT DD SYSOUT=X  
//SYSIN DD *  
    QRY IMSPLEX SHOW(JOB,TYPE, +  
                        STATUS)  
    QRY TRAN NAME(INV1*) SHOW(ALL) /* inventory appl */  
/DIS DB ACCTMSTR
```

The program parameters defined on the EXEC statement are:

- ▶ **IMSPLEX**
The 1-5 character suffix of the IMSplex name.
- ▶ **ROUTE**
The SYSIDs of IMSplex members that are to run the commands. If ROUTE is not specified, all members of the IMSplex will run the commands.
- ▶ **WAIT**
Specifies the wait time for individual commands.

You provide the SYSIN file that contains the commands that you want to execute. The commands are executed serially. When one command completes, the next command is executed until all records from the SYSIN file are processed. Continuation of the SYSIN control statements is specified by a plus sign (+) or a minus sign (-) as the last non-blank character of the line. A plus sign removes the leading spaces from the next line; a minus sign keeps leading spaces. Comments can be included within the SYSIN file and are specified as: /* this is a comment */.

The SYSPRINT file will contain the formatted command responses. If more than one command is issued, the responses will appear in the same order as the commands appear in the SYSIN file. The command responses are formatted similarly to the TSO SPOC display. The default record length is 133. If the records are too long, they wrap to the next line. You can specify DCB information in the JCL or in the data set allocation to allow longer records in the SYSPRINT file.

If SYSPRINT is a SYSOUT file, the System Display and Search Facility (SDSF) can be used to view the batch job output.

Example 5-25 on page 123 shows a sample batch job output where WAIT is either specified or defaulted.

Example 5-25 Sample output from the batch SPOC utility with WAIT= specified or defaulted

```
=====
Log for. . : QRY IMSPLEX SHOW(MEMBER,TYPE,STATUS)
IMSpIex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . : 2007.242 15:36:28.11
Stop time . . . . : 2007.242 15:36:29.17
Return code . . . : 00000000
Reason code . . . : 00000000
Command master. . : SYS3
IMSpIex   MbrName   CC Member   Type   Status
CSLPLEX1  OM1       0 SYS3     DBDC   ACTIVE
CSLPLEX1  OM1       0 IMS2     DBDC   ACTIVE
=====
```

Example 5-26 shows a sample batch job output where WAIT=0 is specified; the command response is not available and not printed for this case; only summary information is printed.

Example 5-26 Sample output from the batch SPOC utility with WAIT=0 specified

```
=====
Log for. . : QRY IMSPLEX SHOW(JOB,TYPE,STATUS)
IMSpIex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . : 2007.242 15:36:28.11
=====
```

5.5.3 TSO SPOC action bar

The TSO SPOC action bar has been enhanced to include new options for managing resources, working with individual resources, setting views, and setting optional preferences. Example 5-27 shows the TSO SPOC action bar on the command entry panel for IMS Version 10.

Example 5-27 TSO SPOC action bar

```
File   Action   Manage resources   SPOC   View   Options   Help
-----
IMSL                                IMS Single Point of Control
Command ==> _____

----- Plex . .           Route . .           Wait . .

F1=Help F12=Cancel
```

The Manage resources pull-down menus invokes the new application for managing dynamic resources (DRD). The SPOC pull-down menu was previously called Display. The View pull-down menu has two new options: Refresh and Reset.

The pull-down menus from the action bar follow:

- ▶ File
 - 1. Save as ...
 - 2. Print
 - 3. Print All
 - 4. Select All
 - 5. Deselect All
- ▶ Action
 - 1. Update
 - 2. Copy
 - 3. Delete
 - 4. Show details
- ▶ Manage resources
 - 1. Query
 - 2. New
 - 3. Update
 - 4. Delete
 - 5. Export
 - 6. Import
- ▶ SPOC
 - 1. Cmd entry & response
 - 2. Cmd entry & log
 - 3. Command status
 - 4. Command shortcuts
 - 5. Expand command...
 - 6. Audit trail...
- ▶ View
 - 1. Find...
 - 2. Sort...
 - 3. Refresh
 - 4. Reset
- ▶ Options
 - 1. Preferences...
 - 2. Set IMS groups...
- ▶ Help
 - 1. Help for Help...
 - 2. Extended help...
 - 3. Keys help...
 - 4. Help Index...
 - 5. Tutorial...
 - 6. About...

5.5.4 TSO SPOC support for dynamic resource definition

From a TSO SPOC session, you can use the ISPF user interface to issue various DRD resource definition and operational commands.

The Manage resources application, a new application option from the IMS application menu, provides a set of ISPF SPOC-like panels to assist in performing dynamic resource definition functions. Refer to Chapter 6, “Dynamic Resource Definition” on page 133 for more details of the Manage resources application.

5.5.5 New TSO SPOC preferences

New TSO SPOC preference options provide the following facilities:

- ▶ Specify whether to clean up command responses when you exit the TSO SPOC. This is listed under Default Exit Option.
- ▶ Choose one of two views for command display: a list view or a command syntax view. These are listed under Manage resources preferences. For more information about the Manage resources application, refer to 6.4.6, “Manage Resources application” on page 153 on DRD.
- ▶ Specify how many commands to keep in the command history and how long a command response is to be kept. These are listed as Storage Management Preferences.
- ▶ Specify a list of IMSplex member names or IMSplex member types from which to display audit trail information. These are listed as Audit Trail Preferences. For further information about the TSO SPOC support for the OM audit trail, refer to 5.5.1, “Support for viewing the OM audit trail” on page 120.

Example 5-28 shows the first panel format for TSO SPOC preferences.

Example 5-28 First panel of TSO SPOC preferences

```
Help
-----
IMSL                IMS Single Point of Control Preferences

Command ==>
Select your options and press the Enter key.

Command Entry Preferences
Default IMSplex . . . . IMSL
Default routing . . . . IMSL

Wait interval . . . . .
Waiting preference . . 1  1. Wait for command to complete.
                        2. Do not wait for command response.

Command shortcuts . . .  1. Use command shortcuts.
                        2. Do not use command shortcuts.

Shortcut processing . .  1. Merge explicit and default parameters.
                        2. Explicit parameters override defaults.

Initial view . . . . .  1. SPOC command panel.
                        2. SPOC status list.

Default exit option . . 1  1. Show exit confirmation panel.
                        2. Keep command responses.

More:      +
```

3. Erase command responses.

Example 5-29 shows the second panel format for TSO SPOC preferences.

Example 5-29 Second panel of TSO SPOC preferences

```
Help
-----
IMSL                      IMS Single Point of Control Preferences

Command ==>
Select your options and press the Enter key.

                                2. Keep command responses.
                                3. Erase command responses.
                                More: -

Manage Resources Preferences
Resource definition
dialog layout . . . . . 1  1. List view.
                        2. Command syntax view.

Storage Management Preferences
Maximum number of commands . . . . . in status list.
Delete command responses after . . . . . minutes

Audit Trail Viewer Preferences
Member List . . . . .

Type List . . . . .

F1=Help  F12=Cancel
```

5.5.6 New help capabilities

Additional help capabilities enable you to position the cursor on the column heading in QUERY display panels and then press F1 to retrieve field-specific help topics.

5.5.7 TSO SPOC input user exit support

The TSO SPOC executable, DFSSPOC, is enhanced to support input user exit routines. When a TSO SPOC session is started with an input user exit routine, the exit routine has the ability to modify or reject the command parameters, set return codes and reason codes, and send text messages to the TSO SPOC session.

TSO SPOC user exit routines run in ISPF and pass information through the ISPF environment. The exit routines can use the ISPF commands VGET and VPUT to view and alter variables used by the TSO SPOC. There is support for both program exits (EXITPGM) and command exits (EXITCMD) when you invoke the DFSSPOC exec.

For further information about the DFSSPOC command, refer to the next topic 5.5.8, “TSO SPOC invocation command enhancements” on page 127.

5.5.8 TSO SPOC invocation command enhancements

New CMD() and TYPE(DISPLAY) parameters for the DFSSPOC command allow application programmers to issue IMS operator commands, display the command responses, and scroll through the data.

The DFSSPOC command supports several parameters. External programs can issue IMS operator commands and display the command response. You can scroll through the data, but no other SPOC interactions are allowed. The DFSSPOC command accepts these new parameters:

- ▶ **DFSSPOC**
Specifies the TSO SPOC command name
- ▶ **CMD(*command*)**
Specifies that a command is issued immediately. The response displays in the first SPOC panel. Only commands supported by the OM API can be used.
- ▶ **OPT(*selection*)**
Jumps to a specific Manage Resources menu without viewing intermediate menus where (*selection*) is a value (1, 2, 3, and so on) for a particular menu
- ▶ **EXITCMD**
Specifies the user exit that TSO SPOC invokes before sending the command to OM. This can be either a single or a list of user exits. Command exits are called with a TSO command processor parameter list. All command exits are called before the program exits. On entry to the routine, register 1 points to the Command Processor Parameter List (CPPL) and is defined by macro IKJCPPL.
- ▶ **EXITPGM**
Specifies the user exit that TSO SPOC invokes before sending the command to OM. Can be either a single user exit or a list of user exits. Program exits are called with a z/OS batch program parameter list. Command exits are called before the program exits. On entry to the routine, register 1 points to a standard parameter list. Register 1 points to a full word, which points to a half-word length, which is followed by the parameter string.
- ▶ **TYPE(DISPLAY)**
Displays the command response only, with no other SPOC interactions
- ▶ **TYPE(SPOC)**
Indicates that the normal TSO SPOC display is used. This is the default if TYPE is not specified.
- ▶ **APPL(*newappl*)**
Indicates a user-specified application ID that is used by ISPF to fence off different applications. This is a one to four character value. The first character is alphabetic and the remaining characters are alphanumeric. If no value is specified, a default of "/CSLU" is used.
- ▶ **PLEX(*plexname*)**
Specifies the name of the IMSplex to which to issue the command. If no IMSplex name is specified, the user's default IMSplex name is used.
- ▶ **ROUTE(*ims_names*)**
Specifies to which members of the IMSplex to route the command. Names must be separated by commas.

- ▶ **ROUTE(*)**
Specifies that the command is routed to all registered command processing clients in the IMSplex.
- ▶ **ROUTE(%)**
Specifies that the command is routed to only one command processing client in the IMSplex that is registered for the command and that has MASTER capability. The Operations Manager chooses the command processing client.
- ▶ **WAIT(*wait_time*)**
Specifies in *MM:SS* format how long OM waits for member responses before returning a response. The default wait time is 5 minutes.

Refer to *IMS Version 10: Operations and Automation Guide*, SC18-9716, for further details about using these DFSSPOC command parameters.

5.6 REXX API enhancements

REXX API enhancements in IMS Version 10 support the new Operations Manager (OM) unsolicited output message capability and provide a new XML parser.

5.6.1 Support for OM unsolicited output messages

The REXX SPOC API has been enhanced to support OM unsolicited messages. REXX applications can now:

- ▶ Subscribe for unsolicited messages using the CSLULSUB function
- ▶ Unsubscribe for unsolicited messages using the CSLULUSB function
- ▶ Retrieve any waiting messages using the CSLULGUM function

Example 5-30 contains a sample program that uses OM requests CSLULSUB, CSLULBGUM, and CSLULUSB to subscribe to OM and to retrieve unsolicited messages.

Example 5-30 Sample REXX program that accesses OM unsolicited output messages

```

/* rexx */
/*-----
| REXX SPOC API example to invoke unsolicited output message |
| functions.                                                    |
|-----*/

/*-----
| tuning parameter:  check every 10 seconds                    |
|-----*/
interval = 10

/*-----
| We want to make syscalls, that is, sleep                      |
|-----*/
Call syscalls 'ON'

/*-----
| Establish IMS rexx environment                                |
|-----*/
Address LINK 'CSLULXSB'
If rc = 0 Then

```



```

Do
  Address IMSSPOC
/*-----
| Subscribe to messages from IMSplex named 'PLEX1' |
-----*/
  continu = 1
  Do while(continu)
    subrc = CSLULSUB('PLEX1')
    say 'subrc=('subrc')'
    If subrc = '01000010X' Then
      Do
        Say time()
        Address syscall "sleep" interval
      End
    Else
      continu = 0
    End
  End

Do a = 1 To 25
/*-----
| wait a little before checking for new messages |
-----*/
  Address syscall "sleep" interval

/*-----
| Check if any unsolicited messages are present. |
-----*/
  results = CSLULGUM('PLEX1','xml.')
  say 'a='a 'results=('results')'

  if xml.0 /= '' Then
    Do
/*-----
| Display any messages in unsolicited message array. |
-----*/
      say 'xml.'0' = ('xml.0')'
      Do idx = 1 To xml.0
        say 'xml.'idx'=('xml.idx')'
      End
    End
  End
/*-----
| Unsubscribe to unsolicited messages. |
-----*/
  usbrc = CSLULUSB('PLEX1' )

/*-----
| clean up REXX SPOC API |
-----*/
  "END"
End
Exit

```

For further details of the new OM unsolicited output message support, refer to topic 5.1.2, “Unsolicited output support” on page 110.

5.6.2 The REXX XML parser

The REXX SPOC API has been enhanced to include a REXX parser for OM XML. There is a new request, CSLULGTP, that can be used to retrieve command responses from OM and put the command response into a REXX stem variable. A *stem variable* is a complex variable into which the command responses are added. The REXX program then refers to the information in the stem variable directly. Before this new enhancement, the REXX SPOC application had to use the CSLULGTS request and parse the XML statements itself by looking for XML tags.

Here is the format of the CSLULGTP request:

CSLULGTP (*stem_name* , *token_name* , *wait_time*)

Parameter descriptions:

► *stem_name*

After the CSLULGTP request completes successfully, the REXX stem variable is populated with the command response that is returned by OM. The REXX program can then refer to the command response and take appropriate action.

► *token_name*

The name of the command and response token (CART). The token name should match the name specified on the CART subcommand.

► *wait_time*

A timeout value for the CSLULGTP command. The CSLULGTP command waits until the command completes, but the wait lasts only as long as the time specified. The wait time is in the format *MMM:SS* or *ssss*. The maximum timeout value is 999:59. Enclose this value in quotation marks.

Note: This timeout value is not the same as the timeout value for the WAIT subcommand; however, this *wait_time* should be at least as long as the value specified on the WAIT subcommand. Otherwise, no command responses are received for long running commands. If no response is received, issue the CSLULGTP function again.

Example 5-31 shows sample code for using a CSLULGTP request. This code issues the IMS command QRY TRAN NAME(A) and uses the CSLULGTP request to retrieve the command response.

Example 5-31 Sample code for retrieving command responses using the CSLULGTP request

```
Address IMSSPOC
"ims PLEX1"
"wait 5:00"
cartid = 'PROD12'
"CART" cartid
"QRY TRAN NAME(A*)"
results = csulgtg('qinfo.', cartid,"1:30")
If qinfoctl.rc = 0 Then
  Do
    say "OM name =("qinfoctl.omname")"
    say "command master =("qinfo.cmd.master")"
  End
```

The user provides the stem name, but the variable names have a fixed portion that corresponds to specific XML tags. See *IMS Version 10: System Programming API Reference*, SC18-9967, for a list of the fixed portion of the variable names.

Archived

Archived



Dynamic Resource Definition

In this chapter, we introduce Dynamic Resource Definition (DRD), which allows you to dynamically create, update, delete, and query resource definitions. We begin by reviewing the different resource definitions traditionally managed by IMS MODBLKS system generation. Then, we compare DRD with the traditional method of modifying these definitions, revealing DRD's significant availability benefits. Next, we list the prerequisites for DRD and show how to install it in your shop. Finally, we review how to use this new capability, including how to fold it into your existing automated operations.

6.1 IMS resource definitions

An IMS system uses databases, application programs, transactions, and routing codes in its operations. These *resources* are represented in an online IMS system by *control blocks*, which each contain the name, attributes, and status of the resource that it represents:

- ▶ The name is 1-8 characters with the following restrictions:
 - Alphanumeric (A through Z, 0 through 9, #, \$, and @)
 - Cannot begin with DFS and cannot be BASICEDT, DBCDM, DBRC, ISCEDT, MSDB, SDB, or WTOR
- ▶ The attributes are properties of the resource specified during resource definition.
- ▶ The status is the availability or usability of the resource.

We now compare the traditional methods of managing these resources with the new DRD capability.

6.1.1 Managing resource definitions without DRD

Without DRD, the IMS system's resources have to be defined by using the IMS system definition (also known as the *IMS system generation* or *SYSGEN*) process, which consists of several macros. When you run these macros, control blocks are created and stored in a MODBLKS data set that you have previously allocated. These control blocks are later used by your online IMS system as resources. See Table 6-1 to determine which control blocks are created for each resource when its SYSGEN macro is run.

Table 6-1 Resources and their respective SYSGEN macros and control blocks

Resource	SYSGEN Macro	Control Block
Database	DATABASE	DDIR
Application program	APPLCTN	PDIR
Transaction	TRANSACT	SMB
Routing code	RTCODE	RCTE

The control blocks contained in the MODBLKS data set are known as *stored resource definitions*. IMS reads these definitions into its control region during cold start. After these resource definitions have been read into the IMS control region, they are known as *runtime resource definitions*. Without DRD, the only way that you can modify these definitions is by running another SYSGEN and either restarting your IMS system or completing a MODBLKS online change. The required system outage or resource quiescing that occurs during the online change process impacts resource availability. See Figure 6-1 on page 135 for an illustration of this process.

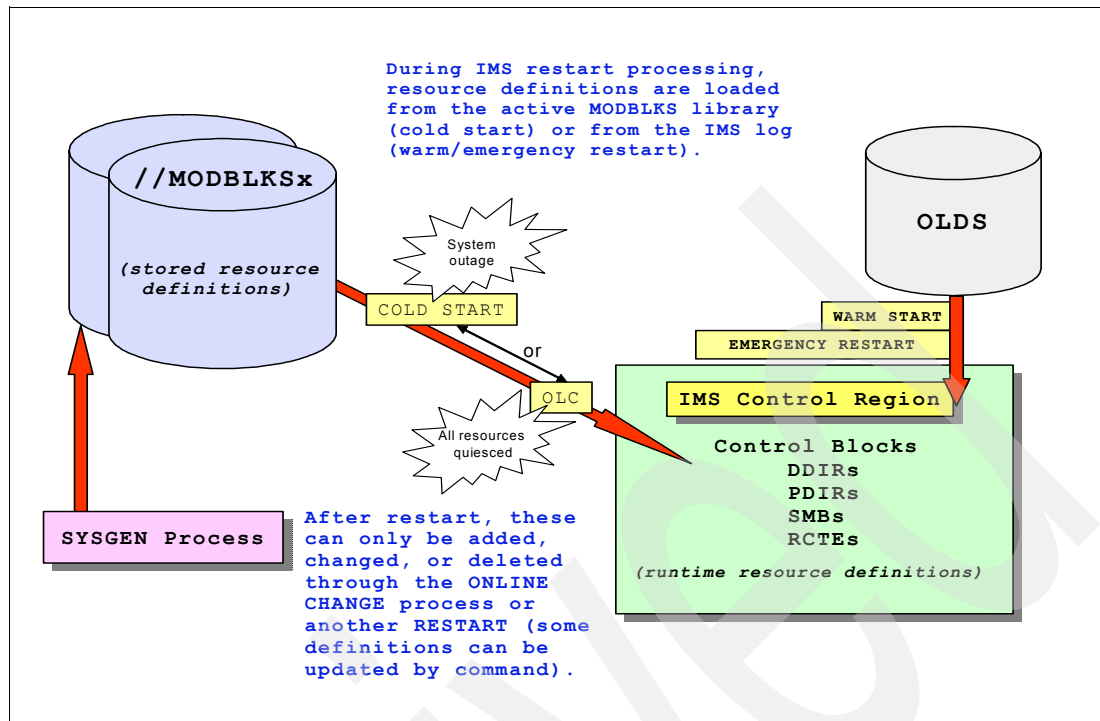


Figure 6-1 Changing resource definitions while IMS is online without DRD enabled

6.1.2 DRD: A lightweight way to manage resource definitions

DRD allows you to dynamically create, update, delete, and query your resource definitions. When you use DRD to manage your resource definitions while IMS is online, the requirement for a MODBLKS SYSGEN is eliminated. Consequently, you also no longer need to recycle your IMS system or perform a MODBLKS online change to bring these newly defined resources online. The primary value of DRD is higher resource availability for your IMS system.

Resource definition data set

DRD uses a BSAM data set called the *system resource definition data set (RDDS)* to hold your stored resource definitions. The first time that you cold start IMS with DRD enabled, the RDDS will be empty. In this case, IMS can import the stored resource definitions from the MODBLKS data set into the online control region. You can then export these resource definitions to the RDDS using DRD's automatic export capability, which allows you to write the definitions to the RDDS at every system checkpoint. The next time that you cold start IMS, it will be able to import the stored resource definitions from the RDDS into the online control region and the MODBLKS data set will no longer be required.

Rather than using the MODBLKS SYSGEN process to manage your resource definitions, you can now utilize DRD commands to dynamically create, update, delete, and query them. See Figure 6-2 on page 136 for an illustration of this process. If you compare this figure to Figure 6-1, you will see that the MODBLKS data set has been replaced as the primary data set by the RDDS. You will also see that the SYSGEN and OLC processes that are used to manage the online resources have been replaced by the automatic import and export functions, as well as by the DRD commands themselves.

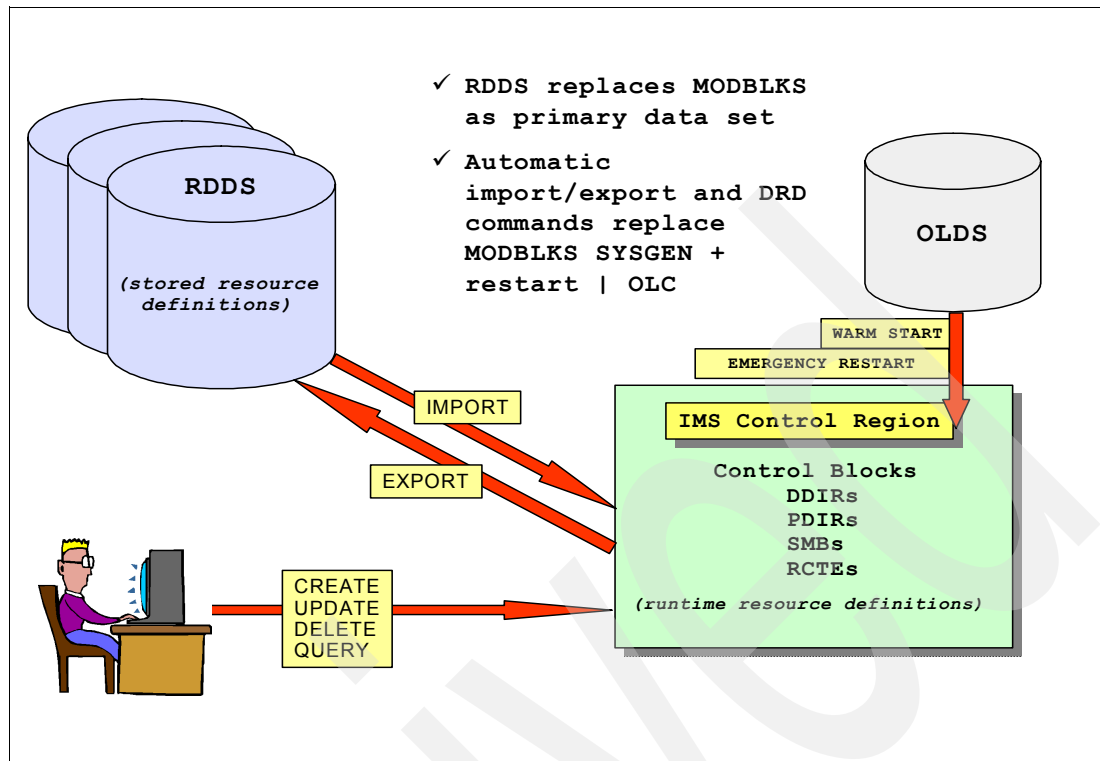


Figure 6-2 Changing resource definitions using the DRD capability while IMS is online

6.2 Prerequisites for DRD

To enable DRD, you must first set up the required environment before starting the actual migration process. This includes installing prerequisite functions and preparing the data sets that you use with DRD.

6.2.1 Install the Common Service Layer

DRD requires the Structured Call Interface (SCI) and Operations Manager (OM) components of the Common Service Layer (CSL). If you have not previously installed the CSL, you must do so now to use the DRD capability. The IMS control region interacts with the CSL via two PROCLIB members: DFSCGxxx and DFSDFxxx. You will find instructions for defining these members in the *IMS Version 10: System Definition Reference*, GC18-9966, and the *IMS Version 10: System Definition Guide*, GC18-9998.

6.2.2 Allocate and define resource definition data sets

While you should have at least two system resource definition data sets (RDDSs) to contain your stored resource definitions, we recommend that you define, allocate, and catalog three of them. The main reason for defining multiple RDDSs is to accommodate any I/O errors that might occur at export time. The RDDS data sets are used in a round-robin fashion. At export time, IMS locates the RDDS with the oldest data and exports the resource and descriptor definitions to that RDDS. If an I/O error occurs while trying to write to that RDDS, IMS locates the RDDS with the next oldest data and tries to write the resource and descriptor definitions to that RDDS. IMS keeps trying different RDDSs until a successful export occurs or until the

only RDDS left is the RDDS with the most recent data. IMS will never export to the RDDS that contains the definitions from the most recent export.

The RDDSs are not managed like the RECON data sets. Automatic export only writes one copy of the resource and descriptor definitions to one RDDS data set. It does not write the same definitions to a second RDDS. If you require a backup copy of the RDDS with the most recent data, you must make a copy of the data set. One option is to use DASD mirroring.

You can allocate your RDDSs using either the Installation Verification Program (IVP) sample DRD application, the IEBGENER program, or the ISPF ALLOCATE function. It is important that each RDDS starts with an End of File (EOF) marker to avoid unpredictable results. Any of these three methods of allocating the data set will ensure that this marker is in place. You might be aware of the IEFBR14 program, which can also be used to allocate data sets. We do not recommend that you use this program, because it will not place the EOF marker in the data set.

The data sets will be dynamically allocated to the IMS control region using DFSRDDDD as the DD name, and their layout will begin with a header record followed by one or more resource definition records. Supporting RDDS allocation documentation is in the *IMS Version 10: System Definition Guide*, GC18-9998.

Using the IVP

If you are already familiar with the IVP, you know that its primary use is for testing an IMS installation and verifying that all of its major functions operate correctly. The IVP allows you to generate and start a sample IMS system, which you can then use to test different sample applications that are also provided by the IVP. One of these is the sample DRD application, which you can use to allocate and define RDDSs that will contain your stored resource definitions. The specific job that allocates the data sets is named *IV_O101J* and can be found in the O series of jobs and tasks in the IVP. If you run this job, the IVP will create three RDDSs named IMS.RDDS1, IMS.RDDS2, and IMS.RDDS3.

You can find more detailed information about the DRD sample application that allocates the RDDSs in the *IMS Version 10: Installation Guide*, GC18-9710 .

Using the IEBGENER program

Another way to allocate and define the system resource definition data sets is by using the IEBGENER program. Creating the data sets using this method will ensure that the EOF marker is placed at the beginning of each data set.

The IEBGENER program JCL must include a block size designation (BLKSIZE) for each data set defined. While the BLKSIZE parameters of the DD statements are variable, we recommend that you set it to the maximum value of BLKSIZE=32760 to reduce the total number of I/Os when IMS is importing and exporting resource definitions to and from the RDDS. Compared to the MODBLKS data set traditionally used to contain the stored resource definitions, the RDDS is approximately 33% larger when it contains these same definitions. If you use the sample JCL included in Example 6-1 on page 138 to allocate your RDDSs but want to modify it to your specifications, remember that each RDDS data set name (DSN) can be 1 to 44 characters in length, as well as the following conditions for the SYSUT2 DD statement:

- ▶ The RECFM should be RECFM=VB.
- ▶ The BLKSIZE parameter can be between 4096 and 32760 (inclusive) and should be the same for each DD statement to avoid reblocking of the records during auto export.
- ▶ The LRECL parameter value should be exactly 4 bytes less than the specified BLKSIZE.

Sample JCL that will allocate three RDDSs is available in the DFSRDDAL member of your IMS.SDFSSLIB data set. For a sample of this JCL, see Example 6-1.

Example 6-1 Sample JCL that allocates three system resource definition data sets

```
//DELETE EXEC PGM=IDCAMS,REGION=1024K
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE IMSTESTL.RDDS1
DELETE IMSTESTL.RDDS2
DELETE IMSTESTL.RDDS3
SET MAXCC=0
/*
//ALLOC1 EXEC PGM=IEBGENER
//SYSUT1 DD DUMMY,BLKSIZE=32760,RECFM=VB
//SYSUT2 DD DSN=IMSTESTL.RDDS1,
//          DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760),
//          UNIT=SYSDA,VOL=SER=VOL001,
//          DISP=(,CATLG),SPACE=(TRK,(10,10))
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
/*
//ALLOC2 EXEC PGM=IEBGENER
//SYSUT1 DD DUMMY,BLKSIZE=32760,RECFM=VB
//SYSUT2 DD DSN=IMSTESTL.RDDS2,
//          DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760),
//          UNIT=SYSDA,VOL=SER=VOL002,
//          DISP=(,CATLG),SPACE=(TRK,(15,15))
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
/*
//ALLOC3 EXEC PGM=IEBGENER
//SYSUT1 DD DUMMY,BLKSIZE=32760,RECFM=VB
//SYSUT2 DD DSN=IMSTESTL.RDDS3,
//          DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760),
//          UNIT=SYSDA,VOL=SER=VOL003,
//          DISP=(,CATLG),SPACE=(TRK,(20,20))
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
/*
```

Using the ISPF ALLOCATE utility

The final alternative for allocating your RDDSs is using the ISPF ALLOCATE utility. Using this allocation method will ensure that the EOF marker is placed at the beginning of each data set as well.

Special RDDS allocation considerations for backup IMS systems

We recommend that you allocate and define a set of RDDSs for an XRF alternate IMS system, as well as for a database control (DBCTL) warm standby. An XRF alternate IMS will remain synchronized with dynamic resource definition (DRD) activity by applying X'22' log records that are written after each DRD command is processed on the active IMS system. The application of an X'4098' log record, which signifies the end of a system checkpoint, will result in an automatic export being done to the RDDS if there are changes present since the

last system checkpoint. A DBCTL warm standby requires a set of RDDSs only if you plan on using the automatic import/export functions for it.

6.3 Enabling DRD

After the DRD prerequisites are in place, you can proceed with the migration process. Begin with shutting down your IMS system normally. The next steps involve defining the DRD parameters, which we will now describe.

6.3.1 Define DRD parameters in DFSDFxxx PROCLIB member

To enable DRD, define the DRD parameters in the DRD and CSL sections of the DFSDFxxx PROCLIB member and then cold start the IMS system. Let us take a look at the parameters and their possible values for each of these sections. The parameter values can be found in *IMS Version 10: System Definition Reference*, GC18-9966.

DRD section

Here is a list of the DRD parameters that you should define in the DRD section of the DFSDFxxx PROCLIB member, along with their possible values (the default value is underlined):

- ▶ AUTOEXPORT=AUTO | NO | RDDS

Specifies whether all resource and descriptor definitions should be exported at checkpoint time (simple or normal shutdown). The automatic export takes place only if definitional changes were made to any resource since the time of the last checkpoint, except for the restart checkpoint. After IMS completes restart processing, a restart checkpoint is taken. Export occurs after the restart checkpoint if automatic export is enabled:

- AUTO

IMS determines whether or not to enable automatic export. Automatic export is enabled if two or more system resource definition data sets are defined and accessible. AUTOEXPORT=AUTO is the default.

- NO

Automatic export is disabled. No resource or descriptor definitions are exported at checkpoint time.

- RDDS

Automatic export is enabled if two or more system resource definition data sets are defined and accessible. All resource and descriptor definitions are exported to the oldest system resource definition data set at checkpoint time, which overlays the oldest data.

- ▶ AUTOIMPORT=AUTO | MODBLKS | NO | RDDS

Specifies whether resource and descriptor definitions should be automatically imported during IMS cold start:

- AUTO

IMS determines whether or not to enable automatic import processing. If IMS enables automatic import, it also determines the data source from which to import the definitions. See Table 6-2 on page 140 to see the conditions under which IMS imports from these different data sources. If the conditions under which the RDDS is imported are not all true, IMS will attempt to import the stored resource definitions from the

MODBLKS data set. If the MODBLKS conditions are not all true, no definitions are imported at all. AUTOIMPORT=AUTO is the default.

– MODBLKS

Resource definitions are imported from the MODBLKS data set if one is defined. If definitions are imported from the MODBLKS data set, the values specified for the IMPORTERR= and RDDSERR= parameters do not apply and are ignored if specified. If an error occurs, IMS cold start is stopped.

– NO

No resource or descriptor definitions are imported when IMS cold starts.

– RDDS

Resource and descriptor definitions are imported from the most recently updated system resource definition data set.

Table 6-2 Automatic importing conditions for importing resource definitions during IMS cold start

Import from RDDS	Import from MODBLKS	No import done at all
Two or more system RDDSs are defined in the DFSDFxxx PROCLIB member data set. All of the defined RDDSs are allocated and able to be read. At least one of the RDDSs contains valid resource and descriptor definitions.	The MODBLKS data set exists. No RDDSs are defined in the DFSDFxxx PROCLIB member data set. The MODBLKS data set exists, and all of the defined system RDDSs are empty.	Two or more system RDDSs are defined, but an error occurs trying to access one or more of the data sets. No RDDSs are defined, and no MODBLKS data set is defined.

► DCLWA=Y | N

Specifies the default log write ahead option for transactions that are defined with the CREATE TRAN command. DCLWA= specifies whether (Y) or not (N) IMS should perform log write-ahead for recoverable nonresponse-mode input messages and transaction output messages. The default is Y. The IMS transaction descriptor DFSDSTR1 is defined with the DCLWA value that was defined in this DFSDFxxx PROCLIB member data set. Any subsequent CREATE TRAN commands that are issued either without the LIKE keyword, or with LIKE(DESC(DFSDSTR1)), are defined with this DCLWA value:

– Y

Specifies that information in the log buffers is written to the IMS log before the associated input acknowledgment or output reply is sent to the terminal. A nonresponse-input transaction is made recoverable across IMS failures before IMS acknowledges receipt of the input, and database changes are made recoverable before IMS sends associated output reply messages.

– N

Specifies that input message integrity and the consistency of output messages with associated database updates is not required. DCLWA does not apply to response mode or Fast Path input processing. If it is specified in these situations, it will be ignored during IMS execution.

► IMPORTERR=ABORT | CONTINUE

Specifies which action to take if an error occurs during automatic import processing due to an invalid resource or descriptor definition. This parameter does not pertain to errors that

occur while trying to access an RDDS; use the RDDSERR= parameter to choose processing options following RDDS-related errors:

- ABORT

Automatic import processing is aborted, and IMS cold start terminates abnormally with a U3397 abend. ABORT is the default.

- CONTINUE

Automatic import processing continues. The resource in error is marked and given a not-initiated status (NOTINIT). If the resource is required, create or update the resource after the cold start completes.

- RDDSERR=ABORT | NOIMPORT

Specifies which action to take if an error occurs when a resource definition data set is accessed during automatic import processing:

- ABORT

Automatic import processing is aborted, and an IMS cold start terminates abnormally with a U3368 abend. ABORT is the default.

- NOIMPORT

Automatic import processing is aborted. IMS cold start processing continues. The IMS system is started with no defined resources in the IMS.MODBLKS data set.

- RDDSDSN=(*dsname1, dsname2,...dsnamen*)

RDDSDSN= specifies the 1- to 44-character names of the system resource definition data sets. The data sets must be cataloged BSAM data sets. The data sets are dynamically allocated and opened in the order in which they are specified on the RDDSDSN= parameter. Each IMS in the IMSplex must have its own set of system resource definition data sets. At least two data set names must be specified. Although any number of data sets can be specified, three RDDSs are recommended. When exporting to a system RDDS, IMS rotates between each of the data sets specified. All resource and descriptor definitions for the local IMS are exported to the data set specified on the RDDSDSN= parameter that contains the oldest data. When importing from a system RDDS, IMS uses the RDDS with the most current data.

IMS always attempts to write to the RDDS containing the oldest data. If a failure occurs while IMS attempts to write to the data set with the oldest data, the data set with the next oldest data is selected. The RDDS with the most current data is preserved. For example, if two RDDS data sets are defined, and RDDS1 contains the most current resource and descriptor definitions, automatic export attempts to write to RDDS2 first. If that fails, automatic export does not attempt to write to RDDS1. The automatic export function is suspended until the error is resolved.

Common Service Layer section

In IMS Version 10, a Common Service Layer (CSL) section was created in the DFSDFxxx PROCLIB member that is used for defining CSL parameters. A new MODBLKS= parameter is required that specifies whether you will use DRD or IMS online change to manage your resource definitions.

Note: This new CSL section can replace the separate DFSCGxxx PROCLIB member entirely. If parameters are defined in both members, the DFSCGxxx member will take precedence over the DFSDFxxx member.

To enable DRD, you must specify MODBLKS=DYN in the CSL section of the DFSDFxxx PROCLIB member (or in the DFSCGxxx PROCLIB member). Here are the possible values for this parameter, with OLC as the default:

► MODBLKS=OLC | DYN

Specifies whether resources in the IMS.MODBLKS data set are defined dynamically or by online change. These resources include databases, programs, routing codes, and transactions. This attribute can only be changed across an IMS cold start. If the value is changed for the next IMS warm or emergency restart, IMS restart terminates, and abend U0168 is issued:

– DYN

Enables dynamic definition for resources in the IMS.MODBLKS data set. Resource definitions can be dynamically added, changed, or deleted using online commands, including CREATE, DELETE, and UPDATE. When you specify MODBLKS=DYN, you cannot online change resources in the IMS.MODBLKS data set with a /MODIFY PREPARE MODBLKS or INITIATE OLC PHASE(PREPARE) TYPE(MODBLKS) command. When MODBLKS=DYN, UPDATE commands that update runtime values (such as a transaction class) are permitted, unlike online change, which has this restriction.

– OLC

Enables online change for resources in the IMS.MODBLKS data set. When you specify MODBLKS=OLC, you can add, change, or delete resource definitions using online change. Online change commands for resources in the IMS.MODBLKS data set include /MODIFY PREPARE MODBLKS or INITIATE OLC PHASE(PREPARE) TYPE(MODBLKS).

Important: Although you should define all of the DRD parameters that we have just reviewed, the key migration parameters for DRD enablement are:

► CSL section of DFSDFxxx or DFSCGxxx:

- MODBLKS=DYN

► DRD section of DFSDFxxx:

- AUTOEXPORT=AUTO
- AUTOIMPORT=AUTO
- RDDSDSN=(*rddsname1,rddsname2,...rddsnamen*)

Automatic import and export

When you cold start IMS for the first time with DRD enabled, the RDDS will be empty. If you specify AUTOIMPORT=AUTO in the DFSDFxxx PROCLIB member, IMS will attempt to read the RDDS, find that it is empty, and then attempt to read the MODBLKS data set. Assuming that IMS finds the stored resource definitions, it will then read them into the IMS control region where they will become runtime resource definitions.

Importing resource and descriptor definitions

With DRD enabled, you can create runtime resource definitions online dynamically by importing the resource definitions from either the IMS.MODBLKS data set or an RDDS. You can create runtime descriptor definitions during cold start processing by importing the descriptor definitions from an RDDS. To import resource definitions during cold start processing from:

- ▶ The IMS.MODBLKS data set, specify `AUTOIMPORT=MODBLKS` in the `DYNAMIC_RESOURCES` section of the `DFSDFxxx` member in the `IMS.PROCLIB` data set.
- ▶ An RDDS, modify the `DYNAMIC_RESOURCES` section of the `DFSDFxxx` member in the `IMS.PROCLIB` data set in the following ways:
 - Specify `AUTOIMPORT=RDDS` or `AUTOIMPORT=AUTO` to enable automatic export.
 - Define a set of system RDDSs using the `RDDSDSN()` parameter.

For IMS to automatically import definitions from an RDDS, one or more of the RDDSs must contain a valid set of resource and descriptor definitions. For the details about parameters and restrictions on the `AUTOIMPORT= keyword`, see *IMS Version 10: System Definition Reference*, GC18-9966.

Also, ensure that you specify `AUTOEXPORT=AUTO` in the `DFSDFxxx` `PROCLIB` member so that at the next system checkpoint, IMS will automatically export the runtime resource definitions to the RDDS. At the next cold start, IMS will then be able to read the now populated RDDS to obtain the resource definitions.

Exporting resource and descriptor definitions

You can keep the stored resource and descriptor definitions in the RDDS synchronized with the runtime resource and descriptor definitions that IMS is using by occasionally exporting (storing) the resource and descriptor definitions to the RDDS. To automatically export the resource and descriptor definitions to the RDDS:

1. Define two or more resource definition data sets to IMS using the `RDDSDSN()` parameter in the `DYNAMIC_RESOURCES` section of the `DFSDFxxx` member in the `IMS.PROCLIB` data set.
2. Specify `AUTOEXPORT=AUTO` or `AUTOEXPORT=RDDS` in the `DYNAMIC_RESOURCES` section of the `DFSDFxxx` `IMS.PROCLIB` member.

At checkpoint time, resource and descriptor definitions are exported to the RDDS containing the oldest data, if the definitional attributes of one or more resources has changed since the last checkpoint. Information about the parameters and restrictions associated with the `AUTOEXPORT= keyword` is in *IMS Version 10: System Definition Reference*, GC18-9966.

6.3.2 Select OM applications to issue commands

DRD commands must be issued through applications that utilize the Operations Manager Application Programming Interface (OM API). Part of the migration process is selecting which applications to use in your DRD operations for your MODBLKS resource management. Examples of such applications are the TSO Single Point of Control (SPOC), the batch SPOC, the REXX SPOC API, and the IMS Control Center. You are not limited to using a particular application, but can use any of them at any time.

6.3.3 Cold start IMS with DRD enabled

Cold start IMS, specifying `DFSDF=xxx` in the control region execution parameters, where `xxx` identifies the suffix of the `DFSDFxxx` member in the `IMS.PROCLIB` data set that IMS will use for this startup. IMS loads the resource definitions from the `IMS.MODBLKS` data set, because the defined `RDDS`s are empty.

The `IMS.MODBLKS` data set is used only the first time that you implement DRD, except for fallback purposes. IMS then uses the resource definitions to create the internal control blocks needed to manage the resources. When the initial checkpoint is taken, the resource definitions are automatically written out to one of the defined resource definition data sets (`RDDS`) because `AUTOEXPORT=AUTO` is specified.

Note: After IMS starts with DRD enabled, you will no longer be able to use the online change process to add, change, or delete database, application program, route code, or transaction resources or resource definitions. Also, because `AUTOIMPORT=AUTO` is specified and the `RDDS`s are no longer empty, IMS loads its resource definitions from the `RDDS` with the most current data the next time that IMS cold starts.

6.4 Using Dynamic Resource Definition

DRD can be used by directly issuing commands through an OM API or by using an application called Manage Resources (MR) that is included with IMS Version 10. We now review the DRD commands syntax and usage and the MR application.

6.4.1 Descriptors

A *descriptor* is a model for defining (creating) a resource or another descriptor, which establishes defaults for attributes not set when it is created. When creating a resource or descriptor, you can omit setting the attribute values and they can be obtained from the specified descriptor.

IMS Version 10 comes with four system default resource descriptors and their definitions (one for each resource type) that contain the IMS system default values. These descriptors and their definitions cannot be deleted or modified. The four IMS-supplied resource descriptors are:

- ▶ `DFSDSDB1` (database descriptor)
- ▶ `DFS DSPG1` (application program descriptor)
- ▶ `DBFDSRT1` (Fast Path routing code descriptor)
- ▶ `DFS DSTR1` (transaction descriptor)

You can use the IMS-supplied descriptors as models for creating resources or additional descriptors. You can also create descriptors without modelling them after existing descriptors. Initially, the IMS-supplied descriptors are set as the system default descriptors, but you can designate one of your own descriptors as the system default by using the `DEFAULT(Y)` parameter.

When you create a new resource that is modelled from a descriptor (using the `LIKE` parameter), the new resource inherits the attributes from the descriptor for any attributes that you do not specify on the `CREATE` command. If there are attributes not defined in your resource descriptor, these values are taken from the IMS-supplied descriptor.

Similarly, when you create a new resource using an existing resource as a model (using the LIKE parameter), the new resource inherits the attributes from the existing resource.

6.4.2 DRD commands

The following DRD commands can be used to manage resource definitions for your transactions, application programs, databases, and routing codes:

- ▶ CREATE
- ▶ DELETE
- ▶ UPDATE
- ▶ QUERY

The CREATE and DELETE commands are new in IMS Version 10, whereas the UPDATE and QUERY commands existed prior to IMS Version 10 but were enhanced in this latest version to support the DRD capability. Here, we review the function of each DRD command, as well as the syntax for the commands that are new in IMS Version 10. This information can also be found in *IMS Version 10: Command Reference, Volume 1*, SC18-9700.

Note: Some parameters have different defaults in DRD compared to IMS system generation:

- ▶ The database access default changes from ACCESS=EX to ACCTYP(UP).
- ▶ The application schedule type default changes from SCHDTYP=SERIAL to SCHDTYPE(PARALLEL).
- ▶ The transaction commit mode default changes from MODE=MULT to CMTMODE(SNGL).

CREATE

Use the CREATE command to create a resource definition or descriptor definition. Example 6-2 contains the command syntax for the CREATE command.

Example 6-2 Command syntax for the CREATE command

```
CREATE (resource type) | (descriptor type)
NAME (name1,name2,...)
LIKE(RSC(resource name)) | LIKE(DESC(descriptor name))
SET(attribute1(value1),attribute2(value2),...)
DEFAULT(Y) <<< valid for descriptors only
```

To determine the correct keyword resource or descriptor type to specify for the CREATE command, see Table 6-3.

Table 6-3 Resource and descriptor definition keywords

Resource type	Resource type keyword	Descriptor type keyword
Transaction	TRAN	TRANDESC
Application program	PGM	PGMDESC
Database	DB	DBDESC
Routing code	RTC	RTCDESC

We now review the other command keywords in a bit more detail:

► **NAME**

The names of the resource or descriptor that you are creating, which will all have the same attributes as each other.

► **LIKE**

Specifies that the resource is created using an existing resource or descriptor as a model, instead of the default descriptor. The default descriptor is either the IMS descriptor or is user-defined. The model type can either be a descriptor (DESC) or a resource (RSC). The new resource is created with all of the same attributes as the model. Attributes set explicitly by the CREATE command override the model attributes. Later changes to the model are not propagated to resources or descriptors that were created from it.

► **SET**

Specifies the attributes of the resource to be created. If the LIKE keyword is omitted, attributes not specified take the value defined in the default descriptor, which is either the IMS or user-defined descriptor. If the LIKE keyword is specified, attributes not specified are drawn from the values defined in the model.

► **DEFAULT(Y)**

Applies only when you are creating a descriptor. Specifying it makes the descriptor the current system default for a particular resource definition.

As you can see, attribute values can be set by a variety of methods. The attribute values specified on the SET keyword will take precedence over all other methods. The next method in the hierarchy is the LIKE keyword. Attributes not set with the SET keyword will be gotten from the model specified on the LIKE keyword. The model can be either a resource or a descriptor. Finally, if neither the SET nor LIKE keywords are included on the CREATE command, the attribute values will be drawn from the current system default descriptor, which can be either an IMS or user-defined descriptor.

See Figure 6-3 for an example of the CREATE command being used to create two programs while setting their scheduling type to serial and specifying that transaction level statistics should be logged. In this example, you can see that one program resource was not created, because a program with the specified name already exists. Even though this occurred, the other program was created successfully.

```

File Action Manage resources SPOC View Options Help
-----
PLEX1          IMS Single Point of Control
Command ==> CREATE PGM NAME(PGMIMS1,PGMIMS2) SET(SCHDTYPE(SERIAL)TRANSTAT(Y))
-----
Plex . . PLEX1 Route . . Wait . . 5:00
Response for: CREATE PGM NAME(PGMIMS1,PGMIMS2) SET(SCHDTYPE(SE...
PgmName  MbrName  CC
PGMIMS1  IMS1      0
PGMIMS2  IMS1      0

```

Figure 6-3 Example of the CREATE command being used to create the program resource definition

DELETE

Use the DELETE command to create a resource definition or descriptor definition. Example 6-3 contains the command syntax for the DELETE command.

Example 6-3 Command syntax for the DELETE command

```

DELETE (resource type) | (descriptor type)
      NAME(* | name1,name2,...namen) OPTION(ALLRSP)

```

To determine the correct keyword to use for the resource or descriptor type, see Table 6-3 on page 145.

Here is a description for each of the other DELETE command keywords:

► **NAME**

NAME specifies the 1-8 character resource names that are to be deleted. It can also be a wildcard character of an asterisk (*), which should be used with caution because it will delete all resources of the specified type:

- Example: NAME(*), NAME(PGM123*)

► **OPTION**

ALLRSP is the only valid value that you can specify for this parameter, and it returns a command response for each resource that the command processes. It is valid only when NAME(*) is specified. If you do not include OPTION(ALLRSP) as part of the command when NAME(*) is specified, you will see a response line of output only for those resources that had an error. This parameter is optional.

See Figure 6-4 for an example of the DELETE command.

File	Action	Manage resources	SPOC	View	Options	Help

PLEX1		IMS Single Point of Control				
Command ==> _						

		Plex . .	PLEX1	Route . .		Wait . . 5:00
Response for: DELETE PGM NAME(PGMIMS1,PGMIMS2)						
PgmName	MbrName	CC				
PGMIMS1	IMS1	0				
PGMIMS2	IMS1	0				

Figure 6-4 Example of the DELETE command being used to delete two programs

You cannot delete a resource that is in use, such as a transaction with messages queued to it or a database that is being referenced by a program. To determine whether a particular resource has work in progress, you can query the resource using the SHOW(WORK) parameter. You can see an example of this command in Figure 6-6 on page 148. To prevent new work from coming in and making the resource “in use” and therefore unavailable, we recommend that you stop the resource with the /STOP command before attempting to delete it.

UPDATE

The UPDATE command was enhanced in IMS Version 10 to support DRD. Using this command, you can also update MSC definitions: MSPLINK, MSLINK, and MSNAME. Note that MSC definitions cannot be created or deleted.

Just like the DELETE command, you cannot update a resource that is in use, such as a transaction with messages queued to it or a database that is being referenced by a program. You can query the resource using the SHOW(WORK) parameter to determine whether a particular resource has work in progress. To prevent new work from coming in and making the resource “in use”, we recommend that you stop the resource with the /STOP command before attempting to update it.

If you want to update both the status and attributes of a resource, you must do so in separate UPDATE commands. DRD does not support updating both of these in one command.

You can also update an existing descriptor to be the current system default by specifying DEFAULT(Y) on the UPDATE command. Any resource that is created thereafter that does not have the SET or LIKE keywords specified will obtain its attribute values from your system default descriptor.

See Figure 6-5 for an example of the UPDATE command being issued to update three transactions' message types to "single segment," which means that incoming messages to these transactions will be one segment in length.

File	Action	Manage resources	SPOC	View	Options	Help

PLEX1		IMS Single Point of Control				
Command ==> UPDATE TRAN NAME(ADDINV,ADDPART,AOP) SET(MSGTYPE(SNGLSEG))_						

		Plex . .	PLEX1	Route . .	Wait . . 5:00	
Response for: UPDATE TRAN NAME(ADDINV,ADDPART,AOP) SET(MSGTYPE...						
Trancode	MbrName	CC				
ADDINV	IMS1	0				
ADDPART	IMS1	0				
AOP	IMS1	0				

Figure 6-5 Example of the UPDATE command changing three transactions' message types

QUERY

The QUERY command has been enhanced in IMS Version 10 to support the DRD process. Specifically, the SHOW(WORK) parameter was added that shows any work in progress that would prevent a DRD command from completing successfully for a given resource. Also, there is now QUERY support for the program and routing code resource definitions.

Before attempting to delete or update a resource definition, you should issue the QUERY command with SHOW(WORK) specified to determine whether the resource is in use. When you have determined that there is no work in progress for the resource, you can stop the resource and proceed with the delete or update. See Figure 6-6 for an example of the QUERY command's output.

```

File Action Manage resources SPOC View Options Help
-----
PLEX1                      IMS Single Point of Control
Command ==> _

----- Plex . . PLEX1 Route . . Wait . . 5:00
Response for: QUERY DB NAME(*) SHOW(WORK) More: +

DBName AreaName MbrName CC TYPE Work
AUTODB          IMS1      0 DL/I
AUTODDBH        IMS1      0
BANKATMS        IMS1      0
BANKFNCL        IMS1      0
BANKLDGR        IMS1      0
BANKTERM        IMS1      0
BE2PCUST        IMS1      0 DL/I
BE3ORDER        IMS1      0 DL/I
BE3ORDRX        IMS1      0 DL/I
BE3PARTS        IMS1      0 DL/I
BE3PSID1        IMS1      0 DL/I
BIBDBD          IMS1      0 DL/I
BIBIDBD         IMS1      0 DL/I
CDCCUSD         IMS1      0 DL/I
CDCCUSX         IMS1      0 DL/I
CDCPROD         IMS1      0 DL/I

```

Figure 6-6 Example of a QUERY DB command with SHOW(WORK) specified

6.4.3 Importing and exporting resource definitions

DRD comes with a capability to automatically import and export your stored resource definitions to and from a Resource Definition Data Set (RDDS). You can find supporting documentation for these two functions in the *IMS Version 10: System Definition Reference*, GC18-9966.

AUTOIMPORT

AUTOIMPORT is a parameter that you specify in the DFSDFxxx PROCLIB member that indicates whether resource and descriptor definitions should be automatically imported during the IMS cold start. Let us take a look at each possible value that you can specify for this parameter:

► **AUTOIMPORT=AUTO**

This is the default parameter setting. IMS will determine whether it should enable automatic import processing. If IMS enables automatic import, it also determines the data source from which it will import the definitions. IMS will automatically import your stored resource definitions either from the RDDS data set or the MODBLKS data set assuming that no read errors occur. Let us now explore the circumstances under which each of these scenarios would occur.

IMS will automatically import the stored resource definitions from an RDDS data set if all of the following conditions are satisfied:

- You have two or more RDDSs defined in your DFSDFxxx PROCLIB member.
- The RDDSs can be allocated and read.
- At least one of your RDDSs contains valid resource and descriptor definitions.

IMS will automatically import the stored resource definitions from a MODBLKS data set if the data set exists and if all of the defined system RDDSs are either empty or not defined in the DFSDFxxx PROCLIB member. IMS will not import any definitions if both of the following conditions are true:

- Two or more RDDSs are defined, but an error occurs while IMS is trying to access them.
- No RDDSs are defined, and no MODBLKS data set is defined.

► **AUTOIMPORT=MODBLKS**

IMS will import your stored resource definitions from the MODBLKS data set if one is defined. When this occurs, the values specified for the IMPORTERR= and RDDSERR= parameters do not apply and are ignored if specified. If an error occurs during the import process, IMS cold start is stopped.

► **AUTOIMPORT=NO**

IMS will not import any resource or descriptor definitions during cold start.

► **AUTOIMPORT=RDDS**

IMS will import your stored resource and descriptor definitions from the most recently updated RDDS.

AUTOEXPORT

AUTOEXPORT is a parameter that you specify in the DFSDFxxx PROCLIB member that indicates whether all resource and descriptor definitions should be exported at checkpoint time (simple or normal shutdown). The only instance in which automatic export will occur is if there have been definition changes made to any resource or descriptor since the time of the last checkpoint, except for restart checkpoint. After IMS completes restart processing, a restart checkpoint is taken. Export occurs after the restart checkpoint if automatic export is

enabled. The only data set to which IMS can export the resource and descriptor definitions is the RDDS. Let us take a look at each possible value that you can specify for this parameter:

► **AUTOEXPORT=AUTO**

This is the default parameter setting. IMS will determine whether it should enable automatic export processing. Automatic export is enabled if two or more RDDSs are defined and accessible.

► **AUTOEXPORT=NO**

No resource or descriptor definitions are exported at checkpoint time due to automatic export being disabled.

► **AUTOEXPORT=RDDS**

IMS will export all of the stored resource and descriptor definitions to the oldest RDDS at system checkpoint, as long as there are at least two defined and accessible RDDSs.

6.4.4 Recovering runtime resource definitions when using DRD

The runtime resource and descriptor definitions exist for the life of an IMS system or until you delete them. Let us say that you are using DRD to dynamically create your runtime resource definitions and one of your IMS systems abnormally ends. How do you recover the definitions that you have created, updated, and deleted thus far? The answer is that it depends on the type of restart that you are performing for this IMS system. We now look at each scenario.

Cold start

To recover your runtime resource definitions across a cold start, you must have previously exported them to your RDDS and automatically import them during this restart process. The definitions that have not been exported to the RDDS before the IMS system was shut down will not be recovered across a cold start.

Warm start and emergency restart

IMS will read the IMS log (also known as the *online log data set* (OLDS)) to obtain your runtime resource definitions during a warm start or emergency restart. When you use DRD, your runtime resource definitions are written to the IMS log every time that they are created, updated, deleted, and also at every system checkpoint.

6.4.5 RDDS Extraction utility (DFSURDD0)

You might feel more comfortable implementing DRD if you can have a backup copy of your stage 1 source code generated from IMS SYSGEN that matches your online IMS control region. Or you might be interested in a way to fold DRD into your existing operations. In either case, the RDDS Extraction utility can assist you. The RDDS Extraction utility is an offline batch utility that can be used to convert the stored resource definitions in an RDDS into IMS stage 1 macro statements or IMS type-2 CREATE commands. You can find supporting documentation about this utility in the *IMS Version 10: System Utilities Reference*, SC18-9968. Let us now take a look at the input to and output from this utility and review how it works.

Input and output

To use the RDDS Extraction utility, the input must be an RDDS data set that contains IMS resource and descriptor definitions. The RDDS must contain valid data from a successful export operation. When you run the utility, the generated stage 1 macro statements and

CREATE commands are written to the SYSOUT data set. Messages issued by the utility are written to the SYSPRINT data set.

When the utility converts resource definitions to stage 1 macro statements, you will get DATABASE, APPLCTN, RTCODE, and TRANSACT macro statements that represent the resource definitions in the RDDS as output. Descriptor definitions are not converted into stage 1 macro statements.

If you choose to have the utility convert your resource definitions to IMS type-2 CREATE commands instead of stage 1 macro statements, you will receive any of the following CREATE commands as output:

- ▶ CREATE DB
- ▶ CREATE DBDESC
- ▶ CREATE PGM
- ▶ CREATE PGMDESC
- ▶ CREATE RTC
- ▶ CREATE RTCDESC
- ▶ CREATE TRAN
- ▶ CREATE TRANDESC

Invoking the RDDS Extraction utility

The RDDS Extraction utility executes as a standard operating system job consisting of JCL statements. In this JCL, you should include a JOB statement, an EXEC statement, and DD statements defining the input and output. To see a sample, see Example 6-4 on page 152. Let us now examine the control statements in closer detail.

EXEC statement

The format of the EXEC statement is:

```
//S1      EXEC PGM=DFSURDD0,MEMLIMIT=4G
```

The RDDS Extraction utility obtains 64-bit storage. The MEMLIMIT parameter must be specified on the EXEC statement to ensure the total size of usable virtual storage above the bar is adequate. MEMLIMIT should be set to 4 G or higher.

DD statements

The following DD statements must be included in the RDDS Extraction utility JCL:

- ▶ **RDDSDSN DD**
Defines the system resource definition data set (RDDS) that is used as input to the utility. This DD statement is required. Only one data set can be specified for the RDDS. You cannot specify concatenated data sets.
- ▶ **SYSOUT DD**
Defines the data set that receives the output generated by the utility. This DD statement is required. The DCB parameters for the SYSOUT data set are LRECL=80, RECFM=FB.
- ▶ **SYSPRINT DD**
Defines the data set that receives the messages generated by the utility. This DD statement is required. The DCB parameters for the SYSPRINT data set are RECFM=FB, LRECL=133.
- ▶ **SYSIN DD**
Defines the source of the input parameters for the utility. This DD statement is required. The DCB parameters for the SYSIN data set are RECFM=FB, LRECL=80. For example:

```
//SYSIN DD *
```

Output format specification statement

You can use the OUTPUT parameter to specify the format of the output that is generated by the utility. This parameter is required. The following are valid values for the OUTPUT parameter:

- ▶ **MAC**
Indicates that the data in the RDDS is to be converted into IMS stage 1 macro statements
- ▶ **CMD**
Indicates that the data in the RDDS is to be converted into type-2 CREATE commands

Example 6-4 Example JCL to run the RDDS Extraction utility (DFSURDD0)

```
//job name JOB CLASS=J,MSGCLASS=A,MSGLEVEL=(1,1)
//JOB LIB DD DSN=library data set name,DISP=SHR
//S1 EXEC PGM=DFSURDD0,MEMLIMIT=4G
//RDDSDSN DD DSN=RDDS data set name,DISP=SHR
//SYSOUT DD DSN=output data set name,DISP=(,CATLG,DELETE),
// UNIT=SYSDA,VOL=SER=Volume name,
// SPACE=(CYL,(1,1),RLSE),
// DCB=(LRECL=80,RECFM=FB,BLKSIZE=800)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
OUTPUT=MAC
/*
```

Return codes

The RDDS Extraction utility can return one of several return codes after it is run. See Table 6-4 for the possible return codes, which you can also find in the *IMS Version 10: System Utilities Reference*, SC18-9968.

Table 6-4 RDDS Extraction utility return codes

Code	Meaning
0	The utility completed successfully.
4	The utility completed successfully; however, either no trans were defined for an application or no routing codes were defined for an IMS Fast Path (FP) exclusive application.
8	An error occurred trying to open the SYSPRINT data set.
12	An error occurred trying to open the SYSOUT data set.
16	An error occurred trying to read the SYSIN data set.
20	An error occurred trying to parse the SYSIN data.
24	An error occurred trying to obtain storage using the MVS GETMAIN macro.
28	An error occurred trying to obtain a buffer from 64-bit storage.
32	An RDJFCB request failed to return the dsname of the RDDS data set.
36	More than one data set is defined for the RDDS.
40	An error occurred trying to open the RDDS data set defined on the RDDSDSN DD statement.
44	An error occurred trying to read the RDDS data set defined on the RDDSDSN DD statement.

Code	Meaning
48	The data set defined on the RDDSDSN DD statement is not an RDDS.
52	The RDDS defined on the RDDSDSN DD statement does not contain data from a successful export operation.
56	An error occurred trying to start the BPE Limited Function Services (LFS).

6.4.6 Manage Resources application

IMS Version 10 comes with an application called *Manage Resources* (MR), which guides you through issuing DRD commands. You can invoke MR from either the TSO SPOC application or from the IMS Application Menu by selecting the Manage Resources option, as can be seen in Figure 6-7. If you have experience with the TSO SPOC, you are already familiar with MR's look and feel because it is a TSO SPOC enhancement. MR alleviates your need to memorize the syntax of different DRD commands, as well as the valid attribute names and values. Additional information about the Manage Resources application can be found in *IMS Version 10: Operations and Automation Guide*, SC18-9716.

<u>Help</u>	
IMS Application Menu	
Command ===>	
Select an application and press Enter.	
1	Single Point of Control (SPOC)
2	Manage resources
3	Knowledge-Based Log Analysis (KBLA)
4	HALDB Partition Definition Utility (PDU)
5	Syntax Checker for IMS parameters (SC)
6	Installation Verification Program (IVP)
7	IVP Export Utility (IVPEX)
8	IPCS with IMS Dump Formatter (IPCS)
9	Abend Search and Notification (ASN)

Figure 6-7 A screen capture of the IMS Application Menu including the MR application option

MR is a series of ISPF panels that guide you through the DRD process, allowing you to issue DRD commands against your resource definitions and descriptor definitions by presenting blank fields about the definitions that you fill in. Using MR, you can create, delete, update, and query the resource definitions supported by DRD:

- ▶ Databases
- ▶ Application programs
- ▶ Transactions
- ▶ Routing codes

Before you can use MR to issue DRD commands, you must set two MR preferences under the Options, then Preferences menu. Here, you must designate the MR view that you prefer, as well as whether you want to see a confirmation of the command that you are about to issue before it is sent to IMS. Less experienced operators, who require a higher assistance level, should select List view, whereas more highly-skilled operators specify Command syntax view. We will discuss these options in more detail later. You can see an example of this panel in Figure 6-8 on page 154.

Help	
SPOC	IMS Single Point of Control Preferences
Command ===> _____	
Select your options and press the Enter key.	
	More: -
	2. Keep command responses.
	3. Erase command responses.
Manage Resources Preferences	
Resource definition	
dialog layout 1	1. List view.
	2. Command syntax view.

Figure 6-8 The preferences that must be set before using MR to use DRD

After you start the MR application, you are presented with a panel such as the one in Figure 6-9. From this panel, you can select the function that you want to perform on your resource definitions.

Help	
PLEX1	IMS Manage Resources
Command ===> _____	
Select an action and press Enter.	
* Action _	1. Create new resources
	2. Delete resources
	3. Query resources
	4. Update resources
	5. Import resources
	6. Export resources

Figure 6-9 Function selection panel in the IMS MR application

You are then presented with a panel that prompts you to enter a resource type, name, and whether it is a resource or descriptor. There is an exception for the Delete function, which will not show this second panel when used in Command syntax view, which we will discuss later in the section entitled “Command syntax view” on page 156. You can see an example of this panel in Figure 6-10 on page 155, where we are creating a transaction resource using the system default descriptor as the model from which to obtain the attribute values.

File	Action	Manage resources	SPOC	View	Options	Help

PLEX1	IMS Create New Resource					
Command ==>						
----- Plex Route Wait						
Select a resource. To base a resource on a template, specify the template information. Press Enter to continue.						
More: +						
* Resource	4	1. Database 2. Program 3. Routing Code 4. Transaction				
Resource name	1	1. Resource 2. Descriptor				
Resource type	1	1. System default 2. Existing resource				
Definition template	1	1. System default 2. Existing resource				
Existing resource Definition template name						

Figure 6-10 The Manage Resources application being used to create a transaction resource

The MR application supports beginning and advanced skill levels by providing two views. These views are called *List view* and *Command syntax view*. We now explore these views while also observing an example creation of a transaction definition using the MR application.

List view

The list view of MR is appropriate for operators who have little experience with IMS, because it contains panels that help build the DRD command. Therefore, knowing the command syntax is unnecessary. You can see a list view example of MR being used to dynamically create a transaction resource named IMSTRAN1 in Figure 6-11.

File	Action	Manage resources	SPOC	View	Options	Help

PLEX1	IMS Create Transactions					
Command ==>						
----- Plex Route Wait						
Press Enter to continue						
More: +						
* NAME	Transaction name . . . IMSTRAN1					
SET						
AOCMD	AOI command option				N	CMD, N, Tran, Y
CLASS	Class				1	1-999
CMTMODE	Commit mode				SINGL	Sngl, Mult
CONV	Conversational				N	Y, N
DCLWA	Log write-ahead option				Y	Y, N
DIRROUTE	MSC direct routing option				N	Y, N
EDITRTN	Input edit routine					
EDITUC	Edit to uppercase				Y	Y, N
EMHBSZ	EMH buffer size					12-30720
FP	Fastpath processing option				N	N, E, P
INQ	Inquiry option				N	Y, N
LCT	Limit count				65535	1-65535
LPRI	Limit priority				1	0-14
MAXRGN	Maximum region count				0	0-999

Figure 6-11 List view of the Manage Resources application being used to create a transaction

Command syntax view

The command syntax view of MR is for more highly-skilled operators, because there is less guidance and often it only contains a single input panel from which DRD commands are issued. In command syntax view, MR creates the skeleton for the command syntax and you either fill in the values or overwrite the default values before issuing the command. You can find an example of MR in command syntax view in Figure 6-12, in which we are once again creating a transaction resource named IMSTRAN1.

File	Action	Manage resources	SPOC	View	Options	Help

PLEX1		IMS Create Transactions				
Command ==>						

		Plex . .		Route . .		Wait . .
Press Enter to continue						
CREATE TRAN NAME(IMSTRAN1)						
SET(AOCMD(N) CLASS(1) CMTMODE(SNGL) CONV(N) DCLWA(Y)						
DIRROUTE(N) EDITRTN() EDITUC(Y) EMHBSZ()						
FP(N) INQ(N) LCT(65535) LPRI(1) MAXRGN(0) MSGTYPE(MULTSEG)						
MSNAME() NPRI(1) PARLIM(65535) PGM()						
PLCT(65535) PLCTTIME(6553500) RECOVER(Y) REMOTE(N) RESP(N)						
SEGNO(0) SEGSZ(0) SERIAL(N) SIDL(10) SIDR(10)						
SPASZ() SPATRUNC() TRANSTAT(N) WFI(N)						

Figure 6-12 Example of command syntax view in the Manage Resources application

6.5 DRD availability benefits summary

Dynamic Resource Definition improves IMS availability by allowing dynamic creation, deletion, and updating of resource definitions. It reduces the requirements for planned outages and eliminates unavailability associated with MODBLKS OLC.

DBRC enhancements

In this chapter, we introduce several DBRC enhancements in IMS Version 10:

- ▶ Parallel RECON access, which eliminates the serialization of multiple DBRC instances against one set of RECONs
- ▶ RECON update support to add to the QUERY support introduced in IMS Version 9
- ▶ DBRC timestamp precision improvement to recognize microseconds
- ▶ RECON READONLY access to improve security granularity

In addition to these enhancements, we have already described fuzzy user image copy support in Chapter 3, “Database enhancements” on page 65.

7.1 Parallel RECON access

DBRC Parallel RECON access is introduced in IMS Version 10 as an option to reduce:

- ▶ RECON contention
- ▶ Transaction response time issues
- ▶ Unplanned IMS system quiesces during online log data set (OLDS) switches
- ▶ Growth constraints for IMS data sharing users
- ▶ Growth constraints for single IMS system users with many batch jobs sharing the RECON data set

Parallel RECON access does this by enabling concurrent, parallel access to the RECONs by multiple DBRC instances.

It is implemented for IMS Version 10 DBRC by exploiting DFSMS Transactional VSAM (TVS) to provide record level locking, buffering, and logging for the RECON data set. We show you the differences with and without parallel RECON access by reviewing the technique used without it first and then discussing the implementation of it.

7.1.1 Non-parallel RECON access

Without parallel RECON access, the RECON data integrity is maintained by serializing access of multiple DBRC instances. Performance is optimized by allocating and opening the RECONs when the DBRC subsystem starts. Dynamic allocation is preferred. This is typically at batch job start, utility start, or online IMS start. Figure 7-1 illustrates the multiple access environment.

For every DBRC access request to the RECONs, the action is serialized and the buffers are purged. The buffer contents might have been invalidated by some other DBRC's updates of the RECONs.

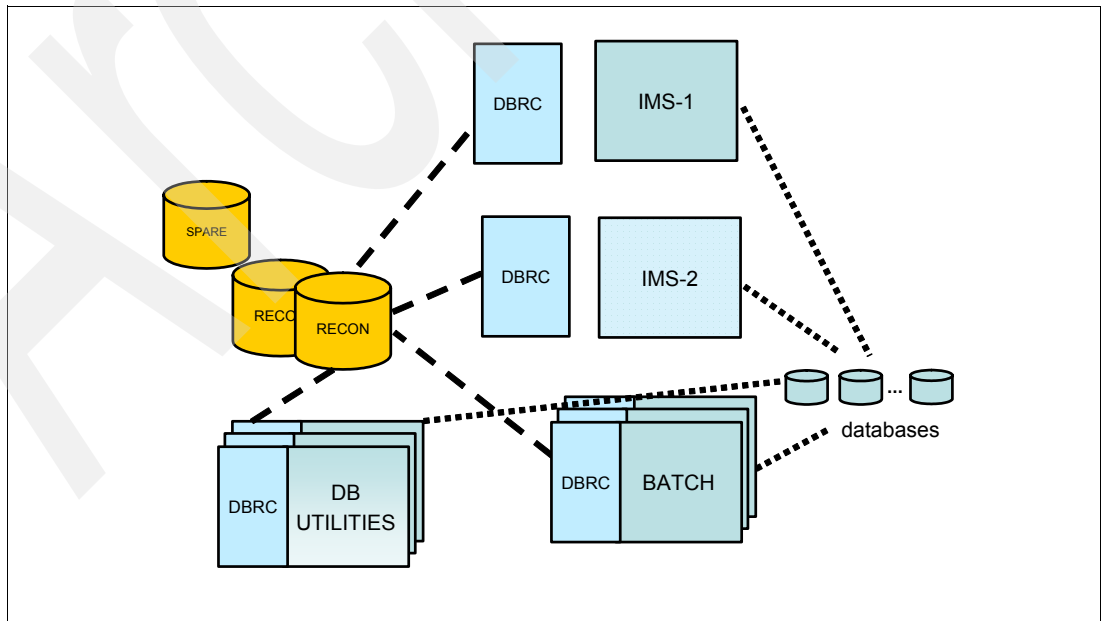


Figure 7-1 Multiple DBRC accessors

Serial access uses the z/OS RESERVE macro service with resource DSPURI01. Batch jobs serialize on another resource first (DSPURI02), which prevents the online systems from being locked out when a series of batch jobs is submitted.

In a Global Resource Sharing (GRS) “star” environment, you should convert the hardware RESERVEs to a global enqueue. If you have a GRS “ring” environment, then the hardware RESERVEs should remain.

7.1.2 Parallel RECON access

Parallel RECON access eliminates this serialization and reduces contention for the RECONS. It could provide better responsiveness, especially in situations where multiple online, batch, or utility executions of IMS are making many requests to the RECONS.

Parallel RECON access uses Transactional VSAM (DFSMStvs), a system facility that is provided by DFSMS. Transactional VSAM provides facilities, such as locking, logging, caching, and commit coordination for VSAM data sets for data sharing equivalent to IMS Block Level Data Sharing. DFSMStvs requires a Parallel Sysplex, including a Coupling Facility (CF). Parallel RECON access requires these even when all of the DBRC instances are running in only one z/OS system.

There is a licensing fee associated with DFSMStvs, but “special bids” will be considered for IMS clients using the parallel RECON access function who do not already have DFSMStvs to acquire DFSMStvs for use restricted to IMS (contact your IBM marketing representative for details).

You use Resource Recovery Services (RRS) for DFSMStvs. RRS is used for commit coordination and is familiar to many IMS users, for example, with APPC or ODBA. You do not need to request RRS for IMS; the RRS=Y IMS execution parameter is not required.

Because parallel RECON access allows RECON requests to be processed in parallel, it potentially reduces RECON contention. However, service times for individual requests might increase due to the overhead of locking, logging, and Coupling Facility accesses that are required. Certain service times might be decreased due to the caching of RECON information in buffer pools and the CF because there is no buffer purging at the start of each DBRC request.

You must provide new operational procedures for recoveries in a parallel RECON access environment. There are new possibilities for failures when the DFSMStvs environment is used compared with native VSAM used by DBRC without parallel RECON access.

We have stated that parallel RECON access is optional in IMS Version 10 and is specified for a set of RECONS. There is a DBRC command for specifying parallel access for a set of RECONS. Some RECON sets use parallel access while others continue to use serial access. Essentially, you choose parallel RECON access for a given data sharing group.

7.1.3 DFSMStvs

The origins of DFSMStvs (TVS) are in data sharing for CICS/VSAM files. There is a list of publications in “Related publications” on page 317 that describe DFSMStvs and its Record Level Sharing (RLS) and VSAM bases. You can start with *z/OS DFSMStvs Planning and Operating Guide*, SC26-7348, *z/OS DFSMStvs Administration Guide*, GC26-7483, and *DFSMStvs Presentation Guide*, SG24-6973.

TVS is based on VSAM Record Level Sharing (RLS), which uses locking and data caching to provide data sharing capabilities for VSAM files used by CICS systems. It allows multiple CICS systems to update VSAM files concurrently. RLS relies on the logging, commit coordination, and backout processing provided by CICS online systems. TVS adds its own logging, commit coordination, and backout support, which allows batch update jobs to share VSAM files between each other and CICS. Parallel RECON access uses this same capability to allow multiple DBRC instances to do concurrent updates to the RECONs.

RLS and TVS execute in the SMSVSAM address space. This is a system address space that is typically started at z/OS initialization. There is only one SMSVSAM address space per logical partition (LPAR) as we illustrate in Figure 7-2. It provides RLS and TVS services to all users in the LPAR. This includes all instances of DBRC, although we do not show them all in this figure.

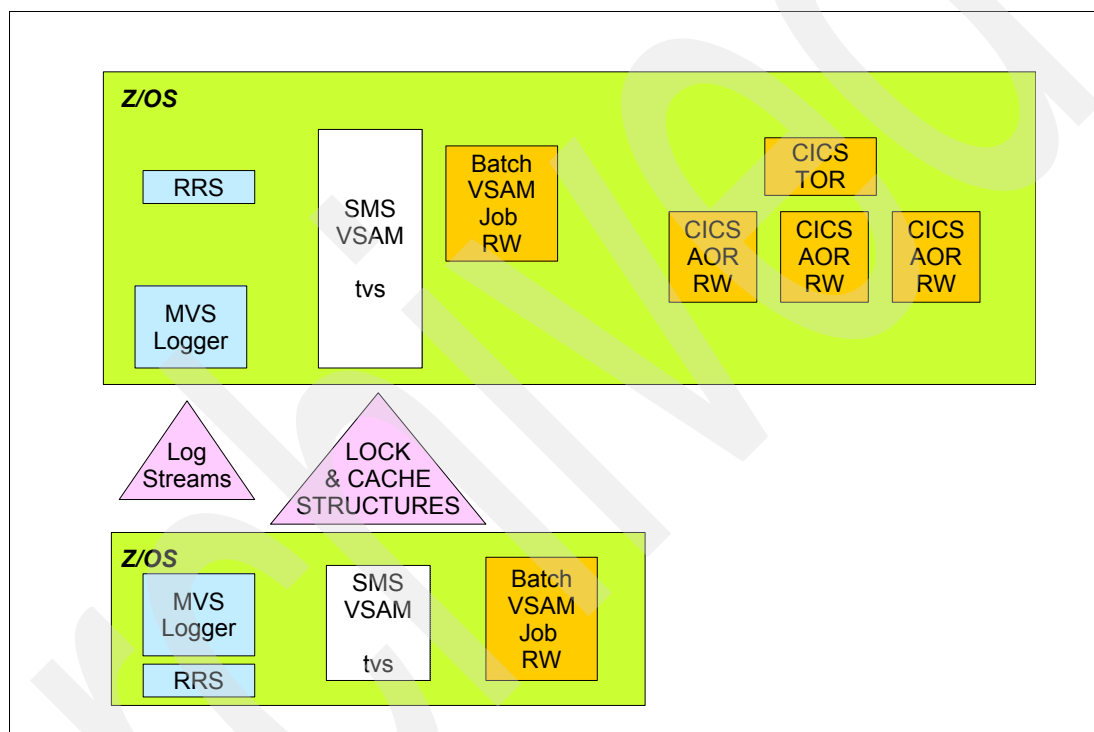


Figure 7-2 DFSMStvs and address spaces

We show a DBRC-only version in Figure 7-3 on page 161 with a DFSMStvs environment and multiple instances of DBRC using parallel RECON access. There is an SMSVSAM address space in both z/OS systems. This address space provides TVS services to the DBRCs in the LPAR. The illustration shows that there is an IMS online system, an IMS batch job (DLI or DBB), and an IMS utility in each system. TVS uses the system logger and RRS. RRS also uses the system logger. The system logger has structures in the Coupling Facility. The SMSVSAM address space also connects to the Coupling Facility for its own structures. These are cache structures and a lock structure. There is a pair of share control data sets (SHCDS) for the VSAM data sets that contain information about the use of the VSAM data sets that are being shared. There is also a spare SHCDS. Of course, there are the RECON pair and spare.

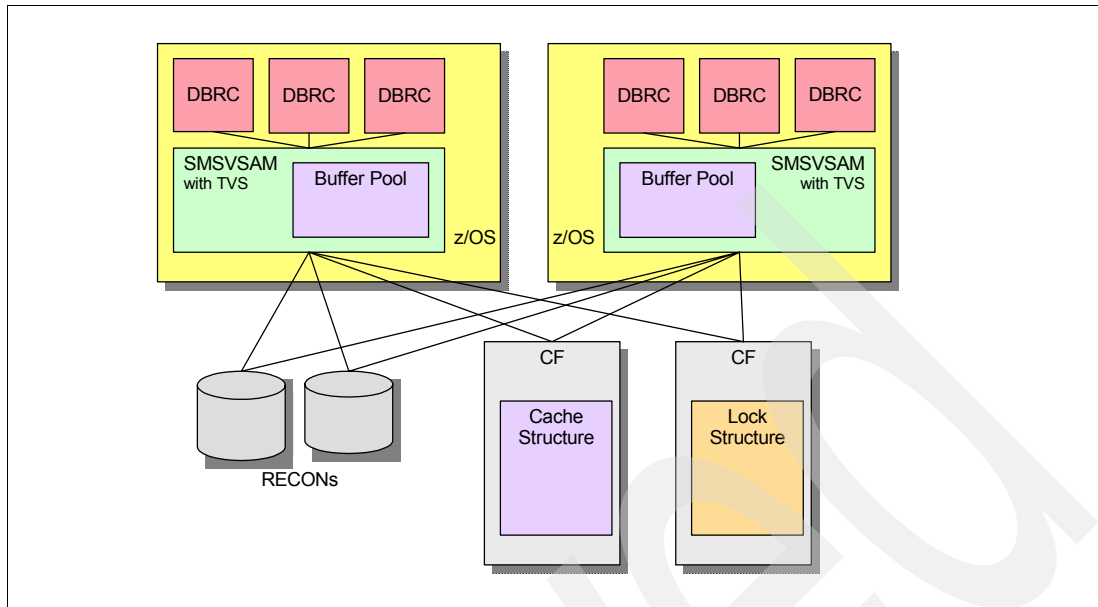


Figure 7-3 Multiple DBRCs and TVS

DFSMSStvs has recovery capabilities to handle various types of failures. Each DFSMSStvs instance has its own undo log. These are log records, which have the *before images* of records:

- ▶ If a user of DFSMSStvs, such as DBRC, fails, its uncommitted updates are backed out by DFSMSStvs. The undo log is used for this purpose.
- ▶ If DFSMSStvs fails or the SMSVSAM address space fails, it is automatically restarted. When it restarts, it backs out any in-flight work and releases any locks held by the in-flight transactions.
- ▶ If a z/OS system fails, peer recovery is invoked. Backouts are done for the failed DFSMSStvs by using another SMSVSAM address space in the Parallel Sysplex. The locks held for the failed work are released.

More information about these processes, when you use TVS for DBRC parallel RECON access, is provided in 7.1.7, “Parallel RECON access recovery” on page 177.

When a DBRC instance accesses a VSAM record in the RECONs, the request is processed by TVS in the SMSVSAM address space. The SMSVSAM address space has its own buffers in buffer pools maintained in data spaces owned by the SMSVSAM address space. When using TVS, DBRC does not have its own buffers for RECON processing. Instead, it uses the SMSVSAM buffers. These buffers are shared with all users of the SMSVSAM address space. These are all of the VSAM data sets in the z/OS system that are using RLS or TVS. SMSVSAM also uses cache structures in the Coupling Facilities. Each shared data set is assigned to a structure. When a VSAM record in an SMSVSAM buffer pool is updated and it resides in buffers in other systems, those buffers are invalidated. This is done using the cross invalidation capability associated with cache structures and Parallel Sysplex.

The processing of VSAM requests includes lock requests. SMSVSAM has its own lock manager. It does not use the IRLM; however, its lock manager provides functions similar to those provided by IRLM for IMS and DB2 databases. The owner of a lock request is the DBRC instance, either an MS online system, an IMS batch job, or an IMS utility. Lock information is held in a lock structure in the CF.

7.1.4 Parallel RECON access definition and setup

To implement parallel RECON access, you must create or modify several specifications, which we describe:

- ▶ IMS definitions
- ▶ RECON definitions
- ▶ SHCDS data set
- ▶ SYS1.PARMLIB member definitions
- ▶ Log stream definitions
- ▶ Structure definitions
- ▶ RACF specifications
- ▶ IMSplex preparation

IMS definitions

Parallel RECON access requires IMS Version 10. To ensure that all users of a set of RECONS are using IMS Version 10, you must specify a MINVERS value of '10.1' for the RECONS. This prevents any lower release level of IMS from using the RECONS.

SCI registration is required for parallel RECON access. This means that you must provide a CSL environment with an SCI address space for each system where an IMS using parallel RECON access will be run. Because SCI registration is required, we recommend that you use a DBRC SCI Registration Exit routine (DSPSCIX0). This ensures that consistent registration is done by all DBRC users, and it eliminates the need for you to add the IMSPLEX= parameter to the JCL for online systems, batch jobs, and utilities. In addition to the IMSPLEX value, IMS Version 10 has added the DBRC group ID. Activation of parallel RECON access requires the CHANGE.RECON IMSPLEX() command to set both the IMSPLEX and DBRC group ID. Use of the IMSPLEX EXEC parameter or DBRC group ID before this step will result in message DSP1136A and subsequent job failure due to unavailable RECON data sets.

The DBRC SCI Registration exit routine interface now contains:

- ▶ A RECON Data Set Name
- ▶ An IMSplex name return area: A 5-byte area in which DSPSCIX0 returns an IMSplex name
- ▶ The IMSPLEX EXEC parameter value: If specified as an EXEC parameter
- ▶ The parameter list version (which is new)
- ▶ A DBRC Group ID return area (new): A 3-byte area set by DSPSCIX0
- ▶ The DBRCGRP EXEC parameter value (new): If specified as an EXEC parameter

The sample exit routine has been updated to support the group ID.

DBRC I/O exit routine

If you use the DBRC I/O exit routine, it must be updated. Some of the meanings of the fields used by the exit routine have changed. A Request Token (a STCK for parallel access) replaces the RESERVE Sequence Number and a DBRC Instance Token, which can be used to qualify the Request Token for parallel access in case of duplicate STCK values, and is passed on every exit invocation. The Instance Token is zeroes for serial RECON access. The Changed Record Count is always zero for parallel access.

There are new exit invocations with parallel access:

- **Commit exit call**

Invoked when the previous updates for the current series have been committed to the RECON data set

- **Backout exit call**

Invoked when the previous updates for the current series have been backed out

Multiple instances of the I/O exit can be running in parallel, which can cause a different behavior if the exits write to common data sets.

You cannot use the Request Token/Changed Record count to determine if updates were missed while using parallel RECON access, because backouts are performed by TVS and are not passed to the exit. If this matters, you should save updates until commit before externalizing them (and discard them if a backout has occurred).

The primary users of the exit routine are products that track changes to the RECONS. Few, if any, installations write their own DBRC I/O exit routines.

DBRC group ID

If multiple RECON sets will be used in the same IMSplex, you must use the DBRC group ID with parallel access. Different RECON sets in the same IMSplex must have different DBRC group IDs. These IDs are used to distinguish which DBRCs use a set of RECONS when there are multiple RECON sets being used within an IMSplex. We illustrate this in Figure 7-4 on page 164. When upgrading RECONS from a previous release, the group ID defaults to 001.

We also recommend using the DBRC group ID for RECONS using serial access when automatic RECON loss notification (ARLN) is used. The IDs ensure that a reconfiguration of the RECONS is only processed by the DBRCs using that set of RECONS.

If multiple RECONS are used in the same IMSplex, the plexname and group ID for a set of RECONS is set by issuing the `CHANGE.RECON IMSPLEX()` command to set or change the values for the set of RECONS. Failure to use the `CHANGE.RECON IMSPLEX()` command will result in message DSP1136A and subsequent job failure due to unavailable RECON datasets.

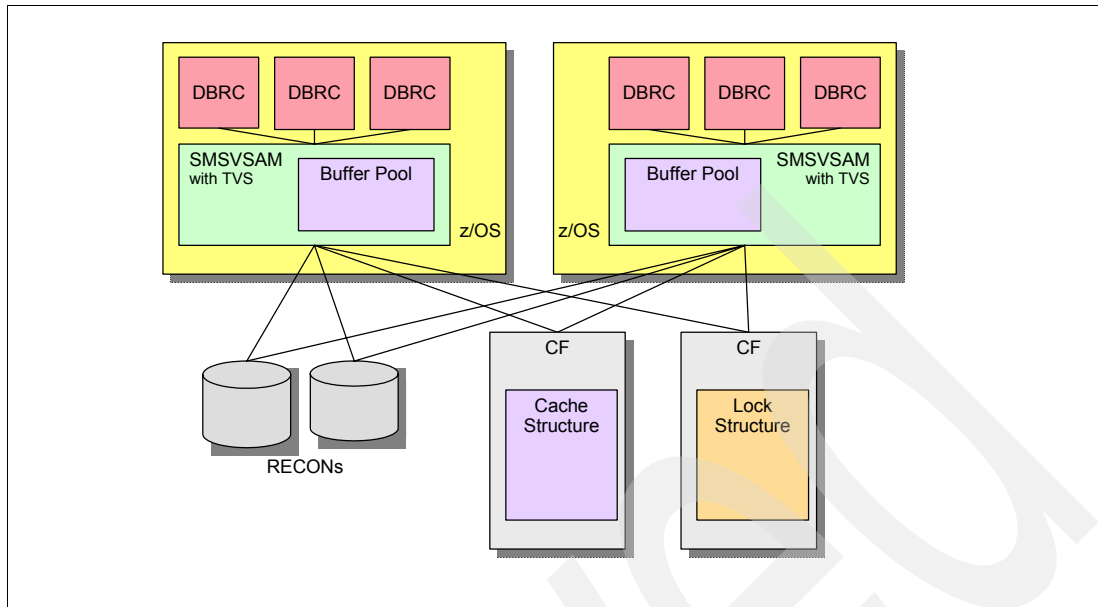


Figure 7-4 RECON GROUP ID

RECON definitions

You must manage the RECONs with SMS for parallel RECON access. That means you define the RECONs with a storage class that has a cache set value. A cache set value assigns CF cache structures to the storage class. Many data sets (spheres) can have the same storage class. Multiple storage classes can use the same cache set. You can simplify the management and understanding of caching for the RECONs by assigning the RECON data sets to a storage class with its own cache set. The cache set needs only one structure defined to it. If the RECONs are the only RLS or TVS data sets in the storage class, they will have their own cache structure.

The data class that you use for the RECONs determines the caching option. This is controlled by the RLSCFCACHE parameter for the data class:

- ▶ ALL specifies that all CIs read from DASD are also stored in the structure.
- ▶ UPDATESONLY specifies that CIs that are updated are written to the structure when they are written to DASD.
- ▶ NONE specifies that no CIs are stored in the structure. When NONE is specified, the structure is used only for buffer invalidation processing.

There is another parameter that affects caching. This is the RLS_MAXCFFEATURELEVEL parameter on the IGDSMSxx member of SYS1.PARMLIB. We will explain this when we discuss IGDSMSxx later in "SYS1.PARMLIB".

Transactional VSAM requires you to use LOG(UNDO) in the definition of the RECONs. This creates a backout log for the data set when TVS is used. DBRC automatically sets this definition. If you use parallel RECON access and you did not specify LOG(UNDO) for the RECONs, DBRC will issue an ALTER to set the parameter to LOG(UNDO). If you do not use parallel RECON access, DBRC will ALTER RECON to LOG(NONE) if another value is defined. Of course, you can set LOG(UNDO) by issuing the DEFINE or ALTER with IDCAMS. In either case, the ALTER requires ALTER authority from the security system, such as RACF.

Do not specify LOG(ALL) or a LOGSTREAMID parameter for the RECONS. LOG(ALL) creates a forward recovery log that is identified by the LOGSTREAMID. DBRC does not support forward recovery of the RECONS.

IFAPRDxx

DFSMSStvs is a separately priced feature of z/OS. You must enable it with a product enablement policy that is defined in the IFAPRDxx member of SYS1.PARMLIB as we show in Example 7-1.

Example 7-1 DFSMSStvs enablement in IFAPRDxx

```
PRODUCT OWNER('IBM CORP')
  NAME('Z/OS')
  ID(5694-A01)
  VERSION(*) RELEASE(*) MOD(*)
  FEATURENAME(DFSMSSTVS)
  STATE(ENABLED)
```

SHCDS

If you have not implemented RLS, you must define the SHCDS data sets. If you already have implemented RLS, you have these data sets defined. These data sets are used to hold the name of the lock structure, the list of subsystems, and a list of open VSAM spheres (data sets). It also contains information about the spheres, such as which subsystems have them open.

You can have multiple SHCDS data sets. It is typical to have two SHCDS data sets. You can also define spares. Having a pair and a spare provides recovery capabilities similar to those for a pair and spare for the RECONS.

The data set name for an SHCDS is always of the form:

SYS1.DFPSHCDS.*qualifier.vvolser*

where *qualifier* is whatever you want it to be and *vvolser* is the volume serial for its volume. See the *DFSMSdfp Storage Administration Reference*, SC26-7402, for complete information about defining SHCDS data sets.

You use VARY commands to make a data set an active SHCDS or a spare as we show in Example 7-2.

Example 7-2 SHCDS activation

```
V SMS,SHCDS(qualifier.vvolser),NEW
V SMS,SHCDS(qualifier.vvolser),NEWSPARE
```

SYS1.PARMLIB

You must define SMS to z/OS as a subsystem by including a statement for SMS in the IEFSSNxx SYS1.PARMLIB member.

You define the names associated with each TVS instance in the SYSNAME parameter of the IGDSMSxx member. See Example 7-3 on page 166 for that and other parameters. Similarly, you give each TVS instance a unique number with the TVSNAME parameter. The IGDSMSxx member of SYS1.PARMLIB is described in the *z/OS MVS Initialization and Tuning Guide*, SA22-7591.

Example 7-3 IGDSMSxx parameters

```
SYSNAME(name1,name2, ...)
TVSNAME(n1,n2, ...)
AKP(n1,n2,...)
RLSINIT(YES|NO)
RLS_MAX_POOL_SIZE(nnnn)
RLS_MAXCFFEATURELEVEL(A|Z)
DEADLOCK_DETECTION(nnnn,mmmm)
MAXLOCKS(nnnn,mmmm)
```

AKP controls the occurrence of activity key points. You can specify separate numbers for the different systems. At these points, log records that are no longer needed for backouts are deleted. Log records for units of recovery that have not logged in two AKPs are moved to the shunt log.

If RLSINIT(YES) is not specified, you must start the SMSVSAM address space with a V SMSVSAM,ACTIVE command.

In RLS_MAX_POOL_SIZE, you specify the maximum size of the VSAM local buffer pool in SMSVSAM in megabytes. This pool is created in a data space. For serial access to the RECONS (not parallel RECON access), IMS uses the DSPBUFFS specification to determine the number of buffers used by a DBRC instance. This defaults to 60 index buffers and 120 data buffers. You might have specified a different value. As a starting point, you can make the SMSVSAM local buffer pool the size of the buffer pools for all of the concurrent DBRC instances that will be run. For example, if you took the default DSPBUFFS, have a RECON CI size of 16 K, and have 10 concurrent DBRC instances, you can start with a buffer pool of 10 x (120 + 60) x 16 K = 29 MB. You should maintain DSPBUFFS in the RESLIB if you might turn off parallel access.

You use the RLS_MAXCFFEATURELEVEL parameter to specify whether CIs larger than 4 K will be cached in the SMSVSAM cache structure or not. "Z" specifies that only 4 K and smaller CIs will be cached. "A" specifies that all can be cached.

You use DEADLOCK_DETECTION(nnnn,mmmm) to specify the number of seconds in a local deadlock detection cycle and the number of local cycles in a global cycle. These default to 15 seconds and 4 cycles.

You use MAXLOCKS to control the issuing of warning messages about the number of locks held. These are messages IGW859I and IGW10074I, which we describe in "Monitoring locks" on page 174.

You can specify RLSTMOUT, but DBRC overrides this value. This is the time that a lock request will wait before it times out. DBRC always uses a value of 2 seconds.

You use CF_TIME to specify at what interval SMF type 42 records containing information and statistics on CF cache and lock structures are gathered.

You can use the SETSMS command to change parameters that were specified in the IGDSMSxx member:

SETSMS AKP(nnn,nnn,nnn,...)	Changes activity keypoint values
SETSMS DEADLOCK(localsecs,globalcycles)	Changes deadlock detection cycle parameters
SETSMS MAXLOCKS(max,incr)	Changes the thresholds for the IGW859I and IGW10074I messages

Log streams

DFSMStvs maintains an *undo* log stream for each instance of SMSVSAM. This means that there is one undo log stream per z/OS system using DFSMStvs. The log stream contains *before* images of updates made by users of this DFSMStvs instance. It is used to back out any updates that have been done by units of recovery that fail or are victims in a deadlock. The log stream name is IGWTV nnn .IGWLOG.SYSLOG where nnn is the TVSNAME that you assigned in the IGDSMSxx member of SYS1.PROCLIB.

Typically, backout records do not need to be maintained for a long time. When the unit of recovery is committed, these log records can be deleted. However, there are occasions when the log records need to be kept for a long time. This occurs if a backout failed due to the inability to do the backout updates to the data set or if a unit of recovery was not committed for a long time. In these cases, log records can be moved to a shunt log stream. Each instance of DFSMStvs has a shunt log stream with the name IGWTV nnn .IGWSHUNT.SHUNTLOG where nnn is the TVSNAME.

Log records are moved from the undo log to the shunt log by activity keypoint processing. The frequency of this processing is determined by the AKP parameter that you specify in the IGDSMSxx member. If a unit of recovery has not logged for two AKPs, its log records are moved to the shunt log.

The use of the shunt log allows TVS to trim the undo log. TVS tracks the oldest log entry for any running unit of recovery. It can trim records that are older than this. This limits the size of the undo log and tends to keep all of its records in the log structure.

Each request to DBRC is a unit of recovery that is tracked and logged by RRS. RRS has multiple log streams that are used for various purposes. DFSMStvs is only one of several components that use RRS. Other IMS users of RRS include APPC, ODBA, and OTMA. Most installations already have RRS log streams defined.

RRS has five log streams that are shared by all systems in the sharing group:

Resource Manager data log	Contains information about resource managers using RRS services
Restart log	Contains information about incomplete units of recovery
Main UR state log	Contains the state of active units of recovery
Delayed UR state log	Contains the state of delayed active units of recovery
Archive log (optional)	Contains information about completed units of recovery

You must define log streams in the LOGR policy for a sysplex. Each log stream is defined with a DEFINE LOGSTREAM statement. The policy assigns the log stream to a structure. The CFRM policy defines the size of the structure.

The definition that we show in Example 7-4 on page 168 includes some of the other parameters that you might define. The LS_SIZE parameter defines the size of the data used for offloading the log stream from the structure. The size is specified in 4 K blocks. STG_DUPLEX(YES) specifies that the log stream is duplexed in staging data sets when the condition defined in the DUPLEXMODE parameter is satisfied. DUPLEXMODE(COND) specifies that duplexing occurs when the CF structure is in a volatile CF or when the CF and the SMSVSAM address space reside in the same machine (CPC). The HIGHOFFLOAD parameter specifies the percentage of the structure that must be full before an offload process occurs. Offload moves log records from the structure to an offload data set. LOWOFFLOAD specifies the percentage of the log stream that can remain in the structure when an offload process completes. All parameters that you can specify in a LOGR policy are documented in *z/OS MVS Setting Up a Sysplex*, SA22-7625.

Example 7-4 LOGSTREAM definition

```
DEFINE LOGSTREAM
  NAME(IGWTV001.IGWLOG.SYSLOG)
  STRUCTNAME(TVSLOG_TV001_SYSLOG)
  LS_SIZE(1180)
  STG_DUPLEX(YES)
  DUPLEXMODE(COND)
  HIGHOFFLOAD(85)
  LOWOFFLOAD(15)
```

Because DBRC *transactions* that update the RECONs should be short-lived, you should define small undo logs for DBRC. The log records are deleted by AKP. In addition, the log records for long-lived transactions will be moved to the shunt log by AKP.

SMSVSAM lock structure

SMSVSAM uses a lock structure. You must provide it even when there is only one system using RLS or TVS. The lock structure holds locks for all of the data sets using RLS and TVS. This can include data sets other than RECONs so you must size it to hold all of these locks.

The lock structure name is always IGWLOCK00. If you specify the size of the lock structure as a power of 2, it is equally divided between the lock table and the record list. If it is not a power of two, the size of the lock table is the largest power of 2 that is smaller than half the structure size. The remaining space is used for the record list.

The lock structure is automatically rebuilt after a failure of the lock structure, a failure of its CF, or a loss of connectivity to its CF. If the SMSVSAM address spaces survive, the locks are repopulated in the rebuilt structure. DBRC will survive this recovery. This applies to all instances of DBRC including online systems, IMS batch jobs, and IMS utilities using DBRC.

System-managed duplexing can be used for the lock structure. Duplexing is not recommended when the structure is placed on a machine that does not contain an SMSVSAM address space connected to the structure. The overhead of duplexing is unlikely to be justified in this case, because recovery is done without the loss of any users.

Locking of VSAM records with parallel RECON access

When parallel RECON access reads the RECON, record locks are taken in DFSMStvs. The locks are either at a share level or an exclusive level. As the levels suggest, multiple DBRC instances can hold the lock for a record at the same time if they are all at the share level. There can be only one holder of a lock for a record at the exclusive level. VSAM get requests use share level locks. VSAM get-for-update requests use exclusive level locks. We provide details of monitoring lock usage in “Monitoring locks” on page 174.

Exclusive level locks are held until commit while share level locks can be released before commit. Commit occurs when the DBRC request from IMS or the DBRC command has completed. Typically, share level locks are held until commit. When you use the CONCURR option for a LIST command, the locks are released as part of the read process. When you use STATIC for a LIST command, the locks are held until commit. See 7.1.5, “LIST.XXX output” on page 172 for a more detailed discussion of LIST options. Sometimes, DBRC browses RECON records of a type looking for a specific record. When this occurs, locks are obtained and released as each record is read. When the specific record is found, the lock is obtained again and held until commit.

A deadlock can occur between requestors any time locking is in progress. The deadlock detection process of SMSVSAM is similar to that done by IRLM. There is a local cycle and a

global cycle. During the local cycle, deadlocks within a SMSVSAM are detected. During the global cycle, deadlocks between units of recovery using different SMSVSAMs are detected. The local cycle uses a waiters list in the detection process. At each cycle, a list is built of those units of recovery that are waiting. If the same wait exists in two successive cycles, the locks are examined to see if a deadlock exists. When a deadlock is found, a victim is chosen. Unlike the IRLM implementation, there is no *worth* value associated with units of recovery. A victim is chosen without regard to the type of DBRC instance in which it is executing. The unit of recovery (UOR), which is the victim in a deadlock, is backed out and its request is reprocessed.

An IGW10072I message is sent to the console if a deadlock occurs. It is followed by IGW10073I messages. An IGW10073I message is issued once for each DBRC in the deadlock chain and includes the record that DBRC holds and the record for which it is waiting. We show the messages in Example 7-5. Deadlocks are retried indefinitely. This is no limit to the number of retries for them.

Example 7-5 VSAM RLS deadlock messages

```
IGW10072I jobname stepname urid VSAM RLS DETECTED A DEADLOCK.
THERE ARE nn UNITS OF RECOVERY IN THE DEADLOCK CHAIN
```

```
IGW10073I {UNIT OF RECOVERY urid | SUBSYSTEM NAME subsys
TRANSACTION ID tranid} RUNNING IN JOB jobname
HOLDS {ADD TO END LOCK | EXCLUSIVE LOCK ON KEY | SHARED LOCK ON KEY} ON BASE
CLUSTER NAME cluster [ALTERNATE INDEX NAME altindex]
AND IS WAITING FOR {ADD TO END LOCK | EXCLUSIVE LOCK ON KEY | SHARED LOCK ON KEY}
ON BASE CLUSTER NAME cluster2 [ALTERNATE INDEX NAME altindex2]
[HOLDING KEY VALUE = key1 WAITING KEY VALUE = key2]
```

An IGW10070I message is sent to the console if a timeout occurs. It is followed by an IGW10071I message for each holder of the lock. We show these messages in Example 7-6.

Example 7-6 VSAM RLS deadlock detection timeout

```
IGW10070I jobname stepname urid A REQUEST TIMED OUT WAITING FOR A LOCK.
THERE ARE nn UNITS OF RECOVERY HOLDING THIS LOCK.
```

```
IGW10071I {UNIT OF RECOVERY urid | SUBSYSTEM NAME subsys
TRANSACTION ID tranid} RUNNING IN JOB jobname
HOLDS {ADD TO END LOCK | EXCLUSIVE LOCK ON KEY | SHARED LOCK ON KEY}
IN BASE CLUSTER NAME cluster [PATH NAME path] CAUSING {TRUE | FALSE} CONTENTION.
[KEY VALUE = key]
```

After a timeout, a random delay is made, before a retry, to resolve situations where two requestors wait on each other multiple times. Message DSP1184W is sent to indicate that five retries have been attempted. Retries continue, and you need to take no action. We show the text of the message in Example 7-7.

Example 7-7 VSAM RLS message at five retries

```
DSP1184W VSAM ACCESS ERROR ENCOUNTERED 5 TIMES RC=0008 RSN=0022
```

If a timeout for the same lock request occurs 15 times for a unit of recovery, the DSP1185A message is sent after the backout of the UOR. The message is a WTOR and you must reply to the message. You might be able to determine which DBRC instance is holding the lock. If so, you can issue the RETRY reply after the holder of the lock has completed its work. We

show the text of the message in Example 7-8. The RSN=0022 in the DSP1184W and DSP1185A messages indicates that the reason for the error is a timeout for a lock request.

Example 7-8 VSAM RLS message at 15 retries

```
DSP1185A VSAM ACCESS ERROR ENCOUNTERED 15 TIMES RC=0008 RSN=0022 - REPLY 'RETRY'  
OR 'CANCEL'
```

As we described previously, a lock timeout or a deadlock results in the backout of the unit of recovery and the reprocessing of the DBRC request. The backout applies to the updates to the RECONS. It does not apply to any other output data sets, such as SYSPRINT or the JCLOUT data set used for GENJCL commands. Timeouts and deadlocks can produce duplicate outputs to these data sets. DBRC recognizes the possibility of these occurrences by issuing the DSP1186I message when a command is retried. This warns the user that there might be duplicate output from the command. You might find duplicate output to be a problem with GENJCL and LIST commands. It can occur for other commands, but the consequences are less serious. For example, the SYSPRINT output of a CHANGE.DB command can contain repeated lines.

The explanation of the DSP1186I message is:

An error that could be retried was detected. DBRC is attempting to reprocess a command that might have produced external output. DBRC retries the processing, which encountered errors, that it considers capable of being retried (for example, deadlock or timeout). In this instance, data might have been written to a data set (for example, SYSPRINT) or the JCLOUT data sets. Since command processing is redriven, output produced prior to this message should be ignored. Output to a JCLOUT data set that is submitted directly to an internal reader may produce duplicate JCL or JCL that does not run.

SMSVSAM cache structures

SMSVSAM uses one or multiple cache structures. You assign data sets to structures through their storage classes. You can assign a set of structures to a storage class. The data sets in the storage class use the structures defined for the storage class. This allows you to set aside a structure for the exclusive use of the RECONS. By doing this, you can size a structure to meet the needs of the RECONS. As a starting point, you can make the structure large enough to hold all of the data that was previously in the DBRC buffers. These are defined with DSPBUFFS.

When a storage class has multiple cache structures, data sets are assigned to a structure when they are opened. SMSVSAM attempts to balance the use of structures in a storage class.

The SMSVSAM cache structures are store-through structures, which means that all of the CIs that are written to the structure are also written to the RECON data sets. All committed updates are in the data sets.

If a Coupling Facility containing an SMSVSAM cache structure fails, the structure is automatically rebuilt on another CF. When the structure is rebuilt, it is not repopulated. All of the buffers containing CIs assigned to the structure are invalidated. This is similar to the actions that IMS takes when a full function cache structure is lost. If a structure cannot be rebuilt but the storage class has one or more other structures, the data sets using the failed structure are reassigned to another structure. DBRC survives the rebuilding of structures and the reassignment of data sets to other structures. These failures and recoveries are masked from DBRC by SMSVSAM.

System-managed duplexing is not supported for SMSVSAM cache structures. It is not needed, because the loss of a structure is easily handled by the rebuild done by SMSVSAM.

Logger structures

Log structures are required for the DFSMSStvs undo and shunt logs and for the RRS log stream.

Parallel RECON access will typically not require much space in these structures. Parallel RECON access *transactions* typically make few updates to the RECONS and do not wait for non-DBRC work. This tends to make its space requirements small. If a structure is not large enough, records are offloaded to logger data sets. This offloading depends on the size of the structure, the sizes of the log streams using the structure, and the HIGHOFFLOAD and LOWOFFLOAD parameters in the log stream definition in the LOGR policy.

Security (RACF) considerations

DBRC issues some DFSMS commands for recovery situations. These commands require update authority to the RECONS and the STGADMIN.IGWSHCDS.REPAIR resource. Read authority is required for all users for the RECONS and the *REPAIR* resource. Update authority is required in certain recovery situations. If a RECON I/O error occurs, it is likely that update authority will be required for the recovery to the spare RECON. A lack of update authority results in the user failing. Obviously, it is good for no users to fail; however, it is probably essential that online systems not fail.

DBRC ALTERs the RECONS when switching from serial to parallel access or from parallel to serial access. This should not be a typical operation. The ALTER is used to change from LOG(NONE) to LOG(UNDO) or from LOG(UNDO) to LOG(NONE). ALTER authority is required for the DBRC jobs that issue the command. DBRC also issues an ALTER when reconfiguring the RECONS if the spare has the wrong setting for the LOG parameter. If the spare has been defined with the correct LOG value, the ALTER is not required.

Parallel RECON access implementation

You turn on parallel RECON access by specifying ACCESS(PARALLEL) on the CHANGE.RECON or INIT.RECON command as we show in Example 7-9. You turn it off with ACCESS(SERIAL). SERIAL is the default for the INIT.RECON command. If some non-DBRC region has the RECON opened in LSR or NSR mode when you try the CHANGE.RECON ACCESS(PARALLEL), you will see an open failure reason x'A8' and DSP1164 message. The access mode will not be changed.

Example 7-9 Enabling parallel RECON access

```
CHANGE.RECON ACCESS(SERIAL|PARALLEL)
INIT.RECON ACCESS(SERIAL|PARALLEL)
```

Because parallel RECON access requires DFSMSStvs, you must have activated a complete DFSMSStvs environment when you issue the command. For example, you must issue the command on a z/OS system with an SMSVSAM address space and have the lock and cache structures defined.

You must ensure a good pair of active RECONS and a spare RECON are available when the CHANGE.RECON ACCESS(PARALLEL) command is issued.

You cannot have an active RSR tracking subsystem when you change to parallel access.

RECON header and header extension processing

With serial access, the RECON header and header extension are read by DBRC at the start of each request for two reasons. First, it is read to determine the current RECON options, such as trace options and time format options. Second, the header is read to determine if there is a multiple update (MUP) record.

A MUP record is written before DBRC makes updates to multiple RECON records as part of one DBRC request. This is used to ensure that all of the updates are made, or if they are not made, that they are backed out. The MUP records are a dynamic log for DBRC and contain information sufficient for backing out updates to the RECONS. After the MUP record is written, DBRC makes the updates to the indicated RECON records. When these are completed, the MUP record is deleted. Each DBRC request first checks for a MUP record. If one exists, it indicates that another DBRC instance has failed in the middle of making multiple updates; remember that RECON access is serialized. The DBRC instance that reads the MUP record backs out the partial updates and deletes the MUP record. This mechanism provides integrity with serial access to the RECONS.

The header extension is read to determine the current RECON configuration. This is the current COPY1, COPY2, and spare data sets. They could have been changed by another DBRC instance since this DBRC last accessed the RECONS.

This processing differs with parallel RECON access. The header record is read by DBRC *logical open* but is locked with a shared lock except when updates are done to it. It is read to check on the current RECON options. Updates are rare, because they are typically done by a DBRC CHANGE.RECON command that you issue.

Parallel RECON access eliminates the need to read the RECON header extension for every DBRC request. The RECON header extension still contains the current RECON configuration information, but DBRC does not need to read the header for every DBRC request. If a DBRC instance reconfigures the RECONS, it uses SCI to communicate the new configuration to the other instances. The configuration information is obtained from the header extension when a new DBRC instance is initialized.

MUP records are not used with parallel RECON access. Backouts of incomplete updates are done by DFSMSStvs using its undo log stream. This does not eliminate the need to read the header for each DBRC request, because it must be read to find any changes to RECON options (for example, if a CHANGE.RECON TRACEON has been issued).

These changes in processing of the header extension record with parallel RECON access eliminate a reason for possible contention on the header extension record. It also reduces the number of I/Os to the RECONS required for most DBRC requests.

7.1.5 LIST.XXX output

The LIST commands have new options with parallel RECON access. They are implemented with the CONCURR, STATIC, and QUIESCE keywords, which are ignored when parallel RECON access is not being used. You set the LIST options with the CHANGE.RECON command, and they can be set even if ACCESS(SERIAL) is in effect so that it is ready when ACCESS(PARALLEL) is turned on. When RECONS are upgraded to IMS Version 10, the default is set to STATIC. The QUIESCE keyword cannot be set by default.

If you use the CONCURR keyword, updates can occur to the RECONS while the list is being produced. This might make the output inconsistent, because some of the records might be listed before a change and some after the change. For example, a database record might include an authorization to a subsystem but the subsystem record might not show the

database as authorized to the subsystem. When you use CONCURR, a lock for a record is held only to read the record.

If you use the STATIC keyword, the listing is consistent. For example, if a database record listing includes an authorization to a subsystem, the subsystem record will show the database as authorized to the subsystem. When you use STATIC without the QUIESCE keyword, parallel RECON access locks each record when it reads it and holds the lock for the duration of the list processing.

You can use the QUIESCE keyword with STATIC. It quiesces all other activity to the RECONS during list processing. Other DBRC instances cannot access the RECONS during this time. Because all other activity is quiesced, locking is not needed to provide integrity or consistency unless there are retained locks. If there are retained locks from a failed DBRC for which recovery has not been done, a lock is held while DBRC is positioned on a record. QUIESCE reduces the overhead of the LIST command, because it eliminates locking. You should consider using QUIESCE when a LIST command will read many RECON records. During the quiesce process, batch jobs and utilities that use DBRC wait even if they are not attempting to access the RECONS. For example, batch jobs perform no DL/I calls during the QUIESCE process.

In Example 7-10, we show a listing produced with the CONCURR option. We specified ACCESS(PARALLEL) for these RECONS, and we set the list default for the RECONS as STATIC. Near the middle of the example, there is a line of *ACCESS=PARALLEL LIST=STATIC*. This indicates that parallel RECON access is active for this RECON and that the default for LIST commands is STATIC. At the bottom of the example is *GROUP ID = GPB*. GPB is the DBRC group ID that we used for this set of RECONS.

Example 7-10 LIST.RECON CONCURR

```

LIST.RECON CONCURR
06.286 10:43:55.891200          LISTING OF RECON  (CONCURRENT)          PAGE 0002
-----
RECON
RECOVERY CONTROL DATA SET, IMS Version 10R1
DMB#=357                      INIT TOKEN=05306F1839312F
NOFORCER LOG DSN CHECK=CHECK44  STARTNEW=NO
TAPE UNIT=3490          DASD UNIT=SYSDA          TRACEOFF  SSID=**NULL**
LIST DLOG=NO              CA/IC/LOG DATA SETS CATALOGED=NO
MINIMUM VERSION = 10.1      CROSS DBRC SERVICE LEVEL ID= 00001
LOG RETENTION PERIOD=00.001 00:00:00.0
COMMAND AUTH=NONE  HLQ=**NULL**

ACCESS=PARALLEL  LIST=STATIC
SIZALERT DSNUM=15      VOLNUM=16      PERCENT= 95
LOGALERT DSNUM=3       VOLNUM=16

TIME STAMP INFORMATION:

TIMEZIN = %SYS

OUTPUT FORMAT:  DEFAULT = LOCORG NONE  PUNC YY
                CURRENT = LOCORG NONE  PUNC YY

IMSPLEX = PLEX1          GROUP ID = GPB

```

When you list a large number of the records using a command, such as LIST.RECON, you might get better performance using LIST(STATIC) QUIESCE because no locks will be taken. For a shorter listing, such as LIST.SUBSYS, a LIST.STATIC might be sufficient.

You should use LIST(CONCURRE) when possible for best performance overall, because the record is only locked while it is being printed.

7.1.6 Parallel RECON access performance considerations

Transactional VSAM access has extra overhead compared with the traditional LSR access used by DBRC in serial mode. Components of this overhead include:

- ▶ Cross-address space access to SMSVSAM server
- ▶ Overhead of CF cache access
- ▶ Overhead of record locking
- ▶ Overhead of logging

However, access to the RECON is no longer serialized across the entire sysplex and “logging” in the RECON (MUP records) is avoided.

As with IMSs Block Level Data Sharing environment, several factors will affect performance, including:

- ▶ SMSVSAM local buffer pool sizes
- ▶ Cache structure sizing for VSAM RLS buffer caching
- ▶ RLS lock structure sizes
- ▶ Log stream structure sizes
- ▶ DEADLOCK_DETECTION settings

Some of these factors can be tailored for DBRC parallel RECON access while other factors are system-wide for other users of RLS and TVS.

If you request BACKUP.RECON in parallel RECON access mode, DBRC will quiesce and close the RECON and will use an NRI lock instead of an CRE lock for reads, if it can, for better performance. It can do this only if there are no retained locks, so it checks the status just prior to invoking REPRO to do the copy.

If you request COPY RECON to copy a good data set to a spare data set, the copy must be done under a quiesce. The input data set is read in the current mode (TVS or LSR) and if it is TVS, NRI will be used. The output data set is opened in NSR mode and written to in NSR mode for efficiency reasons. Note that the spare might have been previously opened in TVS mode and must be closed and reopened in NSR mode before the copy operation.

If writing to an empty RECON, the header record and header extension must be written. For serial, this was done in NSR mode so the RECONS need to be closed and reopened in LSR mode.

DBRC can write to an empty RECON in TVS mode, so no close and reopen are needed. But, after the headers are written, a Commit is done so that the data is in RECON in case a failure occurs after this point.

Monitoring locks

You use the D SMS,CFLS command to display locking information from SMSVSAM. In Example 7-11 on page 175, we show the lock structure size to be 16 384 K or 16 MB. This implies that the lock table size is 8 MB, which is half of 16 MB. There are 48 252 record table (or record list) entries, but only four of these record table entries are being used at the time of the display.

The following lines show the locking rate, the contention rate, the false contention rate, and the average number of requests waiting for locks during the last minute, the last hour, the last eight hours, and the last day. The data returned by the command is for the system where the command is issued and an average of all of the systems. The response shows the lock and contention rates for system SC64. The lines beginning with (03) are averages for all of the systems. The (03) indicates that there are three systems.

Example 7-11 Display SMS,CFLS command output

```

-D SMS,CFLS
IEE932I 236
IGW320I 14:52:52 Display SMS,CFLS
PRIMARY STRUCTURE:IGWLOCK00 VERSION:B8FC73F0F2A13441 SIZE:16384K
RECORD TABLE ENTRIES:48525 USED:4
System   Interval   LockRate   ContRate   FContRate   WaitQLen
14:52:52 Display SMS,CFLS
SC64     1 Minute    3.3        25.969     0.000       0.33
SC64     1 Hour      84.0       4.797     0.416       0.10
SC64     8 Hour     17.8       1.004     0.092       0.00
SC64     1 Day      7.7        0.354     0.044       0.00
(03)     1 Minute    2.2        17.714     0.000       0.19
(03)     1 Hour     56.0       3.220     0.282       0.05
(03)     8 Hour     11.4       0.671     0.059       0.00
(03)     1 Day      4.5        0.236     0.024       0.00
***** LEGEND *****
LockRate = number of lock requests per second
CONTRATE = % of lock requests globally managed
FCONTRATE = % of lock requests falsely globally managed
WaitQLen = Average number of requests waiting for locks

```

SMSVSAM issues messages about excessive locks to warn you of the condition as illustrated in Example 7-12:

- ▶ The IGW326W message is issued when the record table (record list) in the lock structure is 80% full. If the record table becomes full, new locks for updates cannot be granted.
- ▶ The IGW859I message is issued when a unit of recovery first exceeds the first MAXLOCKS subparameter in the IGDSMSxx member.
- ▶ The IGW10074I message is issued after the IGW859I message when the unit of recovery requests more locks. Its issuance is controlled by the second MAXLOCKS subparameter, which specifies an incremental number of locks. The message is issued each time that the incremental number of additional locks is requested.

Example 7-12 SMSVSAM messages about excessive locks

```

IGW326W *** Warning *** DFSMS SMSVSAM RECORD TABLE IN IGWLOCK00 IS percent % FULL.

IGW859I JOB jobname UNIT OF RECOVERY urid HAS REQUESTED nnn LOCKS

IGW10074I JOB jobname UNIT OF RECOVERY urid HAS REQUESTED nnn LOCKS. SYSTEM
MAXIMUM IS mmm

```

You can obtain lock and cache structure usage information from the RMF™ Structure Activity Report. IMS data sharing users are familiar with these reports for the IRLM lock structure and OSAM and VSAM cache structures. The same type of report is available for the SMSVSAM structures.

The Cache Structure report gives size, service times, reads from cache, and cross invalidations.

The Lock Structure report gives size, service times, contention, and false contention rate.

From RMF II, the VSAM RLS Activity report gives reports by storage class or by data set for read rate, the percentage of reads from SMSVSAM pool, cache structure, and DASD, and also, the percentage of reads for which the SMSVSAM buffer was invalid.

You can use the VSAM LRU Overview report from RMF III to get sizing information for the SMSVSAM buffer pools. The size of this pool is dynamically adjusted by SMSVSAM. The RLS_MAX_POOL_SIZE parameter sets the limit for the pool size, although it might be temporarily exceeded. During each LRU cycle, SMSVSAM determines whether the system is over the goal. If it is, adjustments are made. The report illustrated in Example 7-13 shows the percentage of time when the buffer pool exceeded the desired limit. It also reports on the percentages of requests which were satisfied from the buffer pool, the cache structures, and DASD.

The VSAM RLS Activity report includes three storage classes that are used by RLS. These storage classes are RLS1, RLS2, and RLS3. For each class, it reports direct and sequential access activities. Under the *Read* columns, it reports the percentage of requests that are satisfied from the SMSVSAM buffers (BMF%), from the Coupling Facility structures (CF%), and from DASD (DASD%). Under *BMF*, the report shows the percentage of reads that found the CI in the SMSVSAM buffer pool but the buffer had been invalidated by an update to the CI in another system.

Example 7-13 VSAM RLS Activity report

RMF V1R7 VSAM RLS Activity - SYSPLEX									
Line 1 of 12									
Command ==>									
Samples: 59 Systems: 2 Date: 08/29/07 Time: 13.16.00 Range: 60 sec									
LRU Status : Good									
Contention % : 0.0									
False Cont % : 0.0									
Stor Class	Access	Resp Time	Read Rate	BMF%	CF%	DASD%	BMF Valid%	BMF False Inv%	Write Rate
RLS1	DIR	0.004	665.6	88.2	0.5	11.3	100	0.01	0.00
	SEQ	0.000	0.00	0.0	0.0	0.0	0.0	0.00	0.00
RLS2	DIR	0.005	200.0	90.5	0.0	9.5	100	0.00	0.00
	SEQ	0.000	0.00	0.0	0.0	0.0	0.0	0.00	0.00
RLS3	DIR	0.003	213.3	90.5	0.0	9.5	100	0.00	0.00
	SEQ	0.000	0.00	0.0	0.0	0.0	0.0	0.00	0.00

Performance test results

Studies were conducted for a couple of scenarios where RECON access activity is particularly dense to provide generic information about parallel contrasted with serial RECON access. Our test conditions were not established to provide definitive numbers, so the results can only be interpreted as indicative of any performance that you might see in your own

installation. Your environment, RECONs, and access patterns will naturally be different than our environment, RECONs, and access patterns established in the tests, so your results will be different:

- ▶ A large batch workload on a single logical partition (LPAR), which was created by using Image Copy (DFSUDMP0) on 500+ databases in parallel with both Access=Serial and Access=Parallel. This showed an approximately 20% reduction in total elapsed time with a negligible impact on performance.
- ▶ DEDB PREOPEN processing of 6 000 Fast Path (FP) areas in a 2-way IMS data sharing environment showed an approximately 15% improvement in elapsed time, again with a negligible impact on performance

Our overall impression is that parallel RECON access is beneficial to a busy installation. Note however that timeouts, often caused by multiple processes against the same resources, such as SUBSYS records during database authorizations, can have a significantly detrimental effect because each timeout can cost approximately 20 seconds. Appropriately setting the DEADLOCK_DETECTION(*nnnn,mmmm*) values in the IGDSMSxx member of SYS1.PARMLIB will help you with timeouts, but remember that these are system-wide values and affect other RLS and TVS users.

7.1.7 Parallel RECON access recovery

Each DFSMStvs instance has its own undo log and associated shunt log as defined by you. The logs are shared by all users of the DFSMStvs instance, but they are not shared by DFSMStvs instances on different z/OS systems. The undo and shunt logs are used for backouts when there is a failure or a deadlock by a user of DFSMStvs services, such as a DBRC instance.

When you IPL a z/OS system, the SMSVSAM address space is automatically started if it has been defined for the system. If the address space fails, it is automatically restarted by z/OS, and SMSVSAM then backs out any work that was in-flight at the time of the failure. It also releases the locks held for the work. After this recovery, it accepts new work.

DBRC survives SMSVSAM failures and reissues the VSAM requests when SMSVSAM recovers.

Peer recovery is the process of dealing with the work that was left in an incomplete state due to the failure of an instance of DFSMStvs. Peer recovery is done by another instance of DFSMStvs. Peer recovery occurs only in cases of system failure, not merely when DFSMStvs fails. When a z/OS system fails, a peer recovery instance of DFSMStvs is used to back out in-flight work and release locks. The z/OS system where peer recovery is run must have a primary instance of DFSMStvs running on it as we illustrate in Figure 7-5 on page 178.

Peer recovery is not automatically invoked unless Automatic Restart Management (ARM) is used to invoke it. Without ARM, you must issue an operator command to start peer recovery. ARM information is available in the *z/OS V1R4.0 - V1R9.0 DFSMStvs Planning and Operating Guide*, SC26-7348, and in *z/OS MVS Setting Up a Sysplex*, SA22-7625.

Peer recovery backs out the in-flight work of the failed system and releases the locks for this work. The peer does not accept new work. Instead, the peer recovery instance of DFSMStvs does these actions:

1. It registers with VSAM RLS and RRS as the failed instance of DFSMStvs in order to gain access to the failed instance's resources and indicates to RRS that it is beginning restart processing as the failed instance. This registration persists until peer recovery is complete. As a result, should the failed instance restart, it will be unable to initialize

because its attempt to register will fail. It might be necessary to restart DFSMSStvs manually after peer recovery is complete.

2. It reads the failed instance's undo log to retrieve information about in-progress units of recovery.
3. It invokes RRS to retrieve information about unit of recovery status.
4. It processes the log data and the information returned by RRS to determine whether to commit or back out units of recovery. It backs out in-flight work and releases locks for this work.

When all outstanding units of recovery have been processed, it unregisters from RRS and VSAM RLS.

The example in Figure 7-5 illustrates that peer recovery can be done on any z/OS system that has a SMSVSAM address space with a current primary instance of DFSMSStvs.

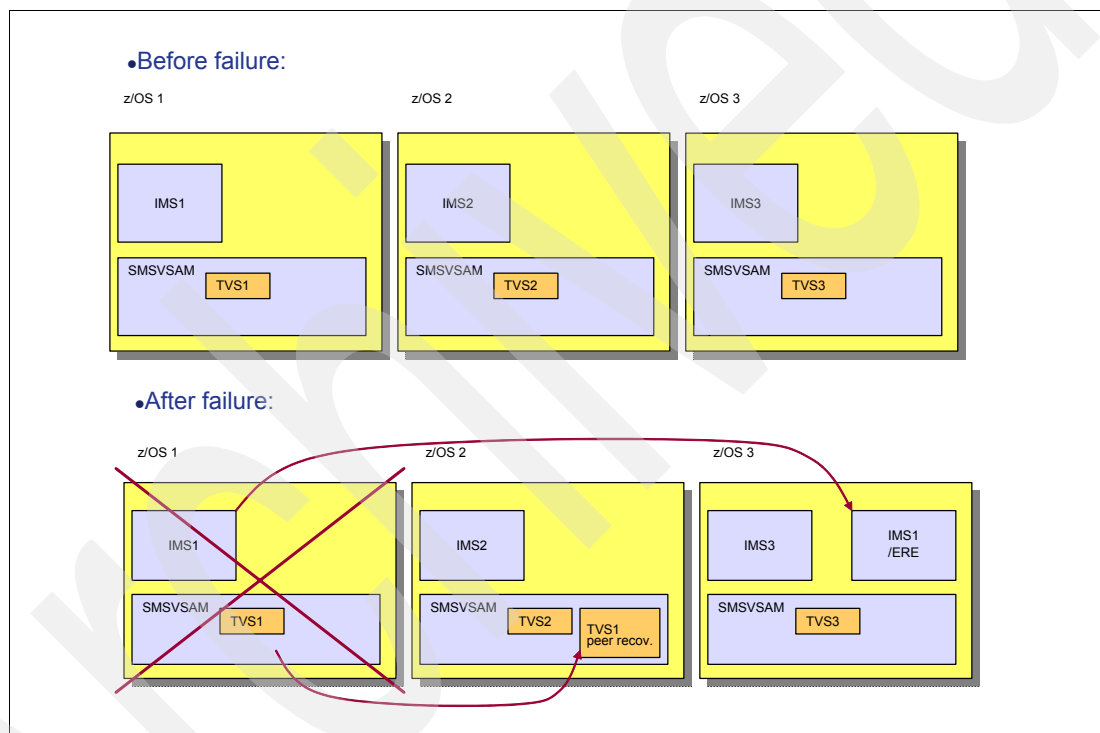


Figure 7-5 DFSMSStvs peer recovery

After the failure, TVS1 peer recovery can be done on z/OS 2 or z/OS 3. Peer recovery does not require restarting the IMSs using it. Of course, you probably want to restart any failed IMS online system. The emergency restart of IMS1 can be done on any z/OS that has a DFSMSStvs instance. It does not have to be done on the same z/OS system where peer recovery is being done for the failed instance of DFSMSStvs.

7.1.8 Summary of serial compared to parallel processing

In Table 7-1 on page 179, we show a summary comparing serial and parallel access to the RECONS:

- Serial processing uses hardware reserves or global data set serialization to serialize access to the RECONS. Only one DBRC instance can access the RECONS at any time. Parallel RECON access allows multiple DBRC instances to access the RECONS at the

same time. It locks individual RECON records to provide integrity. Because it uses locking, there is the possibility of deadlocks or timeouts for lock requests.

- ▶ Serial processing maintains the status of the RECONs in the header/header extension records. Every access to the RECONs first has to check the header/header extension for the current status. Parallel RECON access uses SCI communication to notify other DBRC instances of any change in the RECON configuration. Parallel RECON access does not access the header/header extension with every access to the RECONs.
- ▶ Serial processing has special handling to maintain integrity for multiple updates for requests that require updates to multiple RECON records. It writes a MUP record to the header/header extension before doing its intended multiple updates and then deletes the MUP information after the intended multiple updates are done. Parallel RECON access does not have this processing. It uses TVS to maintain integrity for multiple updates. If one or more RECON records are updated, but the other records in the request cannot be updated, the TVS log is used to back out the updates that were made.
- ▶ Serial processing uses VSAM local shared resources (LSR) pools. Each DBRC instance has its own pools. These pools are invalidated at the beginning of each request to DBRC, which is necessary because another DBRC instance might have modified the CIs that are in the pool. Parallel RECON access does not use LSR pools. It uses the SMSVSAM RLS pool in its LPAR. Parallel Sysplex buffer invalidations are used to invalidate only the buffers that have been updated by DBRC instances running in other LPARs. Cache structures are used in the invalidation process. These cache structures also maintain copies of RECON CIs so that some reads to the RECON data sets might be eliminated:
 - SCI registration and Automatic RECON Loss Notification (ARLN) are optional with serial access. They are required with parallel RECON access.

Table 7-1 Summary of serial compared to parallel processing

Serial	Parallel RECON access
Hardware reserves (or global serialization) serialize access to the entire RECONs.	Individual RECON records are locked for serialization. Deadlocks and lock timeouts can occur.
RECON header extension read during <i>logical open</i> to determine configuration	SCI used to communicate RECON configuration changes
<i>Multiple update</i> processing used for recoveries from failures	TVS log used for recoveries from failures
Uses VSAM LSR pools	Uses VSAM RLS pools and cache structures
SCI registration and ARLN are optional.	SCI registration and ARLN are required.

7.1.9 Migration and coexistence considerations for parallel RECON access

Products and application programs that access the RECONs natively must be adapted for DFSMSStvs. We recommend that you use the DBRC application program interface (API) that supports both parallel RECON access and serial access. If you do not use the DBRC API, your programs must support the DFSMSStvs interfaces. These include the use of RRS for commit and backout, opening the RECONs for DFSMSStvs access, and the use of RPLs that invoke DFSMSStvs access. Even if your programs successfully implement these interfaces, full support cannot be achieved. For example, there is no interface to quiesce user code for the LIST command with the STATIC QUIESCE option.

Parallel RECON access cannot coexist with serial access and requires MINVERS('10.1'). The MINVERS level cannot be downgraded after parallel RECON access is turned on.

The RECON I/O exit routine (DSPCEXT0) must be modified.

You invoke parallel RECON access by issuing the CHANGE.RECON ACCESS(PARALLEL) command. This ALTERs the RECON definition and adds LOG(UNDO). Because parallel RECON access requires a DFSMSxx environment, you must specify IGDSMSxx parameters, the structure and log stream definitions, and the SHCDS data sets. You need to supply the appropriate Security (RACF) authorities for RRS and the SMSVSAM address space.

Because your DBRC environment is now different, you will need to build updated operation and recovery procedures and thoroughly test them.

7.1.10 DBRC serviceability enhancements

IMS Version 10 has enhanced DBRC Dump Formatting by using IMS EDA services to provide formatting of the major control blocks. The formatting routines use BPE formatting services to access and format the dump data and the control block format modelling is done via MVS macros.

In Example 7-14 and Example 7-15, we show how to get to the DBRC formatting by selecting E and then selecting 6. We then show entering S for Select in Example 7-16 on page 181.

Example 7-14 Panel navigation step one

```
----- IMS DUMP FORMATTING PRIMARY MENU -----
OPTION ==> E

0 INIT          - IMS formatting initialization and content summary
1 BROWSE        - Browse Dump dataset *****
2 HI-LEVEL      - IMS Component level formatting *USERID - VANCEGW
3 LOW-LEVEL     - IMS ITASK level formatting *DATE - 04/11/11
4 ANALYSIS      - IMS dump analysis *JULIAN - 04.316
5 USER          - IMS user formatting routines *TIME - 09:40
6 OTHER COMP    - Other IMS components (BPE, CQS...) *PREFIX - VANCEGW
7 OTHER PROD    - Other IMS-related products *TERMINAL- 3278
E EDA           - IMS Enhanced Dump Analysis *PF KEYS -
T TUTORIAL      - IMS dump formatting tutorial *****
X EXIT          - Exit IMS dump formatting

Enter END or RETURN command to terminate IMS component formatting.
Use PFKeys to scroll up and down if needed.

* This product contains "Restricted Materials of IBM". 5635-A01 (C) *
* Copyright IBM Corp. 1991,2000 Licensed Materials - Property of IBM. *
* All rights reserved. U.S. government users restricted rights - use, *
* duplication, or disclosure restricted by GSA ADP schedule contract *
* with IBM Corp. Refer to copyright instructions form number G120-2083. *
```

Example 7-15 Panel navigation step two

```
----- IMS ENHANCED DUMP FORMATTING MENU -----
OPTION ==> 6

1 BROWSE        - Browse dump dataset (IPCS norm) *****
2 DB            - Full Function Data Base *USERID - VANCEGW
3 FP           - Fast Path Data Base *DATE - 04/11/11
4 TM           - Transaction Management and DC *JULIAN - 04.316
```

Enter END or RETURN command to terminate IMS component formatting.

```
----- DATA BASE FORMATTING OPTIONS ----- Row 1 to 1 of 1
```

```

Cmd      Type      Description
-----
v-----vvvvvvvv-----
s      ALL      DBRC Primary control blocks and trace
***** Bottom of data *****

```

Example 7-17 GDB EDA formatting step one

```

GDB: 0C80C5F0
+0000 ID..... DSPGDB      LENGTH... 00000238
+000C GDBTIME.. 2004316F    00101120  0000032D
+0018 CALLLV.. 00000000    0000000F          TRACEOPT. 00
+0024 RLNAREA.. 0C80C578    PLEXNAME. PLEX1LCL
+0030 ADSFLAGS. 00
+0031 ENVFLGS1. 18          MINVERS.. A1          SRVCLVL.. 00
+0034 ENVFLGS2. 002A2004    EXITSAVE. 00000000    HISAVE... 000067A0
+0040 GDBSTOP.. 0C80DA08    GDBTRACE. 00000000    GDBRSTG.. 0001B4F0
+004C GDBMSGAR. 0C80C828    GDBDLTAR. 0C80D1C0    GDBBSAVE. 00000000
+0058 GDBRECAR. 0C80D228    GDBLISAR. 0C80D400    GDBDGCDE. 0000
+0062 GDBSTOPC. 0000        GDBEXTAR. 0C80D420    GDBVERAR. 0C80D450
+006C GDBOPCAR. 00000000    GDBGPDAR. 0C80D460    GDBJCLAR. 0C80D490
+0078 GDBOLCAR. 00000000    GDBRUPAR. 0C80D558    GDBRETAR. 00000000
+0084 GDBCSAVE. 00000000    GDBISAVE. 00000000    GDBRSAVE. 00000000
+0090 MORFLAGS. 00          GDBSTRTM. 00000000    00000000  00000000
+00A0 GDBDBTAD. 00000000    GDBRESAV. 00000000    GDBTSTLN. 00003A10
+00AC GDBDCPRS. 00000000    GDBSTSAV. 0C83719C    GDBSTGET. 0C7ABEC8
+00EC GDBSTFRE. 0C7ABC00    GDBSTCUR. 0C80E000    GDBSTNUM. 0001
+00F6 GDBSTMAX. 0002        GDBLDRSV. 8000CDBE    GDBLOADR. 8C8068F8
+010A RSTFLAGS. 00          GDBINDBD. 00000000    00000000
+0117 GDBRLVL.. A1          GDBVOFLG. 00

```

Example 7-18 shows flag formatting, one byte at a time. We recommend that you include the bit value in the formatted text as a reference.

Example 7-18 GDB EDA flag formatting

```

. . .
+018C CALLSTK.. 0C80D648 STKMAX... 003C      LCLOFF... 032D
+0194 TFMTOPEN. ONP2      TPRCOPEN. 01      TFMTJOB.. ONP2
+019D TPRCJOB.. 01        TFMTCUR.. ONP2      TPRCCUR.. 01
+01A3 TFMTCMD.. 00000000 TPRCCMD.. 00      GDBTZDEF. 0000
+01AA GDBTZNUM. 0000      GDBTZTAB. 0C774C90 TIMEFLAG. 0000
+01B2 TIMEPREC. 0001      KEYMASK.. FFFFFFFF FFFFFFFF 00000000
+01C0 LOGRETN.. 0000001F 00000000 0000000C GDBDSNAR. 0C80D618
+01D0 GDBIMSDD. IMS      SIZEDSNM. 0000000F SIZEVOLN. 00000010
+01E4 SIZEPCT.. 0000005F LOGDSNUM. 00000003 LOGDSNUM. 00000010
+01F0 HISTLIST. 00000000 CMDFLAGS. 00      CMDHLQ... 00000000
+01FC      00000000 DMBPTRK.. 00000000 IMSPLEX.. 00000000
+020C      00000000 GDBPRAB.. 0C80DEDO
Flag analysis for GDBENV01 flags (byte 1):
  GDBPRA (X'10') - Parallel RECON Access
  GDBNSWAP (X'08') - Swapping suppressed
Flag analysis for gdbenv flags (Byte 2):
  GDBESTAE (X'20') - ESTAE set
  GDBMVS (X'08') - OS: MVS
  GDBMVSE (X'02') - OS: MVS/XA
Flag analysis for gdbenv flags (Byte 3):
  GDBRCOPN (X'20') - RECON was opened
Flag analysis for gdbenv flags (Byte 4):
  GDBCHK17 (X'04') - Check 17 char of DSN

```

In Example 7-19, we show the formatting of the BRLSB, which interprets flags.

Example 7-19 BRLSB sample formatting, step one

```

*****
* * * DBRC: Current BRLSB - PREFIX * * *
*****

DFSBRLSB: 00059CA0
      DFSBRLSB Prefix Section:
+0000 BRLPFCFP. 00000000 BRLPECBP. 0003E060 BRLPRLSP. 00059CB4
+000C BRLPWKBP. 000121A8 BRLPRSF.. 0064

*****
* * * DBRC: Current BRLSB * * *
*****

DFSBRLSB: 00059CB4
      DFSBRLSB Section:
+0000 ID..... DFSBRLSB LENGTH... 00000100 BRLBFFLG. 17190002
+0010 BRLBEFLG. 00000000 BRLB5FLG. 00000000 BRLB5FLG. 00E30000
+001C BRLBPRM1. 00000000 BRLBRSCD. 00000000 BRLBPRNT. 00000000
+0028 BRLBRCDE. 0000      BRLBDCDE. 0000      BRLBSID.. 000091C8
+0030 BRLBEMOD. 00000001 BRLBEINF. 0C80C5F0 BRLBTIME. 8001EAOE
+003C BRLBFDBK. 00000000 BRLBINTP. 00000000
      BRLBF1CT (X'17') - Control

```

```

BRLBF2SF (X'19') - Signoff
Flag analysis for BRLBFFLG flags:
BRLBF4FI (X'02') - Feature initialized

```

In Example 7-20, we show formatting of the BRLSB extension.

Example 7-20 BRLSB extension sample formatting, step two

```

*****
* * * DBRC: Current BRLSB Variable Portion * * *
*****

BRLBSLG: 00059D04
      DFSBRLSB IDLOG Extension:
+0000 BRLBLDBD. 00000000 BRLBLDDN. 00000000 BRLBLLOG. 00000000
+000C BRLBLDDL. 00000000 BRLBLVOL. 00000000 BRLBLAUT. 00000000
+0018 BRLBLRIT. 00000000 BRLBLDSN. 00000000 BRLBLRLM. 000091BC
+0024 BRLBLADT. 00000000 BRLBLBCK. 00000000 BRLBLSOT. 0000
+002E BRLBLFSN. 0000 BRLBLHIC. 00000000 BRLBLDMB. 00000000
+0038 BRLBLUOR. 00000000 BRLBLURV. 00000000 BRLBPRTM. C9D4E2C7
+0044 BRLBLDTM. E2C7F140 BRLBSDS#. 00000000 00000000 00000000
+005C      00000000 00000000 00000000 00000000 00000000
+0054 BRLBTRKI. 00000000 BRLBLATM. 00000000 BRLBLUSI. 00000000
+0060 BRLBLRID. 00000000 BRLBLPTK. 00000000 BRLBLGTM. 00000000
+006C BRLBLOGF. 00000000 BRLBBSN. 00000000 BRLBSTOP. 00000000
+0078 BRLBFLGA. 00

```

7.2 DBRC API-2

IMS Version 10 introduces several enhancements to the DBRC API. This section provides an overview of these enhancements. Refer to *IMS Version 10: System Programming API Reference*, SC18-9967, which contains details about the DSPAPI macro, its parameters, and how you use it and also the control blocks created by your DBRC API requests.

This section includes these topics:

- ▶ Alternate RECON and IMS DD names
- ▶ RECON update capability: Made by using DBRC commands (INIT, CHANGE, and NOTIFY)
- ▶ Register as subsystem: Allows application to authorize databases
- ▶ Database authorizations: Allows application to perform utility functions
- ▶ QUERY enhancements: DBDS, Partition, Log, and wildcard support
- ▶ Security enhancements: Extension of DBRC command authorization
- ▶ Miscellaneous enhancements

7.2.1 Alternate RECON and IMS DD names

Your programs using the DBRC API can use alternate names for the IMS (DBDLIB) and RECON DDNAMEs as we show in Example 7-21 on page 184. This can make it easier for you to develop programs that access multiple sets of RECONs. In IMS Version 9, these programs had to use dynamic allocation (SVC 99) to change the data sets for RECON1,

RECON2, and RECON3 to access a different set of RECONS. In IMS Version 10, you can use JCL to allocate multiple sets or you can use DFSMDA to create dynamic allocation members for different RECONS.

Example 7-21 Alternate RECON and IMS DD names

```
DSPAPI FUNC=STARTDBRC IMS=imsddname
      RECON1=r1ddname RECON2=r2ddname RECON3=r3ddname
      ...
```

The DFSMDA macro has been enhanced to support the alternate DD names for RECONS. A new parameter, DDNAME=, has been added to the DFSMDA macro for TYPE=RECON as we show in Example 7-22.

Example 7-22 DFSMDA RECON DD name

```
DFSMDA TYPE=RECON,DSNAME=dsname,DDNAME=ddname
```

Your programs using the DBRC API are still restricted to accessing only one set of RECONS at a time. A FUNC=STOPDBRC request must be processed before a second set of RECONS can be opened with a second FUNC=STARTDBRC request.

As with previous releases, if you invoke the DBRC utility from your program you can use the DSPURXRT entry point. IMS Version 10 has added the capability for you to specify alternate DD names for the RECON data sets. This is done through the expansion of the list of DDNAMES that you pass to the entry point in the second word of the argument list.

7.2.2 Alter RECON update capability

The original implementation of the DBRC API allowed you to query the data in the RECONS by issuing DSPAPI macros. The extensions to the API in IMS Version 10 allow you to update the RECONS by invoking DBRC commands, such as INIT, CHANGE, DELETE, NOTIFY, and RESET. You can also use this interface to issue GENJCL and BACKUP commands that do not update the RECONS. LIST commands are not valid. You must use the DSPAPI FUNC=QUERY function to list data from the RECONS.

The output from commands is returned in an API output block. This block has the same format as the QUERY blocks that were introduced in IMS Version 9. Details of the command blocks are shown in Figure 7-6 on page 185.

You use the FUNC=COMMAND on the DSPAPI macro to issue a DBRC command as we show in Example 7-23. The COMMAND= parameter specifies the address where the command resides. The command consists of a header followed by a DBRC command. The header is a fullword containing the length (in bytes) of the following command. The output block address is returned at the address specified on the OUTPUT parameter. The output of the DBRC command is written in this output block.

Example 7-23 DBRC API command function

```
DSPAPI FUNC=COMMAND COMMAND=cmdaddr TOKEN=taddr,
      OUTPUT=output BUFFERLENGTH=number|4096
      SUPPRESS=YES|NO ...
```

You specify the size of the buffer to be used for the output of the command in the BUFFERLENGTH= parameter. The output is written in the output block produced by the request. It is formatted for display to a terminal rather than printed to a SYSPRINT data set.

The output buffer consists of:

- ▶ A halfword containing the length of the buffer that includes this halfword
- ▶ A halfword of zeroes
- ▶ One or more generated lines of output

Each generated line consists of:

- ▶ A halfword containing the length of the line that includes this halfword
- ▶ A halfword of zeroes
- ▶ The generated line of output

Carriage control characters are removed from each line. Each line is terminated with a New Line (NL) character (X'15'). Blank lines are removed from the output. The output buffer has a default length of 4 096 bytes. You can request a buffer of up to 32 760 bytes. If the buffer is not large enough to hold all of the lines of output, the last line is a warning message as we show in Example 7-24.

Example 7-24 OUTPUT TRUNCATED message

DSP0057I MESSAGE BUFFER FULL, OUTPUT TRUNCATED

You can use SUPPRESS=YES so that no output block is created when a return code of 0 is returned. This eliminates the possible need to release the storage for the block with a FUNC=RELBUF request. SUPPRESS=NO is the default.

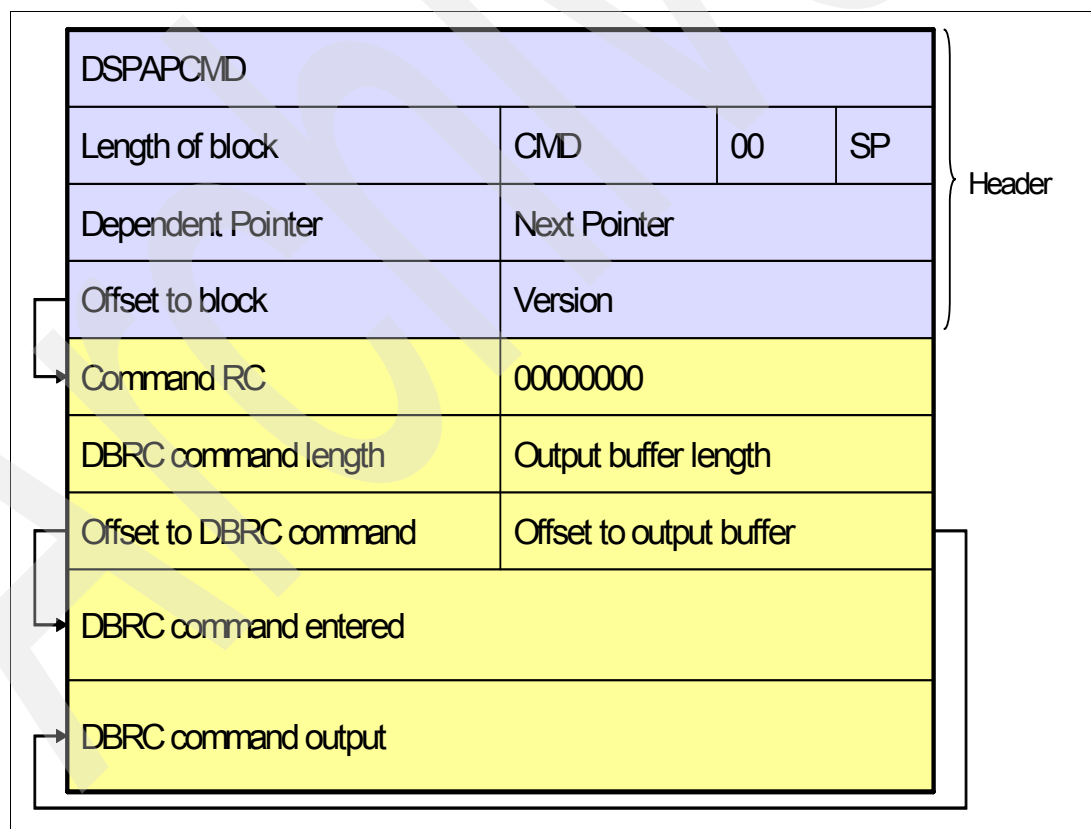


Figure 7-6 Command output block

DSPAPCMD is the name of the mapping macro for the format of the output block created by DSPAPI with FUNC=COMMAND that we show in Figure 7-6. It has the same structure as the

blocks created by DSPAPI with FUNC=QUERY. The first 32 bytes are the prefix, or header, of the block. The diagram is eight bytes wide (except for the last two entries, which are variable):

- ▶ The literal DSPAPCMD is stored in the first eight bytes of the block to get your attention.
- ▶ CMD is a 2-byte indicator that this is a command block.
- ▶ 00 is one byte of x'00'.
- ▶ SP is the subpool in which this block is stored.
- ▶ The dependent pointer and next pointer are not used in the command output block. These blocks are not chained to other blocks.
- ▶ Version is the version of this block. Its value is 2.0 in IMS Version 10.
- ▶ Command RC is the return code from the command.
- ▶ 00000000 is four bytes of x'00'.
- ▶ DBRC command length is the length of the command passed by the DSPAPI FUNC=COMMAND macro.
- ▶ DBRC command entered is the DBRC command entered by the DSPAPI FUNC=COMMAND macro. It is copied from the address specified in the macro to this block.
- ▶ Output buffer length is the length of the output buffer that was specified in the DSPAPI FUNC=COMMAND macro.
- ▶ The offset fields are 4-byte offsets to the indicated fields in the block.

7.2.3 Register as a subsystem

The STARTDBRC function is enhanced to support the creation of subsystem records. Your application can register with DBRC as a subsystem as shown in Example 7-25. You must register as a subsystem to authorize databases. Database authorization support has also been added to the DBRC API. We describe it in 7.2.4, "Database authorizations" on page 187. You register subsystems by adding an SSID= parameter to the DSPAPI FUNC=STARTDBRC macro. The meanings of the other parameters on the macro are unchanged.

Example 7-25 Subsystem registration

```
DSPAPI FUNC=STARTDBRC SSID=ssidname ...
```

A new type of subsystem, DBRCAPI, has been added to those subsystems that are stored in the SUBSYS record, which are used for DSPAPI FUNC=STARTDBRC to create the SUBSYS record.

The QUERY TYPE=SUBSYS has been enhanced to include SSTYPE=DBRCAPI to limit the returned information to only DBRC API subsystems. A new flag, apqss_DBRCAPI, has been added to the output block created by a QUERY TYPE=SUBSYS. It indicates that the subsystem is a DBRC API subsystem. The bit is only set when VERSION=2.0 is specified on the QUERY. Support for this new type has also been added to LIST.SUBSYS, NOTIFY.SUBSYS, and LIST output as follows:

- ▶ QUERY TYPE=SUBSYS

This QUERY request has been enhanced to include SSTYPE=DBRCAPI to limit the returned information to only DBRC API subsystems. A new flag has been added to the output block created by a QUERY TYPE=SUBSYS. It indicates that the subsystem is a DBRC API subsystem. The bit is only set when you specify VERSION=2.0 on the QUERY.

► LIST.SUBSYS and LIST.RECON

The output of the LIST.SUBSYS and LIST.RECON commands includes SSTYPE=DBRCAPI for DBRC API subsystems. You can list only DBRCAPI subsystems by using the DBRCAPI keyword on the LIST.SUBSYS command.

► NOTIFY.SUBSYS

The NOTIFY.SUBSYS command has been enhanced. The new DBRCAPI keyword causes the command to create a DBRC API subsystem record.

You delete the DBRCAPI subsystem records by issuing DSPAPI FUNC=STOPDBRC macros. If any databases are authorized to the subsystem, they are unauthorized when the macro is processed.

If your DBRCAPI application abends or if it terminates normally without issuing the DSPAPI FUNC=STOPDBRC, its subsystem record is not deleted. Any databases authorized to the subsystem remain authorized. You can delete the subsystem record by issuing the same commands that delete any subsystem record. If the application has authorized databases, you use the CHANGE.SUBSYS STARTRCV and CHANGE.SUBSYS ENDRECOV to delete the authorizations. You must delete the database authorizations before the subsystem record can be deleted. You use the DELETE.SUBSYS command to delete the subsystem record.

7.2.4 Database authorizations

You can build your applications using the DBRC API to authorize and unauthorize databases, partitions, and areas. Databases are authorized to subsystems, so authorization requests come from applications that are registered as subsystems. Because authorizations update the RECONs, the application cannot be using READONLY.

You can specify the access intent when authorizing a database, partition, or area. Block level data sharing is not supported for DBRC API users; therefore, UP is not allowed. You must use EX if any changes are to be made to a database, partition, or area.

We show how you can grant utility privileges with an authorization in Example 7-26. This allows authorizations even though the Prohibit Further Authorization, Image Copy Needed, or Read Only flags are on, but your DBRC API application is providing functions similar to IMS database utilities, such as backup, recovery, or reorganization.

Example 7-26 Database authorization

```
DSPAPI FUNC=AUTH ACCESS=EX|RD|RO TOKEN=taddr
AUTHLIST=aaddr UTILITY=IC|RECOV|REORG|NONE
OUTPUT=oaddr . . .
```

taddr Address of 4-byte API token

aaddr Address of authorization list

AUTHLIST= Here you specify the address of the list of databases, partitions, and Fast Path areas to be authorized. The list consists of a fullword that contains the number of elements in the list, and a fullword that contains the length of an element, followed by one or more elements. Each element consists of an 8-character database name, partition name, or Fast Path DEDB name and eight characters of blanks (X'40') or a Fast Path area name.

UTILITY UTILITY is for your use in applications that perform utility functions. UTILITY= values, other than NONE, specify that DBRC is to evaluate

the authorization request as it evaluates a request from the indicated utility. Utility authorization requests bypass some of the database and data set status indicators. Image Copy (IC) can get authorization when the Image Copy Needed or Prevent Further Authorization flags are on. Reorganization (REORG) can get authorization when the Image Copy Needed, Prevent Further Authorization, or Read Only flags are on. Recovery (RECOV) can get authorization when any of the status flags are on. These include the Image Copy Needed, Prevent Further Authorization, Read Only, Recovery Needed, and Backout Needed flags.

oaddr

Address of the output block. You can release the block with an RELBUF API request.

7.2.5 QUERY enhancements

We describe the following four aspects of the QUERY capability:

- ▶ HALDB partition query
- ▶ DBDS query
- ▶ Log query enhancement
- ▶ Wildcard specification in name parameters

HALDB partition query

You could use the IMS Version 9 implementation of `FUNC=QUERY TYPE=DB` to get information about HALDB partitions by specifying a database name and then information about all of the partitions was returned. IMS Version 10 adds `FUNC=QUERY TYPE=PART`, which allows you to get information about a partition without specifying the database name and to retrieve partition information without retrieving the database information.

You specify either the `PARTNAME=` or `DBNAME=` parameter but not both on the same macro. When you specify `PARTNAME=`, you can request a specific partition by name; alternatively, you can request the next or previous partition relative to the specified partition.

When you specify `DBNAME=`, you can request all of the partitions in the database; alternatively, you can request the first or last partition in the database.

As with `FUNC=QUERY TYPE=DB`, you can request DBDS information for the partitions returned and you can request allocation (ALLOC), image copy (IC), recovery (RECOV), and reorganization (REORG) information for these database data sets.

You can disable partitions in a HALDB database with a `CHANGE.PART` command with the `DISABLE` keyword. Definitions for disabled partitions remain in the RECONs, but they are not actively used. That is, they are no longer part of the definition of their master database. Disabled partitions are not returned by `FUNC=QUERY TYPE=PART` macros where `LOC=ALL`, `FIRST`, `LAST`, `PREV`, or `NEXT` is specified; however, they can be returned with `PARTNAME= LOC=SPEC`.

DBDS query

You could use the IMS Version 9 implementation of `FUNC=QUERY TYPE=DB` to get information about database data sets by specifying a database name and requesting DBDS information: information about a specific data set in the database, all data sets in the database, or none of them. IMS Version 10 adds `FUNC=QUERY TYPE=DBDS`, which provides you with an alternative way of getting this information. It also allows you to retrieve information about the first DBDS in a database with one request and the next DBDSs with subsequent requests.

You must specify either the GROUP= or DBNAME= parameter but both of these parameters cannot be specified on the same macro.

When you specify GROUP=, information about all of the database data sets in the group is returned. A group is either a DBDS group or a CA group. You cannot specify a database group or a recovery group.

When you specify DBNAME=, the information returned is determined by the DDN= and LOC= parameters:

- ▶ If you specify DDN=*, blocks for all data sets in the database are returned. DDN=* is the default.
- ▶ If you specify LOC=FIRST without DDN=, the first data set in the database is returned.
- ▶ If you specify DDN=d with LOC=SPEC, a block for the indicated data set is returned.
- ▶ If you specify DDN= with LOC=NEXT, the data set after the specified data set is returned.

As with FUNC=QUERY TYPE=DB, you can request allocation (ALLOC), image copy (IC), recovery (RECOV), and reorganization (REORG) information for these database data sets.

When you specified the IMS Version 9 FUNC=QUERY with a TYPE= value of CAGROUP, DBGROUP, RECOVGROUP, or GSGROUP, you provided the NAME= parameter to specify the name of the group. IMS Version 10 allows the use of the new GROUP= parameter to specify the name of the group. It produces exactly the same results as the use of NAME=. The use of GROUP= was added to be compatible with FUNC=QUERY TYPE=DBDS where GROUP= is used for the name of the group. NAME= is still supported when TYPE= specifies a group.

Log query enhancement

The IMS Version 9 implementation of DSPAPI FUNC=QUERY TYPE=LOG has been enhanced in IMS Version 10. QUERY requests for TYPE=LOG required a STARTIME= parameter to give the start time of a log. IMS Version 10 adds the capability to request information about a range of logs. If you specify FROMTIME= without TOTIME=, all logs from the specified time to the present are returned. If you specify TOTIME= without FROMTIME=, all logs up to the specified time are returned. If you specify both FROMTIME= and TOTIME=, all logs within the time range are returned. You can also specify SSID= to return logs for the specified subsystem. Because information about multiple logs can be returned on the call with these time range requests, the LOGINFO blocks (DSPAPQLI) can be chained with their “next” pointers as illustrated in Figure 7-7.

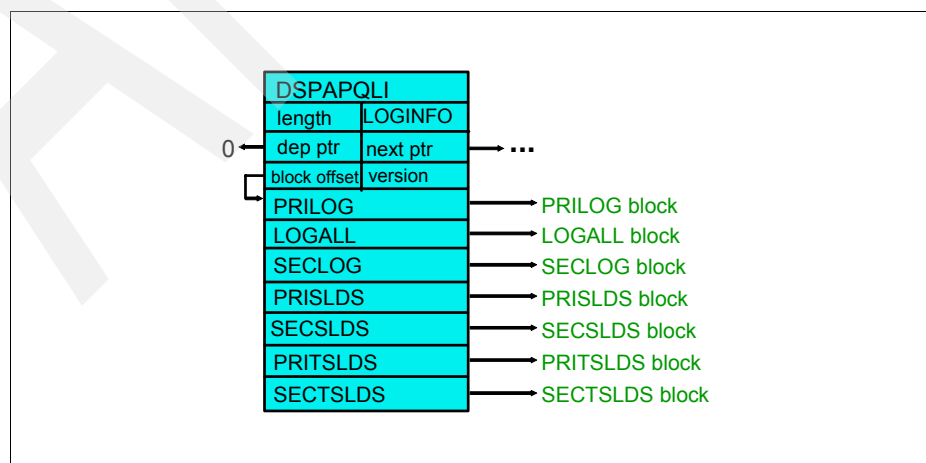


Figure 7-7 Query log returned data

Wildcard specification in name parameters

You can use an asterisk (*) as a wildcard for returning a group of objects that follow a name pattern. The asterisk is placed at the end of a supplied name value with at least one alphabetic character preceding the asterisk. We summarize this in Figure 7-8.

Query type	Allowed	Not allowed
DB	DBNAME	DDN, DBLIST
PART	N/A	DBNAME, PARTNAME, DDN
DBDS	N/A	DBNAME, DDN, GROUP
*GROUP	NAME, GROUP	N/A
LOG	N/A	SSID
OLDS	SSID	N/A
SUBSYS	SSID	N/A
BACKOUT	SSID	N/A
RECON	N/A	N/A

Figure 7-8 Name parameter wildcards

7.2.6 Security enhancements

DBRC command security was introduced in IMS Version 8. You can use it to invoke authorization checking for DBRC commands with RACF, or any SAF security product. Alternatively, you can invoke an exit routine or both the security product and the exit routine. You authorize commands by defining a resource representing the command. In RACF, this is done with an RDEFINE statement.

This authorization is extended to the DBRC API in IMS Version 10. API requests invoke security authorization checking. Security authorization checking uses resources that are defined to secure specific commands or API requests. Resource names have the following form:

hlq.verb.resourcetype.resourcename

The high-level qualifier (*hlq*) in the resource identifies a set of resources used for security authorization. The next variable, *verb*, identifies a DBRC command or API request type. Then, *resourcetype* identifies a resource type on which the command or request operates. For example, a LIST.DB command operates on the database resource type. The *resourcetype* is optional. The *resourcename* identifies a specific resource instance. For example, a LIST.DB DBD(XYZ) command operates on the XYZ resource instance or name. *Resourcename* is optional.

This scheme has been extended for API requests as we show in Table 7-2 on page 191.

Table 7-2 DBRC command security

API request	Resource defined to RACF
FUNC=STARTDBRC and FUNC=STOPDBRC	<i>hlq.STDBRC.ssid</i>
FUNC=QUERY	Equivalent resource for LIST command, for example, <i>hlq.LIST.DB.dbname</i> for TYPE=DB DBNAME= <i>dbname</i>
FUNC=COMMAND	Command resource, for example, <i>hlq.CHANGE.DB.dbname</i>
FUNC=AUTH and FUNC=UNAUTH	<i>hlq.AUTH.dbname</i>
FUNC=RELBUF	Not applicable; security authorization not checked

TYPE=STARTDBRC and TYPE=STOPDBRC requests are checked using a resource of *hlq.STDBRC.ssid*. The variable, *ssid*, specifies a subsystem and is optional. It restricts the use of the requests for a specific subsystem.

In previous releases, the DSP1154A message is sent when DBRC command authorization is denied by the DSPDCAX0 user exit. Message DSP1157A is sent when authorization is denied by RACF or an alternate security product. The text of the message is changed slightly to indicate that an API request, not a DBRC command, failed authorization.

Wildcards in API QUERY requests are converted to using the ALL resource name. An example of the wildcard is QUERY TYPE=SUBSYS SSID=IM*, which is secured by *hlq.LIST.SUBSYS.ALL* profile. AUTH and UNAUTH validate each entry in the AUTHLIST.

Because security was not checked in IMS Version 9, the security checking in IMS Version 10 might cause an application to fail in IMS Version 10 where it ran successfully in IMS Version 9.

The text of the messages for DBRC commands:

- ▶ DSP1154A DBRC COMMAND AUTHORIZATION DENIED BY DSPDCAX0 FOR USER *user ID* RESOURCE NAME=*resource name* RC=*rc*
- ▶ DSP1157A USER *user ID* NOT AUTHORIZED FOR COMMAND RESOURCE NAME=*resource name* REQUEST=*reqtype* SAF RC=*safrc* RACF RC=*racfrc* RACF REASON=*racfrsn*

The text of the messages for API requests:

- ▶ DSP1154A DBRC REQUEST AUTHORIZATION DENIED BY DSPDCAX0 FOR USER *user ID* RESOURCE NAME=*resource name* RC=*rc*
- ▶ DSP1157A USER *user ID* NOT AUTHORIZED FOR REQUEST RESOURCE NAME=*resource name* REQUEST=*reqtype* SAF RC=*safrc* RACF RC=*racfrc* RACF REASON=*racfrsn*

7.2.7 Miscellaneous enhancements

User-specified subpool

You can use the new SUBPOOL= parameter on DSPAPI macros, which create blocks. It determines which subpool will be used for the block, but this new parameter is only available when VERSION=2.0 is also specified on the DSPAPI macro. The FUNC=STARTDBRC request establishes the default for the SUBPOOL= parameter on later requests. If you do not specify it on the FUNC=STARTDBRC request, the default is subpool 0.

The output block header, DSPAPQHD, has been enhanced to include the subpool ID. It is stored in a previously reserved byte of the header. Users should see the *z/OS V1.R9.0 MVS Programming Authorized Assembler Services Guide*, SA22-7608, for valid subpools for their programs.

RECON open

The IMS Version 10 DBRC API has added the capability to open a set of RECONS that are not initialized or that are at a previous release level. This allows your application program to initialize the RECONS with an INIT.RECON command or to upgrade them with a CHANGE.RECON UPGRADE command. This can be useful for utility-type programs that create and upgrade RECONS.

Same TCB restriction enforcement

IMS Version 10 enforces the “same TCB” restriction. All of your DBRC API requests from a STARTDBRC through to a STOPDBRC request must be done under the same TCB. An attempt to make this type of a request under a different TCB will result in a failure with return code x'0C' and reason code x'C900000A'. This is a new reason code in IMS Version 10.

This enforcement is done to avoid a possible ABEND. VSAM requires that a close of a data set is done under the same TCB as the open was done. Closes under different TCBs result in ABENDs. Any access to the RECONS might require a close. One example occurs when a RECON data set is extended. If an instance of DBRC accesses the RECONS and determines that a RECON data set has new extents, it closes and reopens the data set to get the new extent information. Another example occurs when a DBRC instance determines that the RECONS have been reconfigured. That is, a RECON data set has been lost and the spare RECON is now made one of the active RECONS. This DBRC instance closes the lost RECON.

DBRC API macro VERSION= parameter

Each DBRC API macro includes the VERSION= parameter. New functions, such as AUTH, and new options, such as READONLY=YES, require VERSION=2.0. In IMS Version 9, the only valid value for VERSION was 1.0. It was also the default. In IMS Version 10, VERSION= defaults to 2.0.

Your applications written for IMS Version 9 will continue to run in IMS Version 10 without change. Reassembly is not required. In fact, reassembly can cause the program to change because of the change in the default for VERSION=. In some cases, the default of VERSION=2.0 can cause different results from the previous default of VERSION=1.0, although this is not always the case. Some of the changes are only in the use of previously reserved bytes in the control blocks that are produced. In any case, if you wrote a program for IMS Version 9 and reassemble it using an IMS Version 10 macro library, it is safest to specify VERSION=1.0 on the DFSAPI macros before the reassembly.

Because the VERSION= parameter defaults to the latest level of the macros and later levels can produce different results, it is safest for you to specify the VERSION= parameter value explicitly every time. This will ensure that future assemblies of DBRC API programs will produce the same results.

7.3 Timestamp precision

IMS Version 10 records timestamps to the microsecond when MINVERS('10.1') is in effect. Previous releases and IMS Version 10 without MINVERS('10.1') record timestamps to the tenth of a second. When you specify a timestamp, it can be abbreviated. That means that you

do not need to specify times to the microsecond. Unspecified parts of the time are padded with zeros.

You can use an expanded timestamp in DBRC commands and utility control statements in IMS Version 10. In previous releases, you could include times to one tenth of a second. In IMS Version 10, you can include times to a microsecond. Compressed and punctuated timestamp formats continue to be used. The change is the additional significant digits for hundredths through millionths of a second. This is illustrated in Figure 7-9.

External timestamp format in IMS Version 8 and IMS Version 9

Compressed: *yydddhmsst* [offset]

072271030234 -8 (2007, Aug. 15 (day 227), 10:30:23.4 PST)

Punctuated: [*yy*]*yy*|*ddd*|*hh*|*mm*|*ss*|*t* [offset]

'2007,227 10:30:23.4 PST'

External timestamp format in IMS Version 10

Compressed: *yydddhmssthijmu* [offset]

07227103023456789 -8 (2007, Aug. 15 (day 227), 10:30:23.456789 PST)

Punctuated: [*yy*]*yy*|*ddd*|*hh*|*mm*|*ss*|*thijmu* [offset]

'2007,227 10:30:23.456789 PST'

thijmu - tenths, hundredths, thousandths, and so forth of second
Omitted trailing digits are assumed to be zeros.

Figure 7-9 DBRC external timestamp formats

The internal format of the timestamp in the RECONs is not changed in IMS Version 10. The format has included microseconds since IMS V6. The PRILOG start times have had zeros in the positions after the tenth of a second. Most other RECON records have had actual values. Nevertheless, they have been ignored by other processing. IMS Version 10 stores actual values in all time fields. When you use MINVERS('10.1') with IMS Version 10, these positions become significant and are not ignored. The internal timestamp format is used by the DBRC API. It has not changed in IMS Version 10. See the illustration in Figure 7-10. This format is used for all timestamps in the DBRC API.

yyyydddFhhmssthijmuFqq

F - hexadecimal 'F'

hijmu - hundredths, thousandths, and so forth of a second

hijmu are ignored in Version 8 and Version 9

hijmu are ignored in Version 10 unless MINVERS('10.1')

qq - quarter hour offset for time zone

s - sign for offset

Figure 7-10 DBRC internal timestamp formats

You can supply abbreviated timestamps, the year and the day are required. Other values are padded with zeros. Example 7-27 on page 194 lists all log records or OLDS entries that have start times on or before day 227 in 2007. Full precision is not required on the TOTIME value. Many RECON records include the timestamp as their keys. When a RECON record is recorded with a full precision timestamp, commands for that specific record require full precision in their timestamps.

Example 7-27 Abbreviated timestamps in commands

```
LIST.LOG TOTIME('2007.227')
```

Example 7-28 shows a command where you must supply the full precision timestamp. An image copy record was created with a full precision timestamp. The timestamp is part of the key. The CHANGE.IC command in the example changes the data set name of the image copy of the database data set with DDNAME ABC01 in the ABC database, which was run at the time specified in the RECTIME parameter.

Example 7-28 Full timestamp in command

```
CHANGE.IC DBD(ABC) DDN(ABC01) -  
RECTIME('2007.227 16:23:31.123456 -08:00') -  
ICDSN(NEW.DSN)
```

The complete list of commands requiring full precision timestamps is shown in Table 7-3.

Table 7-3 Commands requiring full precision timestamps

Commands requiring full precision timestamps			
CHANGE.BKOUT	CHANGE.CA	CHANGE.IC	CHANGE.PRILOG RLDS
CHANGE.PRILOG SLDS	CHANGE.PRILOG TSLDS	CHANGE.SECLOG RLDS	CHANGE.SECLOG SLDS
CHANGE.SECLOG TSLDS	CHANGE.UIC	N/A	DELETE.CA
DELETE.IC	DELETE.RECOV	DELETE.REORG	DELETE.UIC

If you enter a command with a timestamp that has fewer than six digits for microseconds, DBRC uses zeros for the missing digits. Because NOTIFY commands create new information in the RECONs, their timestamps do not have to match existing data. You can use timestamps without all six digits for microseconds for NOTIFY commands. Zeros will be used for any of the missing digits.

7.3.1 DBRC command and message output

These examples show the output with full precision timestamps when you have specified MINVERS('10.1').

DBRC LIST command

The output from a DBRC LIST command, which is shown in Example 7-29 on page 195, includes full precision timestamps. In previous releases, the timestamp in the listing included only tenths of seconds. If you request the time zone offset to be listed, it follows the timestamp. Because the timestamp is larger by five bytes, the offset is moved by five bytes.

In general, there were at least five blanks following timestamps in listings from previous releases, which means that other information did not have to be moved for full precision timestamps in IMS Version 10. The exception is the timeline information in the listing created by a LIST.HISTORY command. The timeline at the end of LIST.HISTORY outputs still has only tenths of seconds. There was not room to add the additional five bytes to these timestamps in the listing.

Example 7-29 Output from a DBRC LIST command

LIST.LOG OLDS(DFSOLP00) SSID(SYS3) TIMEFMT(L,0,P,4)

2007.227 18:31:24.167342 -07:00 LISTING OF RECON PAGE 0002

PRIOLD
SSID=SYS3 # DD ENTRIES=1
EARLIEST CHECKPOINT = 2007.227 16:01:03.123456 -07:00

DDNAME=DFSOLP00 DSN=IMSTESTL.IMS01.OLDSP0
START = 2007.227 16:00:59.123456 -07:00 FIRST DS LSN= 0000000000000001
STOP = 2007.227 16:13:10.185501 -07:00 LAST DS LSN= 000000000000042C
LOCK SEQUENCE# = 000000000000
STATUS=ARC COMPLT FEOV=NO AVAIL
PRILOG TIME=2007.227 16:00:59.123456 -07:00 ARCHIVE JOB NAME=JT161314
VERSION=10.1

DSP0180I NUMBER OF RECORDS LISTED IS 1
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I COMMAND COMPLETION TIME 2007.227 18:31:27.167342 -07:00
IMS VERSION 10 RELEASE 1 DATA BASE RECOVERY CONTROL PAGE 0003
DSP0211I COMMAND PROCESSING COMPLETE
DSP0211I HIGHEST CONDITION CODE = 00

Database Change Accumulation

The Database Change Accumulation (CA) utility DB0 and DB1 control statements have been modified to support timestamps with greater precision. The new expanded timestamp format can be used in IMS Version 10 but it is not required. DB0 control statements are used to specify the database data sets that are accumulated to the new change accumulation data set. DB1 control statements are used to specify the database data sets that are written to the new output log data set. Because the timestamps now can occupy more columns in the control statements, the positions of the database data set DDNAMEs have moved and only three DDNAMEs can be specified on one control statement.

In previous versions, as shown in Example 7-30, the timestamp was specified in columns 12-37 and DDNAMEs were in columns 38-45, 47-54, 56-63, and 65-72. In IMS Version 10, the timestamp uses columns 12-42 and DDNAMEs are in columns 43-50, 52-59, and 61-68. If a database has more than three data sets that are in the same Change Accumulation group, multiple control statements are required in IMS Version 10.

Example 7-30 IMS Version 8 or IMS Version 9 CA control statements

-----1-----2-----3-----4-----5-----6-----7---
//SYSIN DD *
DB0DI32DB012007.227 12:00:00.1 -08:00DI320101 DI320102 DI320103 DI320104
DB0AB77DB012007.227 12:00:00.1 -08:00AB770101 AB770102
/*

IMS Version 10 uses two control statements, as shown in Example 7-31 on page 196, to replace the first control statement in the IMS Version 9 example. Two control statements are required because four DDNAMEs are specified.

Example 7-31 IMS Version 10 CA control statements

```
-----1-----2-----3-----4-----5-----6-----7---
//SYSIN      DD *
DB0DI32DB012007.227 12:00:00.123456 -08:00DI320101 DI320102 DI320103
DB0DI32DB012007.227 12:00:00.123456 -08:00DI320104
DB0AB77DB012007.227 12:00:00.1 -08:00      AB770101 AB770102
DB1DI32DB0204173150000123456-0800          DDN1      DDN2      DDN3
/*
```

GENJCL.CA has been updated to create the new format of the control statements. These changes to the control statements will have no effect if you create Change Accumulation JCL and control statements with GENJCL.CA as most installations do.

Database Recovery utility

You use the S control statement to specify the database and DDNAME for the database data set that you want to recover. If the recovery is a timestamp recovery, you also specify the timestamp. You can use the new expanded timestamp format in IMS Version 10 but it is not required. In previous releases, column 57 was used for an indicator. The indicator could specify that a user image copy had been restored or that an RSR receive was done. See Example 7-32.

Example 7-32 IMS Version 8 and IMS Version 9 Database Recovery control statements

```
-----1-----2-----3-----4-----5-----6-----7---
//SYSIN      DD *
S  DI32DB01 DI320101          2007.227 12:00:00.1-08:00 C
/*
```

Because the timestamp can use column 57, you now code the indicator in column 63 when needed. The first example for IMS Version 10, Example 7-33, shows the use of the expanded timestamp. The second, Example 7-34, shows the use of the shorter timestamp. Both examples have the “C” that indicates that there is no image copy input.

Example 7-33 IMS Version 10 expanded timestamp

```
-----1-----2-----3-----4-----5-----6-----7---
//SYSIN      DD *
S  DI32DB01 DI320101          2007.227 12:00:00.123456 -08:00 C
/*
```

Example 7-34 IMS Version 10 shorter timestamp

```
-----1-----2-----3-----4-----5-----6-----7---
//SYSIN      DD *
S  DI32DB01 DI320101          2007.227 12:00:00.1 -08:00      C
/*
```

There is another small change in the coding of the timestamp. In previous releases, there was no space between the time and the sign for the time zone offset. In the IMS Version 8 or IMS Version 9 Example 7-32, the sign is in column 50. In IMS Version 10, there is a blank between the time and the sign for the offset. In Example 7-33 and Example 7-34 for IMS Version 10, the blank is in column 55 for the first example and column 50 for the second example.

GENJCL.RECOV has been updated to create the new format of the control statement for timestamp recovery. The change in the control statement will have no effect if you create

Database Recovery JCL and control statements with GENJCL.RECOV, which is what the vast majority of IMS installations do.

GENJCL timestamp precision

You can code the precision value on the TIMEFMT parameter of the %SET statement in skeletal JCL. It is a value from 1 to 6. The default in previous releases was 1. In IMS Version 10, the default depends on the MINVERS value. MINVERS('10.1') sets the default to 6. MINVERS values less than '10.1' set the default to 1.

As already mentioned, GENJCL.CA and GENJCL.RECOV commands do not use the default. They always produce control statements with the full precision required by the IMS Version 10 utilities. GENJCL.USER will be affected by the default unless it is overridden.

Full precision timestamps are not implemented unless the RECONS have MINVERS('10.1') specified. Even when MINVERS is set to a lower value, the IMS Version 10 Change Accumulation and Database Recovery utility control statements require new formats that accommodate full precision timestamps. Nevertheless, this is unlikely to be a concern to you, because GENJCL.CA and GENJCL.RECOV in IMS Version 10 always produce control statements with the IMS Version 10 formats.

If you want, you can specify abbreviated timestamps for most uses. DBRC will interpret the time correctly. Full precision timestamps are required in CHANGE and DELETE commands when a full precision timestamp is part of the RECON record key.

7.4 Read-only RECON

System Authorization Facility (SAF) products, such as RACF, support four levels of data set authority. In ascending sequence of authority, these are:

- ▶ READ
- ▶ UPDATE
- ▶ CONTROL
- ▶ ALTER

Previous releases of IMS required at least CONTROL authority for all users of the RECONS, whereas IDCAMS DEFINE and DELETE of a RECON data set required ALTER authority. DBRC opened the RECONS for update with CONTROL specified in the VSAM ACB for the RECONS. This required CONTROL authority for open and was true even if you only wanted to read the RECONS as you do for a LIST command.

IMS Version 10 adds READONLY support for the RECONS. You invoke this with PARM(READONLY) on the EXEC statement for the DBRC utility (DSPURX00) or by specifying the new READONLY=YES parameter on the DBRC API FUNC=STARTDBRC macro. When you specify READONLY, the RECONS are opened for read. This means that only READ authority is required in SAF (RACF).

IMS Version 10 has made another change to open, which previously required CONTROL authority. In IMS Version 10, the open has changed so that you do not need CONTROL authority if you update the RECONS but only UPDATE authority. Of course, ALTER is still required if you DELETE and DEFINE the data sets. If READONLY is not specified, the open is done for update but CONTROL is not specified in the ACB. This means that only UPDATE authority is required.

As with previous releases, if you invoke the DBRC utility from your program, you can use the DSPURXRT entry point. IMS Version 10 has added the capability to specify READONLY through a parameter passed to the entry point in the first word of the argument list.

Because READONLY causes IMS to open the RECONS for input, you will not invoke recovery processes for the RECONS when you use READONLY. There are two kinds of recovery processes. The first is recovery from RECON errors. If an I/O error occurs on a RECON data set, updaters can reconfigure the RECONS, which includes copying the good RECON to the spare. READONLY users cannot do writes, so they cannot do this recovery process. The second recovery process is recovering from a failed DBRC instance. When a DBRC instance (batch job, utility, or online system) updates multiple RECON records, it first writes a Multiple Update Record (MUP) to the RECONS. It then does the updates and, finally, deletes the MUP record. If it fails in the middle of this process, another DBRC instance reads the MUP record and either completes or backs out the changes. If the other DBRC instance is a READONLY user, it cannot perform this recovery because it cannot write.

If your READONLY execution attempts to update the RECONS, message DSP0030E is issued and the application abends. The variable text in the message indicates the operation that was attempted. The possible values are:

CONFIG	An attempt was made to reconfigure the RECON data sets.
DELETE	An attempt was made to delete a record from the RECON.
INSERT	An attempt was made to create a new RECON record.
UPDATE	An attempt was made to update an existing RECON record.
UPGRADE	An attempt was made to upgrade the RECON data sets.
MODE-SW	An attempt was made to switch the accessing mode.

7.5 DBRC migration

You upgrade IMS Version 8 RECONS directly to IMS Version 10 by using the CHANGE.RECON UPGRADE command. For coexistence with IMS Version 8, apply the IMS Version 8 Small Programming Enhancement (SPE) APAR PK06145. It allows IMS Version 8 to use RECONS that have been upgraded to IMS Version 10. Similarly, for coexistence with IMS Version 9, you should apply the IMS Version 9 SPE APAR PK06147. It allows IMS Version 9 to use RECONS that have been upgraded to IMS Version 10. There is no support to upgrade RECONS from previous releases directly to IMS Version 10. You need to apply these APARs to IMS Version 8 or IMS Version 9 before its RECONS are upgraded to IMS Version 10.

You upgrade the RECONS to IMS Version 10 by using the DBRC CHANGE.RECON UPGRADE command with the IMS Version 10 DBRC utility (DSPURX00). You can run the upgrade while the RECONS are allocated to and being used by IMS Version 8 or IMS Version 9. Of course, these systems must be able to use IMS Version 10 RECONS. The upgrade checks the RECONS to ensure that any subsystems using the RECONS are capable of using IMS Version 10 RECONS. This check is performed by examining the SUBSYS records in the RECONS.

Certain IMS utilities, such as Change Accumulation, Log Archive, DSPURX00, HALDB Partition Definition Utility, and IMS Version 9 DBRC API applications do not create SUBSYS records. Thus, the upgrade cannot determine if they are running. You must ensure that any IMS utility that is running at the time of the upgrade has the appropriate maintenance (PK06145 or PK06147) that allows it to read IMS Version 10 RECONS.

IMS Version 10 has added the coexistence level to the RECON listing of subsystem records as we show in Example 7-35 on page 199. This has also been added to IMS Version 8 and

IMS Version 9 by the IMS Version 10 coexistence SPEs for these releases. The VERSION= field has existed in previous releases. It indicates the IMS release level of the subsystem. The COEXISTENCE LEVEL= field indicates if the coexistence maintenance for a later release has been applied. In this example, the IMS Version 10 DBRC coexistence maintenance has been applied to the IMS Version 9 system used by this subsystem. You could have produced this listing by an IMS Version 9 DBRC utility with the IMS Version 10 coexistence SPE applied or by the IMS Version 10 DBRC utility.

Example 7-35 RECON coexistence level

SSYS

SSID=IMS1 LOG START=07.229 17:25:44.2

SSTYPE=ONLINE ABNORMAL TERM=OFF RECOVERY STARTED=NO BACKUP=N

TRACKED=NO TRACKER TERM=OFF SHARING COVERED DBS=NO

IRLMID=**NULL** IRLM STATUS=NORMAL GSGNAME=**NULL**

COEXISTENCE LEVEL=10.1

AUTHORIZED DATA BASES/AREAS=4

VERSION=9.1

XRF CAPABLE=NO

ENCODED

-DBD-	-AREA-	-LEVEL-	-ACCESS INTENT-	-STATE-
PDHDOKA	**NULL**	0	UPDATE	6
PDHDOKB	**NULL**	0	UPDATE	6
PDHDOKC	**NULL**	0	UPDATE	6
PDHDOKD	**NULL**	0	UPDATE	6

The upgrade of the RECONS includes the reading of the subsystem (SSYS) records to ensure that these subsystems are running with the DBRC coexistence SPE. If not, the subsystem could not use the RECONS and the upgrade fails.

The upgrade changes a few records in the RECONS. The header and header extension records are changed to include the correct version indicator and changes to values for the parameters shown. The Cross DBRC Service Level ID (CDSLID) is set to the higher of the value in the RECONS before the upgrade and "1". The Image Copy records are rewritten to accommodate the larger IC type field. This field was expanded to include new image copy types for FlashCopy and fuzzy user image copies. The subsystem records are rewritten to accommodate the DBRC API flag for subsystems using the DBRC API.

Because only a few records are updated, a typical upgrade will be quick.

MINVERS is the parameter on the INIT.RECON and CHANGE.RECON commands that controls the minimum level of IMS that can use the RECONS. The minimum level of IMS that can use IMS Version 10 RECONS is IMS Version 8. If the previous MINVERS value was for '7.1', it is changed to '8.1' by the upgrade. Otherwise, upgrades do not change the MINVERS value.

MINVERS values of 81 and 91 (specified without the decimal point or the single quotation marks) are accepted by IMS Version 10 for compatibility; however, using the single quotation marks and the decimal point are required for specifying IMS Version 10 ('10.1') and recommended for specifying IMS Version 8 ('8.1') and IMS Version 9 ('9.1').

The IMSplex name is optional. It is required for Automatic RECON Loss Notification and parallel RECON access. The IMSplex name is specified with up to five characters. It is specified either in the IMSPLEX= execution parameter or by the DBRC SCI Registration exit routine. When first specified, it is stored in the RECONS. IMS Version 8 and IMS Version 9 store the IMSplex name as 'CSLxxxx' where 'xxxxx' is the value specified in the IMSPLEX= parameter or in the exit routine. When the RECONS are upgraded to IMS Version 10, the

value stored is 'xxxxxyyy' where 'yyy' is the DBRC Group ID. The upgrade sets the DBRC Group ID to '001' which is the default value.

IMS Versions 8, 9, and 10 list only the five characters of the IMSplex name in listings of the RECON header. These listing include a line with IMSPLEX=xxxxx when an IMSplex name has been stored in the RECONS. If there is no value stored, the line includes IMSPLEX=**NONE**. IMS Version 10 listings also include the DBRC Group ID on this line. If there is no IMSplex name, the Group ID is listed as GROUP=**NONE**. If there is an IMSplex name, the Group ID is listed as GROUP=yyy where yyy is the Group ID.

The DBRC SCI Registration exit routine (DSPSCIX0) can be used to specify the IMSplex name, as in previous releases, and the new DBRC Group ID. We recommend the use of the exit routine for users of IMSplex. It removes the requirement to specify IMSPLEX= for the execution of all IMS jobs that use DBRC. This includes batch jobs and utilities. With IMS Version 10, the exit also can specify the DBRC Sharing Group ID. This removes the requirement to specify DBRCGRP= for IMS executions.

IMS Version 8 and IMS Version 9 systems can tolerate the specification of the DBRC Group ID in the RECONS. DBRCGRP= is not a valid parameter on the EXEC statement for IMS Version 8 and IMS Version 9. When the exit routine is invoked in an IMS Version 8 or IMS Version 9 environment, the DBRC Group ID is not passed to it. The exit routine cannot specify the DBRC Group ID. Even though an IMS Version 8 or IMS Version 9 instance cannot specify the DBRC Group ID, it can join an IMSplex where IMS Version 10 instances are using DBRC Group IDs. The IMS Version 8 or IMS Version 9 instance will be passed all ARLN notifications from the IMSplex group. If an IMS Version 8 or IMS Version 9 system reconfigures its RECONS, its ARLN notification will be processed by all members of the IMSplex. This will include all IMS Version 10 systems. If there are multiple DBRC Groups, all members of all groups will process the notification. For these reasons, you must not use multiple DBRC Group IDs in an IMSplex while you are still using IMS Version 8 or IMS Version 9 systems.

The RECON status or header listing has some added and changed information.

Of course, the IMS version is listed as "Version 10R1." This means that the RECONS have been upgraded to IMS Version 10.

There is a new line that lists the type of RECON access, which is either SERIAL or PARALLEL. On the same line, the default for the DBRC LIST command, which is either STATIC or CONCURR, is shown.

On the line where the IMSPLEX value is shown, the DBRC Group ID value is also shown. In this example, these parameters have no values so "*** NONE **" is listed.

The sample listing shown in Example 7-36 includes the "CROSS DBRC SERVICE LEVEL ID." This also appears on IMS Version 9 RECON listings when the maintenance for APARs PQ98655 and PK01097 is applied and on IMS Version 8 RECON listings when the maintenance for APARs PQ98654 and PK01096 is applied. The service level ID is used to invoke functions that require a consistent level of maintenance on all IMS systems using the RECONS.

Example 7-36 IMS Version 10 RECON listing

```
RECON
RECOVERY CONTROL DATA SET, IMS Version 10R1
DMB#=231                               INIT TOKEN=07227F0536577F
NOFORCER LOG DSN CHECK=CHECK44        STARTNEW=NO
TAPE UNIT=3480      DASD UNIT=SYSDA    TRACEOFF  SSID=**NULL**
```



```

LIST DLOG=NO                      CA/IC/LOG DATA SETS CATALOGED=NO
MINIMUM VERSION = 8.1             CROSS DBRC SERVICE LEVEL ID= 00001
REORG NUMBER VERIFICATION=NO
LOG RETENTION PERIOD=00.001 00:00:00.0
COMMAND AUTH=NONE HLQ=**NULL**
ACCESS=SERIAL LIST=STATIC
SIZALERT DSNUM=15 VOLNUM=16 PERCENT= 95
LOGALERT DSNUM=3 VOLNUM=16

```

TIME STAMP INFORMATION:

TIMEZIN = %SYS

```

OUTPUT FORMAT:  DEFAULT = LOCAL  NONE  PUNC YY
                  CURRENT = LOCAL  NONE  PUNC YY

```

IMSPLEX = ** NONE ** GROUP ID = ** NONE **

-DDNAME-	-STATUS-	-DATA SET NAME-
RECON1	COPY1	IMSTESTS.DSHR.RECON1
RECON2	COPY2	IMSTESTS.DSHR.RECON2
RECON3	SPARE	IMSTESTS.DSHR.RECON3

Here we show the DBRC steps for migration to IMS Version 10:

1. Install IMS Version 8 or IMS Version 9 DBRC Migration/Coexistence SPEs.
2. Install IMS Version 10 DBRC Type 4 SVC.
3. The IMS Version 10 Type 4 SVC can be used with IMS Version 8 and IMS Version 9.
4. Upgrade RECONs using the IMS Version 10 SDFSRESL library.
5. Begin using IMS Version 10.
6. Discontinue all use of IMS Version 8 and IMS Version 9.
7. CHANGE.RECON MINVERS('10.1').
8. Full precision will be used in timestamps.

The first set of steps allows you to begin using IMS Version 10. The migration/coexistence SPE must be installed for the coexistence. The IMS Version 10 DBRC Type 4 SVC must be installed before you can use IMS Version 10. The upgrade of the RECONs to IMS Version 10 requires that you use the SDFSRESL library created by the installation of IMS Version 10. The upgrade using this library will be to the IMS Version 10 format. After the RECONs have been upgraded, you can begin using IMS Version 10. You can also continue to use IMS Version 8 or IMS Version 9.

After you have discontinued all use of IMS Version 8 and IMS Version 9, you can change the MINVERS value to '10.1'. This causes IMS to begin using the increased timestamp precision. Before changing MINVERS to '10.1', you must ensure that you specify full precision in the timestamps in the IMS utility control statements. The control statements generated by GENJCL statements will always generate control statements with the correct timestamps. Remember that the position of the timestamp in the control statements for the IMS Version 10 Change Accumulation and Database Recovery utilities does not depend on the MINVERS value. However, if MINVERS is not '10.1', the low order part of the timestamp does not matter, because these positions in timestamps are not recorded in the RECONs unless MINVERS('10.1') is specified.

Archived



Connectivity enhancements

This chapter describes the Open Transaction Manager Access (OTMA) and IMS Connect enhancements of IMS Version 10.

8.1 OTMA enhancements

IMS Open Transaction Manager Access (OTMA) enhancements address several high availability requirements, which include the following areas:

- ▶ Routing enhancements:
 - Destination routing descriptors
 - Resume transaction pipe (TPIPE) security
- ▶ Automatic flood detection and control of input messages
- ▶ Timeout control
- ▶ TPIPE storage cleanup
- ▶ Member level security
- ▶ OTMA security refresh by user ID
- ▶ Asynchronous message enhancements
- ▶ Enhanced OTMA display information
- ▶ New restart options
- ▶ Commit mode 0 ignore purge support

8.1.1 Routing enhancements

Prior to IMS Version 10, IMS systems that enabled OTMA and also produced alternate program communication block (ALTPCB) outbound messages for external destinations required system programmers to code several assembler OTMA routing exits including DFSYPRX0 and DFSYDRU0. This requirement often inhibited or delayed the adoption of new connectivity implementations, such as IMS Connect. These routing enhancements provide the support for the new IMS callout function.

Destination routing descriptors

IMS Version 10 introduces new OTMA destination routing descriptors that can eliminate the requirement to code the OTMA exits by externalizing the definitions and specifications that the exits provide. Note, however, that if the exits exist, they will be called with the routing information provided by the descriptors already set. Additionally, these descriptors have the ability to route from OTMA to non-OTMA destinations, such as SNA printers and terminals. Future support for WebSphere MQ is under consideration.

The new 'D' descriptor type for the DFSYDTx member of IMS.PROCLIB includes these keywords:

- ▶ TYPE=
TYPE determines if output is destined for IMS Connect (IMSCON) or non-OTMA (NONOTMA). This is a required keyword.
- ▶ TMEMBER=
TMEMBER specifies a 1 to 16 character client name. Required for TYPE=IMSCON. Ignored for TYPE=NONOTMA.
- ▶ TPIPE=
TPIPE specifies a 1 to 8 character TPIPE name. Optional for TYPE=IMSCON, defaults to Destination name. Ignored for NONOTMA.
- ▶ SMEM=
SMEM indicates if this destination is a super member. Optional keyword for TYPE=ICON, defaults to SMEM=NO. Ignored for TYPE=NONOTMA. If "YES", the name defined in the TMEMBER keyword becomes the super member name and can only be four characters.

► ADAPTER=

ADAPTER is the 1 to 8 character name of the IMS Connect Adapter to be used for the message, for example, an adapter for XML transformation. Optional for TYPE=IMSCON and ignored for TYPE=NONOTMA.

► CONVERTR=

CONVERTR is the 1 to 8 character name of the converter to be used by the adapter. Required for TYPE=IMSCON if ADAPTER is specified. Ignored for TYPE=NONOTMA.

Up to 50 lines can be used in the specification of a descriptor. Columns 1 through 10 must be the same for each line of a continuation. Descriptors are read and initialized at IMS startup. Multiple OTMA descriptors can be defined in the same DFSYDTx member.

Example 8-1 illustrates six descriptors, which we described.

Example 8-1 Example of DFSYDTxx member in IMS.PROCLIB

```
M HWSICON1          DRU=DFSYDRU0 INPUT=5000 T/0=5
D OTMACL99 TYPE=IMSCON TMEMBER=HWS1 TPIPE=HWS1TP01
D OTMACL*  TYPE=IMSCON TMEMBER=HWS2
D PRNTR3A  TYPE=NONOTMA
D SOAPGW1  TYPE=IMSCON TMEMBER=HWS2 TPIPE=HWS2SOAP
D SOAPGW1  ADAPTER=XMLADPTR CONVERTR=XMLCNVTR
D SOAPGW*  TYPE=IMSCON TMEMBER=HWS3 TPIPE=HWS3SOAP
D SOAPGW*  ADAPTER=XMLADPTR CONVERTR=XMLCNVT3
```

The descriptors in the example are:

► HWSICON1

HWSICON1 is a TMEMBER descriptor specifying the DRU exit for TMEMBER “HWSICON1”.

► OTMACL99

OTMACL99 is a destination routing descriptor for destination OTMACL99 that will be routed to IMS Connect TMEMBER “HWS1” with TPIPE “HWS1TP01”.

► OTMACL*

OTMACL* is a destination routing descriptor for destinations matching the mask “OTMACL*”. The messages will be routed to a TMEMBER of “HWS2” with a TPIPE of the destination matching the mask, for example, OTMACL04 if the destination using this descriptor was OTMACL04. Note that this along with the second descriptor illustrate that more specific destinations must be coded ahead of generic ones.

► PRNTR3A

PRNTR3A is a destination routing descriptor for destination “PRNTR3A” that will be routed to non-OTMA destination.

► SOAPGW1

SOAPGW1 is a destination routing descriptor for “SOAPGW1” that will be routed to IMS Connect TMEMBER “HWS2” with TPIPE “HWS2SOAP” and will result in XML translation.

► SOAPGW*

SOAPGW* is a masked destination routing descriptor for destinations that begin with “SOAPGW*”. This last descriptor will not be used for SOAPGW1, because the previous descriptor already specifically addressed that destination name. Note that all the destination matches for this descriptor will be routed to the one specified TPIPE “HWS3SOAP”.

To correctly enable the use of the new callout descriptors, application programmers must code the destination of an asynchronous output message switch (ALTPCB) to match the destination name of a corresponding descriptor. IMS systems programmers or system administrators, however, will also need to create the new descriptors in the DFSYDTx member of IMS.PROCLIB. Because descriptors can be masked (end in an “*”) and are searched in the order that they are coded, more specific descriptors should appear before generic ones.

The DFS2385E error message has been enhanced to catch syntax errors associated with the new descriptor keywords. The new error text includes:

- ▶ DESTINATION NAME NOT GIVEN OR BEGINS AFTER COLUMN 3
- ▶ DESTINATION NAME LONGER THAN 8 CHAR
- ▶ INVALID TYPE SPECIFIED
- ▶ DUPLICATE TYPE KEYWORD
- ▶ TMEMBER REQUIRED FOR TYPE=IMSCON
- ▶ TMEMBER MUST BE 1 TO 16 CHARACTERS LONG
- ▶ DUPLICATE TMEMBER KEYWORD
- ▶ TPIPE MUST BE 1 TO 8 CHARACTERS LONG
- ▶ DUPLICATE TPIPE KEYWORD
- ▶ SMEM MUST BE YES OR NO
- ▶ DUPLICATE SMEM KEYWORD
- ▶ ADAPTER MUST BE 1 TO 8 CHARACTERS LONG
- ▶ DUPLICATE ADAPTER KEYWORD
- ▶ CONVERTR MUST BE 1 TO 8 CHARACTERS LONG
- ▶ DUPLICATE CONVERTR KEYWORD
- ▶ CONVERTR REQUIRED WITH ADAPTER

The DFS2385E message is also issued if a previous release of IMS attempts to read DFSYDTx with the new descriptors.

Resume TPIPE Security

The Resume TPIPE Security enhancement addresses a security exposure that could occur when asynchronous output messages were retrieved by a Resume TPIPE request.

IMS transactions and commands that flow through OTMA from various clients are protected by current security classes, namely: TIMS and CIMS. The responses are guaranteed to be delivered to the client that initiated the transactions and commands. Output messages in the hold queue that are generated as a result of asynchronous processing, however, are not protected by any security class. When those messages are retrieved by an OTMA client using RESUME TPIPE, a security exposure can occur.

The function provided by Resume TPIPE Security protects these output messages by establishing a security class named RIMS within RACF or any non-IBM security product. Within this class, the security definitions are associated with the TPIPE name along with the list of user IDs or group names under this TPIPE. The enhancement, therefore, allows IMS installations to optionally authorize the user ID, together with the TPIPE name that is contained in the Resume TPIPE command message, before any of these messages are sent to a client.

OTMA also provides the DFSYRTUX security exit routine as an opportunity to overrule the SAF/RACF decision or to extend the security check to allow modifications as needed by the environment. OTMA security user exit routine DFSYRTUX is invoked after the call to SAF/RACF regardless of result, and it is always invoked if it exists regardless of whether or not RIMS is defined.

To take advantage of Resume TPIPE Security, the following actions are required:

- ▶ Define a new resource class, TPIPE name, and user IDs in RACF (refer to *z/OS V1R9.0 Security Server RACF Security Administrator's Guide*, SA22-7683) or applicable security server for the environment. The resource class comprises the resource class type of "R" and the resource class name whose value is taken from the RCLASS parameter of the SECURITY macro or in the DFSDCxxx member of PROCLIB. If RCLASS is omitted, the resource class name defaults to "IMS." The resulting class, then, is Rxxxxxxx where xxxxxx is the value of RCLASS or RIMS as the default.
- ▶ If needed, code, assemble, and bind the user exit DFSYRTUX in a library that is concatenated with IMS SDFSRESL under DD name STEPLIB or JOBLIB. The default routine always provides zero return code for compatibility and thus must be modified if further protection is required.
- ▶ Ensure that the OTMA client, for example, IMS Connect, is at the correct IMS Version 10 level to pass the TPIPE name in the OTMA CTL prefix and the user ID/group in the security prefix header as part of the message for the Resume TPIPE command.

Authorization is performed when Resume TPIPE is initiated, but before retrieving the messages from the hold queue. The authorization procedure comprises two security levels.

The first security level determines if resource class RIMS or Rxxxxxxx is defined in the system. If it is not defined, the messages in the hold queue can be sent to the user through the OTMA client only after the user exit is invoked. If it is defined, authorization logic not only verifies and validates the security header, but also authorizes the user ID under the TPIPE name using RACF facilities.

The second security level invokes the user exit regardless of the success or failure of the first security level. The user exit can either take the result of the first security level, override it, or enforce its own security rules.

When authorization is successful, output messages in the hold queue are returned to IMS Connect.

A rejection message of the Resume TPIPE command is sent to the client when authorization fails. A x'33' indicates that authorization failed either in RACF or in the user exit, DFSYRTUX. Several reason codes have been added to specifically identify the type of failure.

Resume TPIPE Security also takes advantage of the OTMA RACF Accessor Environment Element (ACEE) aging value capability. The OTMA ACEE aging value is used to refresh the user ID ACEE. In addition, the date and timestamp are stored in the user ID extension entry. This date and timestamp identify when the RACINIT command was last issued. The date and timestamp are checked whenever a user ID initiates a Resume TPIPE request. If the time exceeds the OTMA aging value, the user ID ACEE is considered expired, and RACROUTE is reissued for this TPIPE and user ID.

The implementation of Resume TPIPE Security could possibly impact existing clients. In previous releases, a user ID that was provided in the Resume TPIPE request was authenticated if IMS Connect was configured with security enabled. As long as the Resume TPIPE request, however, passed in the correct TPIPE name (client ID) then the associated asynchronous output messages could be retrieved.

In IMS Version 10, if the Rxxxxxxx | RIMS resource class is defined, security violations can occur where they previously did not. When retrieving the asynchronous output from IMS, the client signals IMS with a Resume TPIPE command and an IRM timeout value. This timeout value expires if there are no messages received by the client. Additionally, with the new support, a security check can result in success or failure. If the security validation is a

success, normal processing takes place. However, if a security violation occurs, a new NAK message is sent to the client.

8.1.2 Message flood detection and control

Message flood detection and control capability provides a mechanism to automatically monitor the growth of active input messages through OTMA and the control blocks associated with these requests. Specifically, when an OTMA member or client sends a transaction to IMS, OTMA internally creates a control block called the Transaction Instance Block (TIB) to track each active input message. For a send-then-commit (CM1) message, the control block is used for input and output processing after which the storage is freed or reused. For a commit-then-send (CM0) message, the control block is only used for input processing. If, however, several thousand OTMA input transactions are received and waiting to be processed, thousands of control blocks representing the requests could fill up LSQA storage below the line and possibly cause the IMS system to fail with an S40D abend. To prevent this type of OTMA message flood condition, OTMA supports the suppression or control of the input messages for OTMA based on a maximum value for the number of TIBs allowed for an OTMA member in the system.

Note: Message flood detection and control capability is also delivered in IMS Version 8 and IMS Version 9 via APARs PK04461 (V8) and PK04463 (V9).

By default, the message flood detection and control capability is always on and the maximum threshold value is set to 5000. To override this default, several choices are available:

- ▶ The OTMA descriptor DFSYDTx in the IMS.PROCLIB library can provide a value that IMS detects at initialization. It is not, however, until the TMEMBER associated with the descriptor is actually started that IMS implements the override value.
- ▶ At any time, an operator can issue the /START TMEMBER command that not only starts the member but also provides an override flood detection value that supersedes anything provided in the descriptor.
- ▶ As a member joins the group, the client-bid protocol message can also provide an override. This value is honored only if it is less than the value that is already in effect based on the overrides provided by the descriptor or /START TMEMBER command.

Note that the deactivation of the input message flood control capability can be requested by either the DFSYDTx descriptor or the /START TMEMBER command by specifying an INPUT value of 0. A client-bid message cannot override this specification.

We discuss the specifics of the implementation in the following paragraphs.

A new INPUT parameter is in the OTMA descriptor DFSYDTx in IMS.PROCLIB. The parameter allows values from 0 to 9999. If the value is set to 0, the capability for the message flood detection is disabled. Values between 1 and 200 are set to 200 and anything over 9 999 is set to 9 999.

The /STOP TMEMBER *member-name* | ALL command suppresses new input transactions or commands from a specific OTMA member or all OTMA members. This command does not affect the rest of communications between the stopped member and IMS. That means the following operations can still be performed for a stopped member: the client and server XCF connection remains unchanged; all of the running transactions currently scheduled in the IMS can still be processed, and the responses can be delivered; IMS conversational transactions can continue processing the existing conversation until it ends; all of the OTMA protocol commands including ACK/NACK can still be processed by OTMA. Note that after this

command has been issued, a STO-INPUT status will be displayed on a subsequent /DISPLAY TMEMBER command for the appropriate OTMA members.

The /START TMEMBER INPUT command has the same range and meaning of the values 0 to 9 999.

There are new fields in the client-bid protocol interface between the OTMA member and OTMA. For the client-bid protocol message, a new 2-byte field in the offset x'3E' of the state data can specify the maximum number of active input messages for the member. The value can be specified between 0 and 9 999. If the value is 0, OTMA will take the default of 5 000 for the maximum active input message limit (note that specifying 0 does not turn the capability off). If the value is between 1 and 200, it will be treated as 200. If it is over 9 999, it will be set to 9 999. A further restriction is that the specified request must be less than any value previously set by a descriptor or command. If the client-bid tries to send in a higher value, it will be reset to the value established by the descriptor or command. Additionally, a new flag TMAMMAXI x'80' in the offset x'2D' of the state data needs to be set to inform the OTMA server that the maximum active input message checking function is requested. During processing, if OTMA detects that the active input message limit has been reached, a new NAK code of x'30' will be sent the OTMA member.

OTMA monitors the growth of the active input messages from members. A warning message, DFS1988W, is sent to the console to indicate that the input message buildup is approaching the maximum limit. The message will be sent starting at 80% of the message limit and every 5% thereafter. When the maximum limit is reached, an error message, DFS1989E, is sent to the console and any subsequent OTMA input messages are rejected with a new OTMA sense code x'30'. After the situation has been alleviated, DFS0767I is issued.

After the warning message, DFS1988W, is issued, problem determination should begin. The /DISPLAY TMEMBER TPIPE command shows the message counts and gives an indication of the severity of the queue buildup. The /DISPLAY TMEMBER *member-name* command can also be issued to show the value in effect for the maximum input message limit. Additionally, the IMS system should be reviewed to determine whether the dependent regions are active to process input messages; whether any IMS resources have been stopped; whether just one or multiple OTMA client members are flooding IMS, and so forth. It might be necessary to control or stop one or more OTMA client members to prevent the flooding. The /STOP TMEMBER command can be used to inhibit further input from a specific OTMA member. If the limit was originally set too low, the TMEMBER can be started with a higher INPUT value.

8.1.3 Send-then-Commit (CM1) ACK timeout control

OTMA provides a new timeout control capability for CM1 (Send-then-Commit) interactions. For an OTMA send-then-commit (CM1) response message with synclevel=confirm or synclevel=syncpt, IMS expects an ACK/NAK from the OTMA client. Due to the possibility of a client programming error or a network failure or delay, the expected ACK/NAK might not be received by IMS. A missing or delayed ACK/NAK results in a "wait-syncpoint" condition for the IMS dependent region that processed the OTMA transaction. To resolve this situation, OTMA has been enhanced to detect this "wait-syncpoint" condition and take an appropriate timeout action. The default timeout value is 120 seconds.

The default 120 second timeout value for CM1 (send-then-commit) messages can be overridden in several ways:

- ▶ During IMS initialization with a new parameter in the OTMA descriptor member DFSYDTx. It is not until the member is actually started that the override value is honored.
- ▶ At any time with a new TIMEOUT specification in the /START TMEMBER command.

- By OTMA member request either when the member joins the group using the client-bid protocol message or, at a lower level of granularity, whenever a CM1 (send-then-commit message) flows into OTMA.

If none of these methods is used to set the timeout value, the OTMA default of 120 seconds is used to determine when to perform the timeout action.

If needed, the /START TMEMBER TIMEOUT command and the OTMA descriptor can deactivate the OTMA timeout function by specifying a timeout value of 0. After the function is deactivated, OTMA will not perform the timeout action. However, OTMA will still detect a long-waiting dependent region for a missing ACK or NAK and issue the following warning message: DFS0808W IMS REGION *region-id* IN BACK-END IMS *aaaa* HAS BEEN IN [WAIT-SYNCPPOINT] or [WAIT-RRS] FOR *otma-membname/tpipename* FOR OVER *xxxx* SECONDS.

The details of the implementation are:

- There is a new T/O parameter in the OTMA descriptor member DFSYDTx in IMS.PROCLIB. The new parameter defines the timeout value in seconds for OTMA send-then-commit response messages. The value specified can be between 0 and 255. If the value is 0, OTMA will deactivate the timeout function.
- There is a new TIMEOUT specification in the /START TMEMBER command.
- There is a new specification in the OTMA member request either when the member joins the group using the client-bid protocol message or, at a lower level of granularity, whenever a CM1 (send-then-commit message) flows into OTMA:
 - A 1-byte timeout field at offset x'41' specifies a timeout value; value must be less than the value set by IMS Command or OTMA descriptor
 - Flag TMAMTMOT x'20' at offset x'2D' specifies that timeout is requested

If a client-bid protocol message is used to specify the timeout value, the criteria of choosing the timeout value is provided through new flag specifications. Note that the client-bid cannot override the timeout specified by an OTMA descriptor or command. If the client-bid timeout value is equal to or greater than the current timeout value set by the descriptor or command, OTMA will ignore the timeout value in the client-bid message. If the client-bid timeout value is less than the current timeout value set by the command or descriptor, the value from the client-bid will be used for the timeout action for this member.

On an individual message basis, additional flags have been provided for an even lower level of timeout specification:

- A 1-byte reserved field at offset x'1E' of the message control information prefix
Specifies timeout value for the input transaction
- New flag TMAMTTMO, x'08' in byte 5 of the state data section
Specifies that OTMA can take the message level timeout value specified in the control data

This capability supports a message timeout value, which can be different from the timeout value set for the entire member. Note, however, that this value follows similar restrictions in that it cannot override the member timeout value set by an OTMA descriptor or IMS command.

When an OTMA timeout occurs, OTMA will first back out the transaction in order to get out of the wait-syncpoint or wait-RRS condition for any missing ACK/NAK region. Subsequently, an OTMA CM1 deallocation message will be sent to the member with the existing ABORT flag and the new "timeout" flag. The IMS system console operator will also receive a DFS0809E

message IMS REGION *region-id* IN BACK-END IMS *aaaaaaaa* HAS TIMED OUT FOR *otma-membername/tpipename* FOR *xx* MINUTES. When OTMA takes the timeout action, byte 3 TMAMCCCI, of the OTMA commit-confirmation flag in the message control data prefix is set to TMAMCTMO, X'08', to indicate that the transaction was aborted due to the OTMA timeout condition.

8.1.4 TPIPE storage cleanup

TPIPE storage cleanup provides a more efficient way to control unused storage associated with idle TPIPEs. Transaction Pipes (TPIPEs) are OTMA control blocks that represent logical connections between the client and IMS. They are analogous to an IMS logical terminal (LTERM) and allow IMS to associate all input and output with a particular OTMA client. After they are created, they occupy storage whether or not they are used again. This cleanup enhancement determines whether an inactive TPIPE can be deleted and its storage released. A TPIPE is considered inactive if it has been idle for two consecutive checkpoints.

Note: This capability is available also in IMS Version 8 and IMS Version 9 via APARs PQ99983 (V8) and PK00386 (V9).

The TPIPE storage cleanup function is activated when OTMA is activated in an IMS system. Logic added to IMS system checkpoint processing determines whether a TPIPE is active or idle based upon whether there are any input or output messages associated with the control block in addition to whether any TPIPE status conditions exist. TPIPEs that are idle for two consecutive checkpoints are deleted.

Active TPIPEs include those that are processing commit-then-send (CM0) messages in a Shared Queues environment, have incomplete send-then-commit (CM1) messages, or have queued commit-then-send (CM0) output messages.

Certain TPIPEs are never considered for removal. These include synchronized MQ TPIPEs and TPIPEs with outstanding status indicators.

8.1.5 Member level security

Prior to IMS Version 10, OTMA did not allow different security levels defined for various members. The security setting requested is considered a system-wide setting for all OTMA members. In IMS Version 10, the OTMA command, /SECURE OTMA, has been enhanced to allow specification of member security so that each OTMA client can have its own security level.

Note: Messages are always processed with the security level that was in effect when the message was received. Even if a new security level is introduced by command, the security level associated with the message is based on the level in effect at the time of the message receipt.

The addition of the TMEMBER keyword to the /SECURE OTMA command provides the ability to define any of the security options for a specific OTMA member.

The /DISPLAY command has also been enhanced to provide a mechanism to display the security option in effect for a specific member.

8.1.6 OTMA security refresh by user ID

Small Programming Enhancement (SPE) PK41554 allows clients to refresh OTMA cached ACEEs for a specific user ID across all TMEMBERs using the /SEC OTMA REFRESH USER *user ID* command.

For performance purposes, when OTMA security is enabled, OTMA caches RACF User Profiles as a RACF Accessor Environment Element (ACEE) in memory. When changes are made to a User Profile in the RACF database on DASD, the changes are not reflected in the cached ACEE until the ACEE is refreshed. OTMA allows its clients to specify an aging value that indicates how long the cached copy should be used before being automatically refreshed. Clients also can refresh ACEEs by issuing the /SECURE OTMA REFRESH command globally against OTMA or against individual OTMA clients. There is now a USER parameter to the /SECURE OTMA REFRESH command to allow clients to refresh cached ACEEs by user ID, so that only the ACEE for that specific user ID is refreshed.

The USER user ID parameter is added to the /SECURE command as follows:

```
/SECURE OTMA REFRESH USER user ID
```

When USER is specified, OTMA refreshes across all TMEMBERs only the ACEEs that include the specified user profile.

If your RACF Security Administrator modifies the user's access privileges for OTMA in RACF, the user's old access privileges might remain unchanged in online memory unless the RACF ACEEs cached by OTMA are refreshed, either automatically when the ACEE aging limit is reached or manually by issuing the /SECURE OTMA REFRESH command.

To ensure that the cached ACEEs are refreshed promptly, the RACF Security Administrator should inform the IMS operator to refresh all OTMA cached ACEEs. The RACF Security Administrator must also provide the RACF User Profile name that must be entered in the USER parameter of the /SECURE OTMA REFRESH USER command. When the IMS operator issues the /SECURE OTMA REFRESH USER command, IMS scans all OTMA cached ACEEs and refreshes only those that contain the specified user ID.

8.1.7 OTMA asynchronous message enhancements

Several enhancements are available to address asynchronous commit mode 0 (CM0, commit-then-send) messages in OTMA environments that use IMS Shared Queues, as well as those that implement load balancing or IP spraying techniques, such as Sysplex Distributor or the WebSphere Edge Server. These enhancements extend the super member, purge, and reroute capabilities of IMS Connect.

Super Member capability

The Super Member capability in OTMA facilitates the delivery of asynchronous (CM0) messages by any instance within a set of IMS Connects and IMS subsystems. With this function, affinity to a particular IMS or IMS Connect is removed and the use of Shared Queues, as well as solutions, such as Sysplex Distributor, become more viable.

Note: Also delivered in previous releases:

IMS Version 9 and IMS Connect: PK09946, PK30086, PK10911

IMS Version 8: PK09944, PK30103, IMS Connect V2.2: PK10910

A *super member* is a special OTMA member name that can be shared by a set of IMS Connects to handle the CM0 hold queue messages.

When an IMS Connect attaches to IMS, OTMA creates a regular member structure name unique to that instance to track the connection status and to record the connection options for later transaction processing. If a super member name, which can be considered a group name for a set of IMS Connects using Sysplex Distributor or a similar mechanism, is given by an IMS Connect during the connection time, OTMA will additionally create a member structure called the *super member structure*. If the super member structure already exists, OTMA will use it instead of creating a new one. A regular member structure is dedicated exclusively to the IMS Connect for which it was created. However, a super member structure is shared among a set of IMS Connects so that a Resume TPIPE can be issued from any IMS Connect. The role of the super member is to store and deliver the CM0 hold queue messages.

IMS Connect has been enhanced to support the Super Member capability. The HWSCFGxx configuration file member provides a new SMEMBER= parameter in the HWS statement. The value provided must be different than the value provided in the MEMBER= parameter of the DATASTORE statement and must follow the OTMA naming conventions.

When the client-bid protocol message is sent from an IMS Connect member that has specified a super member value, the state data of the OTMA prefix carries the defined SMEMBER value in a new 4-byte field at the offset of x'36' in the state data section. In addition to the super member name in the client-bid message, a new flag TMAMFGSM (x'08') at the offset x'2D' of the state data informs OTMA that the super member processing is requested.

Additionally, the /DIS, /TRA, /STA, and /STO commands have been enhanced to support the Super Member capability. The /DISPLAY OTMA command output includes a new SMEM column to display the super member name if one exists. The /TRA TMEMBER ... TPIPE, /STA OTMA, and /STO OTMA commands can be issued against either a super member name or a regular IMS Connect.

The value of the Super Member support

Asynchronous CM0 output messages in IMS are queued to a message queue construct that is identified by TMEMBER and TPIPE and, therefore, associated with a specific IMS Connect instance. As a result, there is a potential issue when using any of the load balancing or sysplex distribution mechanisms or even when a message is processed in a back-end IMS in a Shared Queues group. To retrieve the message, the remote program needs to establish a connection through the appropriate IMS Connect to the actual IMS system that queued the message. This can be a challenge because there is no easy mechanism for a remote program to discover the required connection path, that is, a specific IMS Connect to a specific IMS. Additionally, the remote programs might not want to know specific connection paths to IMS, because that negates the value of using load balancing and distribution mechanisms. Figure 8-1 on page 214 shows this scenario.

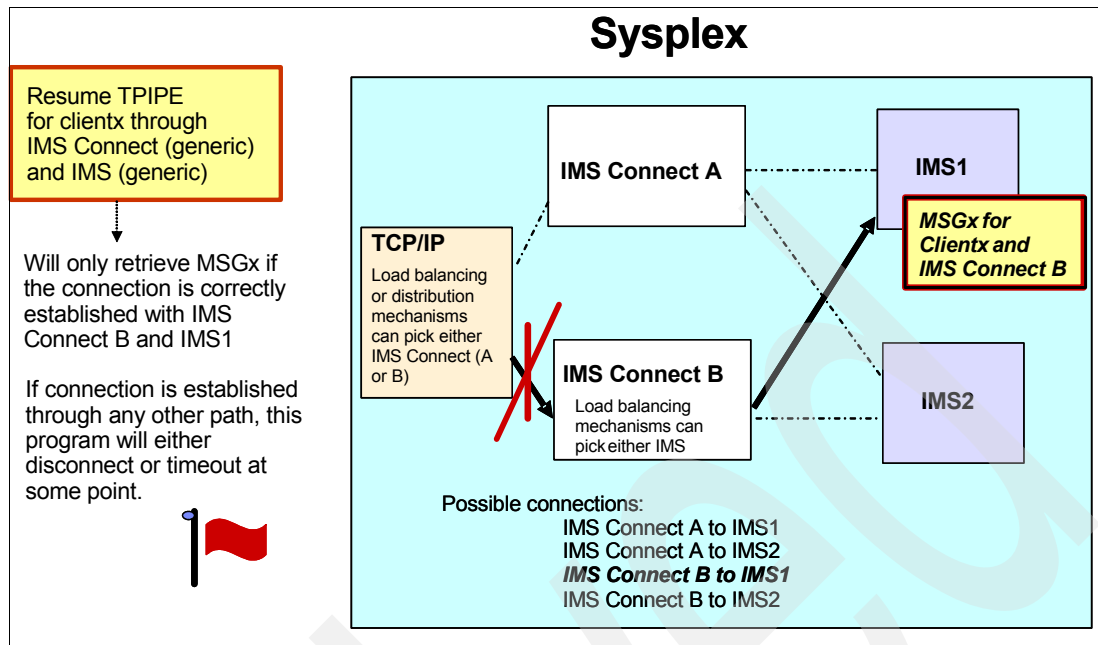


Figure 8-1 A challenge for retrieving the asynchronous messages

The OTMA Super Member function resolves the issue by allowing a RESUME TPIPE request to retrieve CM0 output across all combinations of IMS Connect and IMS systems. If multiple IMS systems are involved, those IMS systems must also have IMS Shared Queues implemented. If there is only one IMS system but multiple IMS Connects, then Shared Queues support is not required.

As shown in Figure 8-2 on page 215, the Resume TPIPE request for clientx can be routed through any load balancing or distribution mechanism to either IMS Connect A or IMS Connect B. Both systems are identified to IMS1 and IMS2 by their unique XCF member names, as well as the global Super Member name of SM01. The request to retrieve the output message for TPIPE clientx can be sent to either IMS1 or IMS2 because both have access to the Shared Queues and, more specifically, to all the messages under the Shared Queues construct for SM01 and TPIPE clientx.

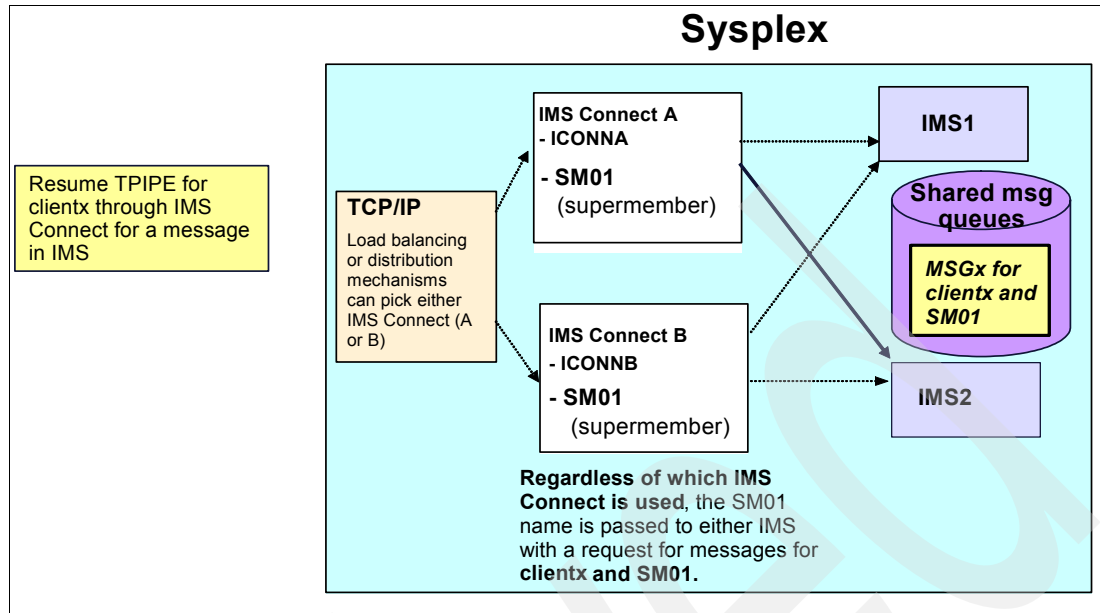


Figure 8-2 The challenge is resolved with the OTMA Super Member support

Purge and reroute support

OTMA asynchronous message enhancements have been extended to provide greater flexibility for messages that cannot be delivered or are rejected by the remote client. The two actions, which are mutually exclusive, are to reroute messages that cannot be delivered to an alternate destination or to purge them.

The reroute and purge support is only for OTMA hold queue-capable OTMA clients, for example, OTMA members, such as IMS Connect, that keep asynchronous messages on the IMS message queue until requested by the remote application.

The reroute enhancement provides a mechanism for an IMS Connect client to request that undeliverable Commit Mode Zero (CM0) output associated with a send/receive from the client application be rerouted to an alternate IMS Connect destination. When a Client Reroute Request is made, IMS Connect notifies OTMA to remove the message from the current queue and requeue it to the provided Reroute name queue (TPIPE).

The Purge Not Deliverable extension allows the remote client application or the appropriate User Message Exit to specify whether the output should be purged if it is undeliverable.

Note that if both capabilities are requested, neither action is performed and the DFS2407W message is issued.

Note: The reroute and purge support is also delivered in previous releases via the following APARs:

- ▶ PK16934, PK22480, PK24907, PK09543, PK12013 (IMS Version 9 and IMS Connect)
- ▶ PK21868, PK09542 (IMS Version 8), PK12012 (IMS Connect 2.2)

Both the OTMA headers and the IMS Connect headers have been enhanced to request either the PURGE or REROUTE capability:

- ▶ OTMA header enhancement for reroute support:
 - OMUSR_FLAG1
New setting of OMUSR_REROUT (X'01')
 - OMUSR_ARCLEV
New setting of OMUSR_AL02 (X'02')
 - Field OMUSR_REROUT_NM
Holds the reroute name
- ▶ IMS Connect header enhancement for reroute support:
 - IRM_ARCH
New setting of IRM_ARCH1 (X'01')
 - IRM_F3
New setting of IRM_F3_REROUT (X'08')
 - New field IRM_REROUT_NM
Holds the reroute name
- ▶ OTMA header enhancement for purge support:
 - OMHDRPND EQU X'10'
Purge if not deliverable
- ▶ IMS Connect header enhancement for purge support:
 - IRM_F3 has a setting of IRM_F3_PURGE (X'04')

IMS Connect implementation for OTMA purge and reroute support

IMS Connect has been enhanced to take advantage of the new capabilities. The Resume TPIPE protocol provides flags to indicate one or the other type of request.

Another enhancement in this area allows RESUME TPIPE requests to be queued when requested for the same TPIPE name.

Additionally, the Send-Only protocol allows specification of a reroute queue name for the output in the case of an initial rejection of the output reply. This capability allows IOPCB output from the Send-Only input transaction to be rerouted to a dedicated TPIPE instead of the inputting TPIPE. The user of the IMS Connect Send-Only transaction will need to turn on the reroute flag and specify a reroute TPIPE name in the input stream to activate the capability. Note that this is not supported by the local option capability, HWSIMSO0, HWSIMSO1, and HWSJAVA0. Reroute queue name support is also delivered in IMS Version 9 via APARs PK17421 and PK18555.

Note: IMS Version 9 was the last release to support the following user message exits: HWSIMSO0 and HWSIMSO1. Clients using these exits need to migrate to HWSSMPL1.

8.1.8 OTMA /DISPLAY command enhancements

The /DISPLAY OTMA and /DISPLAY TMEMBER command outputs have been enhanced to provide more information about the OTMA environment and specific TMEMBERs. To contain all the information, the single line display output has been increased to two lines.

Several new USER-STATUS indicators provide the following information:

- ▶ **SMQ BACKEND**

This status indicator on a TMEMBER line shows up on a back-end IMS in a Shared Queues group. It shows that OTMA has duplicated the specific TMEMBER environment and control blocks needed on the back-end to process a message that was received from that TMEMBER which is attached on the front end. The same member name on the front-end IMS is shown in “connected” state with no SMQ BACKEND indicator. Whereas the corresponding control blocks on the back-end IMS which processes the message is shown in “disconnected” state with an SMQ BACKEND indicator.

- ▶ **STO-INPUT**

This status shows that the /STOP TMEMBER command has been issued for a specific member-name and no new input can be accepted.

- ▶ **FLOOD**

This status indicator shows that a specific TMEMBER is in a message flood condition and that the maximum input message count that was specified has been reached.

Additionally, the /DIS TMEMBER TPIPE command has been enhanced to show the number of input messages currently on the queue under the new column INPCT. The value in this column wraps after 65 535.

8.1.9 New OTMA startup option

In addition to the existing OTMA values of Y and N, IMS Version 10 introduces the option of OTMA=M (manual). When IMS first initializes, the value of OTMA=M functions similarly to OTMA=N in that OTMA is not started. After IMS is up and running, the /STA OTMA command can be issued but becomes non-recoverable. The setting of OTMA=M, therefore, takes effect when IMS terminates either normally or abnormally and has to be restarted. During restart processing, OTMA is not automatically restarted. This capability was introduced to prevent looping abend situations where IMS might have terminated as a result of OTMA error conditions.

An additional impact on OTMA restart processing is introduced for environments where IMS is initialized with OTMA=N. If a /START OTMA NOCHECK command is issued, the command is also not recovered during either a warm start or an emergency restart.

Note: This option is also delivered in IMS Version 8 and IMS Version 9 via APARs PK14679 (V8) and PK14680 (V9).

8.1.10 Commit-then-Send (CM0) Ignore Purge support

OTMA has always handled multiple output messages that are inserted to the IOPCB differently for CM1 and CM0 specifications. This affects programs that issue repeated iterations of ISRT followed by PURGE calls to the IOPCB. Ordinarily, a PURGE call triggers the delineation between output messages. With OTMA, multiple ISRT/PURGE combinations for CM1 interactions result in one multi-segment output message. The same scenario, when invoked in a CM0 mode, results in multiple output messages, one corresponding to each PURGE call.

Remote programs that are coded to use one or the other mode, CM1 as opposed to CM0, need to be aware of this anomaly. IMS Version 10 provides a new TMAMIPRG bit setting in the TMAMHCFL flag byte of the OTMA header to request that the IOPCB output is processed

as one message even when the mode is CM0 and multiple ISRT/PURGE iterations have been issued. This enhancement allows remote programs to consistently expect only one output message and then parse the information to determine if the message contains more than one segment.

To take advantage of this change, remote applications that use CM0 will need to be coded accordingly. The IMS TM Resource Adapter client has added a new `ignorePURGCall` property to the `IMSInteractionSpec` with default value of *false*. An override specification of *true* is only valid for `SYNC_SEND_RECEIVE` interactions on shareable persistent socket connections.

8.2 OTMA enhancement migration considerations

The following considerations address issues that you must consider when migrating from a previous release of IMS. The assumption is that migration to IMS Version 10 is based on existing functionality without adding any new capabilities during the migration process:

- ▶ The message flood control enhancement in IMS Version 10 is automatically enabled with a default limit of 5 000 messages. To provide compatibility with previous releases and deactivate the support, either specify an `INPUT` value of 0 in a descriptor or issue the `/STA TMEMBER INPUT` command.
- ▶ Likewise, the timeout support for the synchronous CM1 message is automatically enabled with a default value of 120 seconds. To provide compatibility with previous releases and deactivate the support, specify a `T/O` value of 0 in a descriptor or issue the `/STA TMEMBER TIMEOUT` command.
- ▶ Note that the `/DISPLAY` command output associated with OTMA and TMEMBER requests has been expanded to two output lines and includes new information. As a migration consideration, this is a key issue for automated operations.

8.3 OTMA enhancement benefits

This section summarizes the benefits of OTMA enhancements:

- ▶ Destination routing support:
 - Facilitates outbound messages through an architected OTMA layer
 - Minimizes the need for coding OTMA exit routines through descriptors
 - Enables growth of e-business environments using the IMS TM Resource Adapter and the IMS SOAP Gateway
- ▶ Resume TPIPE Security:
 - Ensures that only authorized users can retrieve output messages in the hold queue
 - Supports IMS Connect clients using Resume TPIPE commands
- ▶ Message Flood protection:
 - Provides a mechanism to prevent OTMA clients from flooding the IMS message queues and causing S40D abends
- ▶ Timeout:
 - Allows automatic detection and resolution of “hung” conditions identified as wait-syncpoint or wait-RRS

- ▶ TPIPE cleanup:
 - Greater efficiency of storage usage for OTMA control blocks
- ▶ Member level security:
 - Greater control of the security environment for OTMA
 - Allows different options on a member level
- ▶ Asynchronous message enhancement:
 - Super Member support
 - Ensures delivery of CM0 messages in a Shared Queues environment or with the use of IP load balancing techniques
- ▶ Purge and Reroute:
 - Provide greater control of output messages that cannot be delivered to the original destination
- ▶ /DIS OTMA and /DIS TMEMBER output display enhancements:
 - Provide more comprehensive information about the environment
- ▶ OTMA=M option and /STA OTMA with NOCHECK option:
 - Protect IMS from restarting an OTMA system that has caused IMS to terminate abnormally
- ▶ CM0 Ignore Purge support:
 - Greater consistency for the way IOPCB output messages are delivered

8.4 IMS Connect enhancements

IMS Connect provides several usability, availability, and security enhancements in the following areas:

- ▶ ACEE aging value support
- ▶ Client password change request
- ▶ RACF mixed case password
- ▶ Send-then-Commit (CM1) ACK timeout control
- ▶ Message flood control
- ▶ Asynchronous message enhancements:
 - Super member, Reroute, and Purge Not Deliverable
 - Port affinity
 - Alternate client ID
- ▶ XML Adapter support
- ▶ IMS service-oriented architecture (SOA) Composite Business Application Support

8.4.1 ACEE aging value support

The Access Control Environment Element (ACEE) is a control block that represents a verified user ID to IMS. The ACEE is used to determine the user's authorization to the IMS command or IMS transaction requested in the input message. When built in OTMA, the ACEE for each user ID is cached and the aging value associated with each OTMA client, for example, IMS Connect, is kept in a table. The aging value is then used to determine when the cached control block expires and is refreshed. IMS recreates the ACEE if a message associated with the user ID is received but the age of the current ACEE is greater than the aging value. The aging value is used to balance performance (possible RACF I/O to refresh the ACEE) and

integrity. For IMS Connect, the ACEE expiration value is specified during the client-bid process and is set to a default of no expiration.

IMS Version 10 provides a new parameter, OAAV, in the DATASTORE statement of the HWSCFGxx file for specification of an OTMA ACEE aging value. If not specified, the default continues to be 2147483647 which, in essence, means no expiration and is the maximum value supported by OTMA.

The VIEWHWS, VIEWDS, QUERY MEMBER, and QUERY DATASTORE command output displays have all been enhanced to show the aging value that is in effect for the associated environment.

8.4.2 Client password change request

IMS Connect provides a new mechanism that allows a remote client to request that the SAF/RACF password associated with a user ID is changed.

As provided, the new capability will be supported in the HWSSMPL0, HWSSMPL1, and HWSJAVA0 exit routines. The routines check for a leading keyword of "HWSPWCH" to determine whether it is a request to change the password. This "HWSPWCH" string can be viewed as a transaction code but new logic in the routine, HWSPWCH0, is called to process the special request. HWSSMPL0, HWSSMPL1, and HWSJAVA0 can be modified to define the HWSPWCH constant with a unique keyword value other than "HWSPWCH."

The exit routines pass HWSPWCH0 the request keyword length in IMSEA_PWCHKEYL field. This allows HWSPWCH0 to process the password change request independently from the request keyword. After regaining control, the exit routines check the return code in register 15. A zero return code means the request is successful. If it is a non-zero return code, the exit routine checks the IMSEA_ERCD field for a valid error code, and replies back to the client with an error message specified in the IMSEA_MSGTEXT and IMSEA_MSGLEN fields.

In order to establish the HWSPWCH0 address, the HWSPWCH0 object code must be included in the exit routine and an "INCLUDE TEXT(HWSPWCH0)" statement added to the exit routine JCL for the binder (link-edit) step. During execution, if a request for password change is received and the HWSPWCH0 address does not exist, the exit routine will send a message back to the client stating that the password change function is not supported.

The password change support in HWSJAVA0 can be invoked by the IMS TM Resource Adapter (formerly called IMS Connector for JAVA or IC4J) client. After the OTMA headers, the message sent begins with the defined keyword HWSPWCH followed by the old and new passwords.

Similarly, other clients that invoke exit routines based on HWSSMPL0 or HWSSMPL1 can supply the HWSPWCH request after the IRM header.

As a result of processing the password change request, IMS Connect sends a message back to the client indicating success or, if a failure, the reason for the rejection. The client receives one of the following messages:

- ▶ Password change successful
- ▶ Invalid error code X
- ▶ Password change function not supported
- ▶ RACF is not active
- ▶ The password is not authorized
- ▶ The password has expired
- ▶ The new password is invalid

- ▶ Password change error
- ▶ Invalid old password specified
- ▶ Invalid new password-1 specified with non-alphanumeric or lower case character
- ▶ Invalid new password-2 specified
- ▶ New passwords do not match
- ▶ Invalid request syntax

8.4.3 RACF mixed case password

The support for a RACF mixed case password in IMS Connect is aligned with the IMS Version 10 support for mixed case passwords. The capability in IMS Connect allows the password to be preserved exactly as the remote client provided it and passes the string to RACF without translation to upper case.

To turn on this function:

- ▶ A new parameter, PSWDMC= in the HWS= statement, defines the option of mixed case passwords where PSWDMC=N is the default. This setting can be changed using the IMS Connect SETPWMC or UPDATE command.
- ▶ IMS Connect also provides a command, SETPWMC, that can override the HWSCFGxx specification.
- ▶ The PSWDMC keyword is also available to the IMS Connect UPDATE command as another way to request the support.

The IMS Connect support requires that RACF enable mixed case passwords through the RACF SETROPTS(MIXEDCASE) command. Note that the RACF enablement of this support does not constitute the IMS Connect usage of this support. Also note that the mixed case support for IMS Connect can only take effect when RACF is enabled.

8.4.4 Send-then-Commit (CM1) ACK timeout control

IMS Connect supports the new OTMA timeout control function for send-then-commit CM1 interactions. OTMA provides a default value of 120 seconds after which transactions that are held in *Wait-Syncpoint* or *Wait-RRS* status are released and backed out. If provided, the value in the ACKTO parameter of the IMS Connect configuration DATASTORE statement is passed to OTMA during client-bid processing.

Note that a specified value of 0 is reset to 120. If the timeout value is zero or not specified, a value of 120 seconds as an OTMA ACK timeout default value will be set. If the specified value is less than zero or greater than 255, an abend U3401 will be issued.

The value cannot be greater than what is defined in OTMA. If the ACKTO value is equal to or greater than the timeout specified in IMS by an OTMA descriptor or /STA T MEMBER command, OTMA will ignore the IMS Connect request. However, if the ACKTO value is less than the current timeout value set by the OTMA descriptor or command, the value that is passed to IMS by the IMS Connect client-bid process will be used for the timeout action for this member. Also note that IMS Connect cannot disable the OTMA ACK timeout support. Disabling the capability can only be done in OTMA

IMS Connect command output has been enhanced to display the CM1 timeout value. The applicable commands include: VIEWHWS and VIEWDS output command and the MVS MODIFY command for QUERY MEMBER and QUERY DATASTORE.

8.4.5 Message flood control

IMS Connect also takes advantage of the OTMA Message Flood Control capability to automatically monitor the growth of active input messages. OTMA provides a default value of 5 000 messages after which input messages from a specific IMS Connect instance will be rejected. If provided, an override value in the MAXI parameter of the IMS Connect configuration DATASTORE statement is passed to OTMA during client-bid processing.

Note the following considerations:

- ▶ A specified value of 0 is reset to 5 000.
- ▶ A value between 0 and 200 is reset to 200.
- ▶ Any value specified in error between 9 999 and 65 535 is reset to 9 999.
- ▶ Any value outside the 0 - 65 535 range results in an abend U3401.

The value cannot be greater than what is defined in OTMA. If the MAXI value is equal to or greater than the INPUT value specified in IMS by an OTMA descriptor or /STA T MEMBER command, OTMA will ignore the IMS Connect request. However, if the MAXI value is between 1 and 200, a value of 200 will be sent to IMS. If the IMS Connect value is less than the value specified in OTMA, then the IMS Connect MAXI value will be used. A specification of 0 will be ignored, because IMS Connect cannot disable the OTMA message flood control support. Disabling the capability can only be done in OTMA.

8.4.6 Asynchronous message enhancements

There are several enhancements for commit mode 0 processing that are described in detail in the OTMA enhancements section, and it also includes the specifics for IMS Connect applicable enhancements such as:

- ▶ Super member support
- ▶ Reroute
- ▶ Purge

Refer to 8.1.7, “OTMA asynchronous message enhancements” on page 212 for the details.

8.4.7 Resume TPIPE enhancements

Resume TPIPE processing has been enhanced in the following areas:

- ▶ Port affinity
- ▶ Alternate client ID
- ▶ Security

We have covered the Resume TPIPE Security enhancements earlier in 8.1.1, “Routing enhancements” on page 204 and we describe the other two items in this section.

Port affinity

Using concurrent Resume TPIPE connection requests of the same client ID across several ports can cause problems, such as keeping an IMS OTMA TPIPE in WAIT-H status. To address the issue and support this concurrency requirement, IMS Connect provides a new parameter to enforce all the correlated interaction, such as the retrieval of a message and associated ACK or NAK, to the same remote client instance. The PORTAFF parameter in the

TCPIP statement controls whether commit-then-send (CM0) output messages sent by IMS to an IMS Connect system have affinity to the port on which IMS Connect received the original input message as follows:

► PORTAFF=Y

IMS Connect returns all CM0 output for this IMS Connect client through the same port on which it received the original input message.

► PORTAFF=N

IMS Connect attempts to return the CM0 output to the first client it finds on any available port with an outstanding request from this client ID.

Note: If running in a sysplex environment that has implemented redundancy, load balancing, super member support, and so forth, PORTAFF=N is a reasonable choice. We do not recommend using the same instance of a single client ID across multiple ports.

When PORTAFF=N, IMS Connect attempts to return the CM0 output to the first port found on which the client ID of this IMS connect is present. PORTAFF=N is the default. There is a consideration in this area because if a message is retrieved from the Hold Queue, then the PORTID in OMUSER_PORTID is ignored and IMS Connect assumes the port ID is the generic ICONNECT port. This assumption tends not to be a problem unless an error, such as a connection failure, occurs. When that happens and a message has been received in IMS Connect for delivery, IMS Connect scans all the ports under the generic ICONNECT port and delivers the message to the first one that it finds.

Figure 8-3 on page 224 shows a situation where two Resume_TPIPE AUTO requests specifying the same client ID, clientABC, are sent into a single IMS Connect. One request (from RMTA) is sent to IMSA and the other (from RMTB) is sent to IMSB. IMSA has two messages, MSG1 and MSG2, on the Hold Queue. IMSB has no messages at the moment and so clientABC on RMTB just waits. IMSA sends MSG1 to IMS Connect, which delivers the message to the outstanding request for clientABC on RMTA, which responds with an ACK. In this scenario, the connection fails for one of several reasons: ACK timeout, network problem, or another reason. Because the Resume_TPIPE had originally specified AUTO, IMSA sends MSG2 after the ACK for MSG1 is received. When IMS Connect receives the message, it detects that the connection to RMTA is no longer there and scans the ports under ICONNECT to find the first one available. IMS Connect sends MSG2 to the waiting clientABC on RMTB. This instance of ClientABC retrieves the message and sends an ACK back to IMSB, which is not expecting an ACK. Meanwhile, IMSA's Hold Queue for clientABC is in WAIT-H status waiting for an ACK that will never be received.

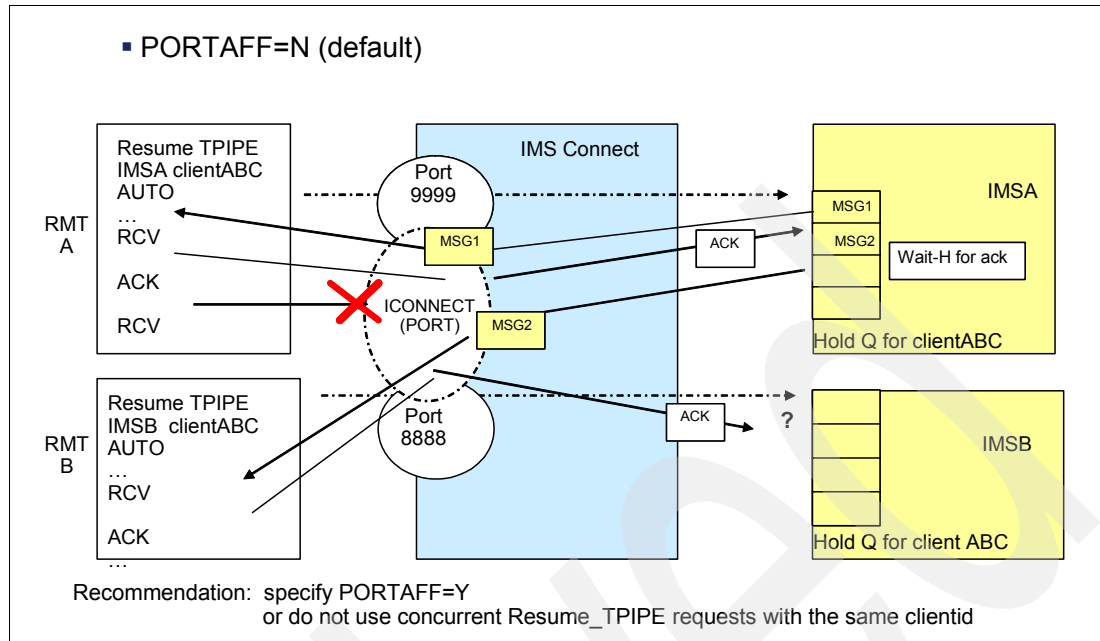
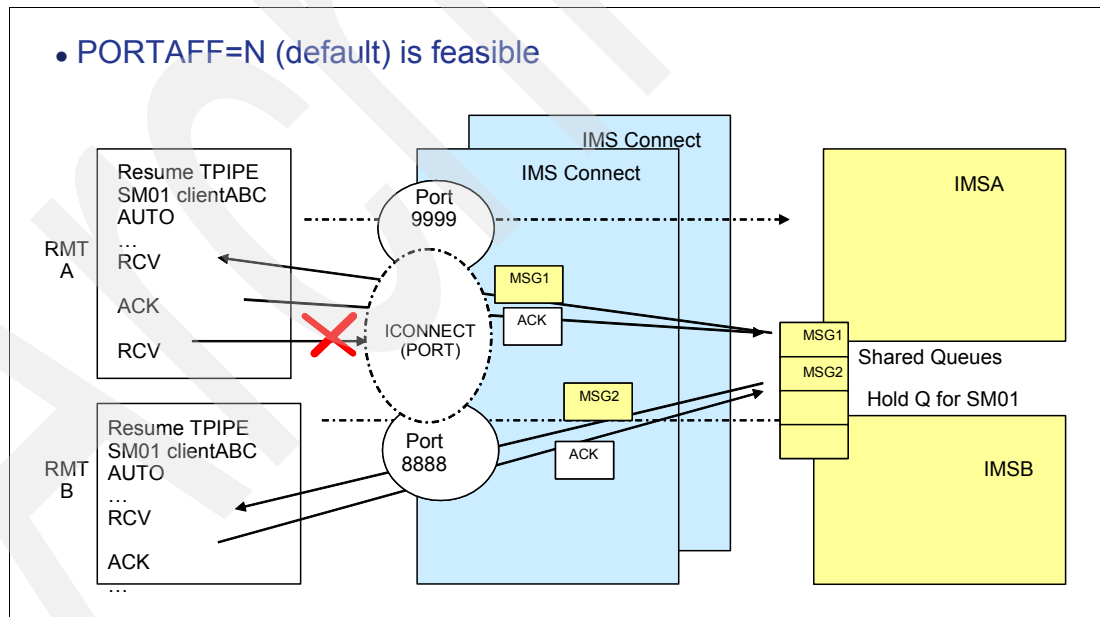


Figure 8-3 Consideration: concurrent requests using the same client ID

In an environment that supports full redundancy including Shared Queues, Super Member support, Sysplex Distributor, and so forth, specifying PORTAFF=N is feasible. This configuration with concurrent Resume_TPIPE clients using the same client ID presumes that any of the remote clients can retrieve any of the messages.



Alternate client ID support

IMS Connect introduces a new protocol that allows client applications to specify an alternate client ID in the RESUME TPIPE request. IMS Connect forwards the alternate client ID to OTMA, and OTMA returns the asynchronous messages that are queued to the TPIPE of the alternate client ID name to the client application that issued the Resume TPIPE request.

This Alternate Client ID function differs from the Reroute capability that was discussed earlier. Reroute requests address the situations when messages cannot be delivered or are not acknowledged (NAKed). In these situations, OTMA queues the message onto the TPIPE name associated with the reroute request when the undeliverable condition occurs. The Alternate Client ID function, however, supports Resume TPIPE requests that retrieve messages that are already queued to a TPIPE name but the name is different. This new capability can be used to provide a programmable solution in the OTMA/IMS Connect environment that is comparable to the way that IMS users can assign an LTERM and all messages queued to it to a different node.

This support applies to:

- ▶ Remote client applications that invoke user message exits HWSSMPL0 and HWSSMPL1
- ▶ IMS SOAP Gateway client applications using user message exit HWSSOAP1
- ▶ IMS TM Resource Adapter client applications using user message exit HWSJAVA0
- ▶ Any other user-written IMS Connect message exits

No support is provided for the local option, HWSIMSO0, and HWSIMSO1.

Note: IMS Version 9 was the last release to support the following user message exits: HWSIMSO0 and HWSIMSO1. Clients using these exits need to migrate to HWSSMPL1.

To take advantage of the new capability, optional fields have been added to both the IRM and OTMA headers. Either field, IRM_RT_ALTCID in the IRM or OMUSR_RT_ALTCID, can be used to set a valid alternate client ID value. The feature is not enabled if the fields are both set to blanks.

8.4.8 XML Adapter support

XML Adapter support is a capability that supports translation between XML messages and IMS messages. The IMS Connect client, for example, IMS SOAP Gateway, sends an XML message with a request for translation. For inbound messages, IMS Connect invokes the XML Adapter to translate the message for IMS, removes XML tags, and if necessary, converts from UNICODE to EBCDIC. For outbound, IMS Connect invokes the XML Adapter to prepare an XML message and creates the XML tags. Also, if necessary, it converts the message from EBCDIC to the appropriate UNICODE encoding schema.

Note: XML Adapter support is delivered with IMS Version 9 via the APARS PK24912 and PK29938.

XML Adapter support opens up IMS Connect to: receive and recognize messages containing XML tags; provide the conversion into a message format for IMS including unicode to EBCDIC translation if needed; receive IMS Application reply messages that do not contain XML tags; and perform the conversion to XML, as well as any EBCDIC to unicode translation prior to replying to the remote client.

Figure 8-4 on page 226 shows the current supported configuration for the XML Adapter support in IMS Connect. The IMS Soap Gateway is an IBM-provided function that must be at a minimum level of V9.2 to take advantage of the XML Adapter in IMS Connect. A Roll Your Own (RYO) client can also take advantage of the new functionality by adhering to the application protocol and sending the appropriate settings in the IRM header.

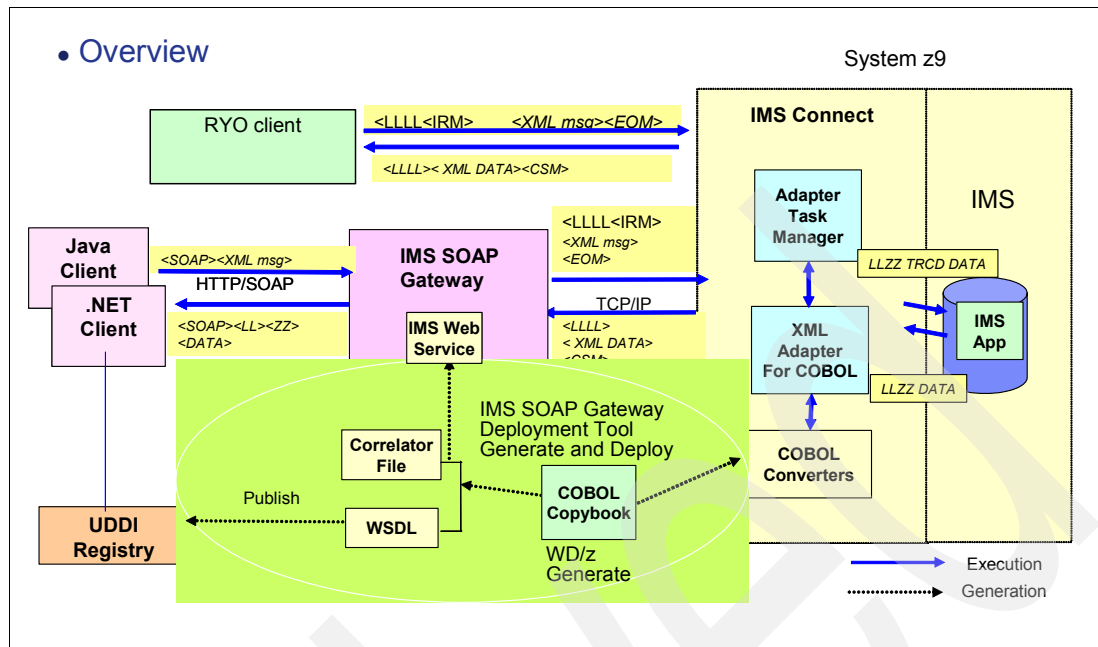


Figure 8-4 Supported XML Adapter configuration

WebSphere Developer for zSeries® (WDz) supports the generation of XML Converters for COBOL applications using COBOL copybooks. The XML Adapter support in IMS Connect, in conjunction with the generated WDz XML Converters, facilitates the conversion of XML transactional requests into byte stream application data structures and vice versa. Clients can send transactional requests in XML to IMS Connect, where the XML Adapter calls the user-specified WDz XML Converter to convert it to byte streams that the IMS application understands, and then IMS Connect sends it to IMS. Responses when received by IMS Connect will be converted to XML before being sent back to the clients. Each IMS application expects its messages to be in a certain data structure; therefore, one XML Converter is needed for each IMS application.

Input messages must adhere to the application protocol, as defined and documented for IMS Connect, by providing an IRM header that provides the name of the message exit routine that is to be called. For XML Adapter support, the IRM provides flags and values that can be used to invoke the appropriate functions.

The IRM header always begins with an LL (2-byte length field) followed by two bytes that are usually zeros for non-XML message processing. When the XML Adapter is to be invoked, the first zz byte can be used to signal the inclusion of 16 bytes in the user data section of the IRM header for specification of the TAG and ADAPTER names. The second byte details whether the tagged data includes the transaction code or not.

At the end of the user data section, two fields provide the information for the XML environment:

► **IRM_TAGNM**

Provides eight bytes to specify the TAG name. This is the name of the COBOL Driver, the XML metadata interchange (XMI), or whatever is required for the Adapter to call or use to perform the XML transformation.

► **IRM_ADAPTNM**

Provides eight bytes to specify the ADAPTER name. This is the name of the routine that is to gain control from IMS Connect to remove and add the XML tags, defined by the TAG

name. The same Adapter that processes the input message (removes the XML Tags) will be required to process the output message (add the XML Tags).

Reply messages are sent back by IMS Connect in tagged format. When the message is sent by IMS to IMS Connect, the user message exit invokes the Adapter to process the byte array data and convert it to XML tagged data before sending the message to the remote client.

XML Adapter support implementation

To enable the support in IMS Connect, several definitions have to be created:

- ▶ A new ADAPTER statement in the HWSCFGx file provides the option of identifying the presence or absence of the XML Adapter support. The parameter XML= provides two values. XML= Y/N. Yes requests the XML Adapter function. The default of No disables the function.

The only adapter that is currently supported is the IBM-provided COBOL Adapter HWSXMLA0.

- ▶ The definitions for the adapter are provided in an EXITDEF statement in a special PROCLIB member. The name of the member can be HWSEXIT0, as shown in this example, or it can be any name that is meaningful to the installation. The values of the EXITDEF statement are:
 - TYPE=XMLADAP must be coded as is and defines the exit as an XML Adapter plug-in to IMS Connect.
 - EXITS=HWSXMLA0 also must be coded as shown for the IBM-supplied COBOL Adapter.
 - ABLIM = 0 (unlimited), 1, up to 2 147 483 647 defines the abend limit, which is the number of times that the XML ADAPTER can abend before it is disabled.
 - COMP=HWS must also be coded as is and requests the HWS component of IMS Connect.
- ▶ The name that is chosen for the member containing the EXITDEF statement must also be defined to IMS in the BPE configuration file statement EXITMBR.

IMS Connect provides a new user message exit routine that specifically understands the XML interface. In addition to the existing message functions of READ, XMIT, INIT, and TERM, the new exit routine, HWSSOAP1, includes three new interfaces: RXML, EXML, and XXML. On input from the IMS Connect Client Application, IMS Connect calls HWSSOAP1 with the function type set to 'RXML', which translates the IRM header and returns the ADAPTER name and the TAG name back to IMS Connect. Upon returning to IMS Connect, error analysis will be performed. If an error exists, function 'EXML' is invoked to process the error message. If no error exists, IMS Connect calls and passes control to the ADAPTER passing a set of parameters that determine: if this ADAPTER should process the message; if translation is required or not; and what XML processing is required. When the reply is ready to be sent back, IMS Connect receives the message and invokes the 'XXML' function. Again, the ADAPTER is called to prepare the reply in a tagged format.

New fields in the EXPRM parameter list include:

- ▶ EXPRXML_TAGNM

Defines an output field and contains the TAG NAME to be used by the ADAPTER, the TAG name represents one of these names: XML name for user purposes, COBOL Driver name for IMS Connect support of the COBOL Adapter, or the user Map Name for user purposes

► **EXPRXML_ADPTNM**

Defines an output field and contains the Adapter name to be used by IMS Connect to load the correct Adapter.

► **EXPRXML_RETCODE**

Return code returned by the User Message Exit back to IMS Connect.

► **EXPRXML_RSNCODE**

Reason code returned by the User Message Exit.

To perform the correct conversion for each message, the Adapter requires information about how to do this. This information is called a COBOL XML Converter. COBOL XML Converters are COBOL programs generated from WebSphere Developer for zSeries (WDz) using the target IMS COBOL application's copybook. For each copybook, WDz generates three COBOL programs: an Inbound Converter (XML schema), an Outbound Converter (XML schema), and a Driver (COBOL code). These three programs are combined into one file and are referred to as a *COBOL XML Converter Driver*. A COBOL XML Converter Driver has to be created for each IMS COBOL transaction application that wants to support XML transaction messages. For details about generating COBOL XML Converter Drivers using WDz, see the WDz documentation.

On an inbound request to IMS, the XML Adapter calls the COBOL XML Converter Driver and passes it the inbound function code. The COBOL XML Converter Driver calls the Inbound Converter to parse the incoming XML message, convert the parsed message into the COBOL data structure byte streams, and then return it to the XML Adapter. On an outbound reply from the IMS application program, the XML Adapter calls the COBOL XML Converter Driver with an outbound function code, which then calls the Outbound Converter. The Outbound Converter converts the IMS output message into XML and passes it back to the XML Adapter, which then sends it to IMS Connect to be sent back to the client.

The COBOL XML Converter routines must be added into an Authorized Program Facility (APF)-authorized library that it concatenated into the STEPLIB list of the IMS Connect startup JCL.

XML Adapter support message flow

Figure 8-5 on page 229 gives a high-level view of the flow of messages. When an IMS Connect Client sends a transaction message in XML to IMS Connect, IMS Connect receives the input message and checks a flag in the IRM that indicates whether the input message is in XML.

If it is XML, IMS Connect calls the XML Adapter. The XML Adapter converts the XML input message into the COBOL Application input data structure and sends it back to IMS Connect. IMS Connect processes the returned stream and forwards it on as an `LLzz_trancode_data` message to IMS so that the IMS Application can process the input message. The IMS COBOL Application returns an output message to IMS Connect. IMS Connect again calls the Adapter. The XML Adapter converts the IMS COBOL Application output message into an XML output message and sends it back to IMS Connect. IMS Connect processes the message from the XML Adapter and sends the XML output message to its Client.

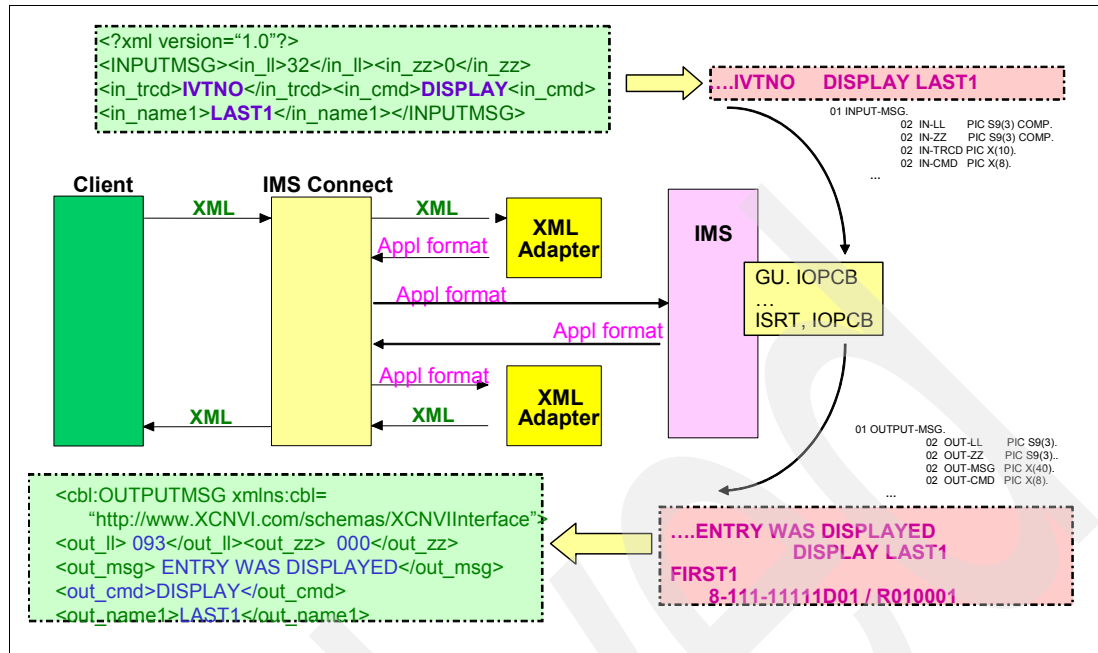


Figure 8-5 High-level view of the flow of messages

The IMS COBOL Application is not aware of any message conversions. It receives messages in the same application specific format that it normally expects. The diagram in Figure 8-5 shows the message formats received by each component.

Several IMS Connect error messages have been added to provide more information about possible problems and failures:

```

HWSA0300E INTERNAL ENQ OF AWE FAILED FOR ADAPTER adptname; R=rc, M=mc
HWSA0305E ADAPTER adptname DID NOT RETURN type BUFFER; M=mc
HWSA0310E FAILED TO RELEASE type BUFFER FOR ADAPTER adptname; R=rc, M=mc
HWSA0315E INVALID EXIT FUNCTION REQUESTED; FUNC=ft, msgid1/msgid2,M=mc
HWSA0320E USER MESSAGE EXIT RETURNED INVALID RETURN CODE; R=rc, M=mc
HWSA0325E ADAPTER adptname RETURNED INVALID RETURN CODE; R=rc, M=mc
HWSA0330E NO ADAPTER NAME RETURNED FROM USER MESSAGE EXIT; M=mc
HWSA0335E ADAPTER adptname IS IN POSSIBLE LOOP, REQUESTING OUTPUT BUFFER; M=mc
HWSA0340E type MESSAGE NOT PROCESSED BY ADAPTER adptname; R=rc, M=mc
HWSA0345E ADAPTER adptname ENCOUNTERED AN ERROR; R=rc, M=mc
HWSA0350E ADAPTER adptname RETURNED AN INVALID DATA LENGTH; LEN=len, M=mc
HWSA0355E ADAPTER adptname FAILED TO INITIALIZE; ADAPTER SUPPORT IS DISABLED;
RC=rc, M=mc
HWSA0360E ADAPTER PROCESSING REQUESTED, BUT ADAPTER SUPPORT IS DISABLED; M=mc
  
```

XML Adapter considerations

There are two types of runtime errors generated in the XML Adapter: the errors from WdZ COBOL XML Converter Drivers and the errors from the XML Adapter. Both types of error messages are returned from the XML Adapter in XML-tagged messages if the output code page is available. IMS Connect does not distinguish between the two types of errors. If the output code page is not available, the XML Adapter returns an XML Adapter Status Message (XASM). The XASM includes a return and reason code to indicate the type of error.

Besides these two types of errors, the XML Adapter also passes other error messages from and to IMS Connect, for example, the DFS error messages generated by IMS transactions.

For any DFS messages, either errors or non-errors, the XML Adapter wraps the messages with XML tags and passes them back to the caller. The DFS message in XML format is defined as `<IMSDFSMessage>DFS.....</IMSDFSMessage>`.

An IMS Connect restart is required to recognize any changes that have been made to the data set containing the COBOL XML Converter drivers. However, you can refresh the Adapter exit routine for bringing in a new version of the ADAPTER plug-in user exit routine while IMS Connect is active by using the Base Primitive Environment (BPE) refresh capability, for example:

```
F ICONN01, REF USRX NAME(XMLADAP)
```

Note that the COBOL XML Converter Driver requires a Language Environment to run. The XML Adapter brings up a Language Environment during its initialization processing. Additionally, The XML Adapter only supports single-segment IMS transaction applications.

8.4.9 IMS SOA Composite Business Application support

IMS SOA Composite Business Application support is a new capability for IMS TM Resource Adapter clients that invokes IMS conversational transactions. IMS TM Resource Adapter (formerly IC4J) applications running in a WebSphere Application Server (WAS) can take advantage of either dedicated or shareable persistent sockets. A dedicated persistent socket remains dedicated to a particular user-specified client ID for commit-then-send CM0 interactions until released by a client application's request. Shareable persistent sockets, however, can be shared (serially reused) by multiple applications in the WAS server running either send-then-commit CM1 or commit-then-send CM0 interactions. For the latter type of socket, IMS TM Resource Adapter generates a client ID and does not allow user-specified client IDs.

Although the IMS TM Resource Adapter has supported interactions with IMS conversational transactions, IMS Version 10 enhances this capability by allowing the iterations between a conversation to span multiple shareable sockets.

IMS SOA Composite Business Application support enhances the IMS interaction with the process choreography function of WebSphere Process Server. Prior to this enhancement, process choreography with IMS was limited to non-conversational transactions. With IMS Version 10, process choreography can now include conversational transaction support.

Figure 8-6 on page 231 discusses the flow and a potential issue in the existing implementation.

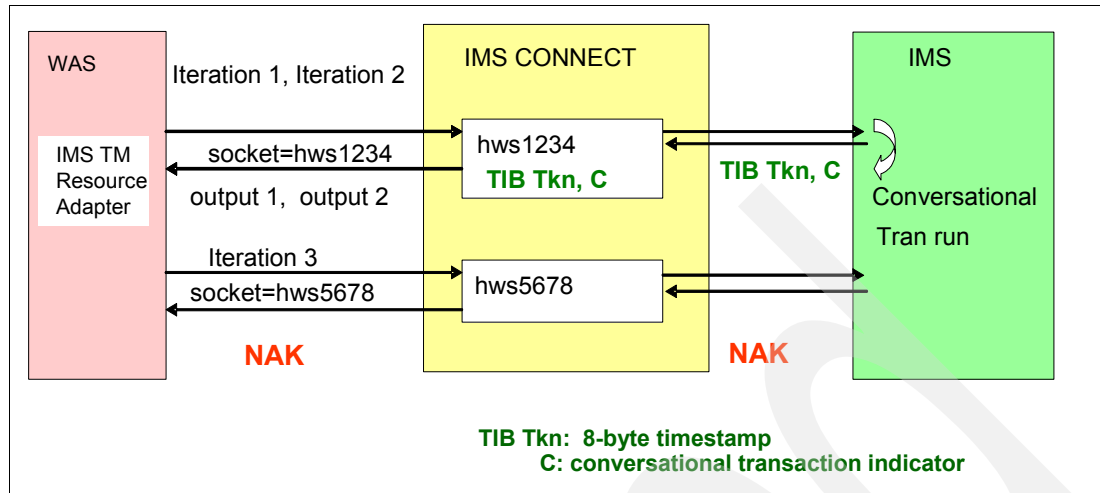


Figure 8-6 The situation in previous releases

The first request from the IMS TM Resource Adapter application invokes the first iteration of an IMS conversational transaction flow. The IMS TM Resource Adapter either uses an existing shareable persistent socket connection or generates a new one. For this example, the generated client ID is hws1234. At this point, neither the remote client nor IMS Connect is aware of whether the input is a conversational transaction. When the first reply message for this conversational iteration is created, IMS includes a conversational indicator (C) and a Transaction Instance Block token (TIB Tkn) in the message prefix that is sent back to IMS Connect. IMS Connect then saves these two pieces of conversational information in an internal control block associated with the client ID, hws1234. IMS Connect also sends the C indicator and the TIB Tkn with the output of the first iteration to the IMS TM Resource Adapter, which returns them to the Java application.

When the second iteration of the same conversation begins, the message is routed to one of the shareable sockets in the pool. If the second iteration comes in through the same socket, hws1234, IMS Connect uses the saved information in its internal control block to associate this conversation iteration with the previous one.

If the third iteration is routed to a different socket in the WAS pool, for example, HWS5678, IMS Connect cannot associate the request with the intended internal control block (hws1234) and the conversation information saved in it. IMS Connect, therefore, creates a different internal control block for hws5678, which has no knowledge of the existing conversation. When IMS receives the input, it rejects the request with a NAK.

IMS Connect defines a new conversational option, which is available to IMS TM Resource Adapter applications. Setting the OMUSR_CONV_OPTION to true (X'80') invokes the function. If the function is not enabled, IMS Connect uses the existing conversational logic. Note that setting the OMSUSR_CONV_OPTION to false (X'00') does not imply a non-conversational transaction.

IMS Connect creates the conversation token (the TIB token, Conversation indicator, and conversation option flag), cleans up the internal control blocks, and passes the conversation token to the IMS TM Resource Adapter. The Adapter passes the information to the application, which has to maintain the information to pass back on the next iteration of the conversation. This architecture allows the conversation to be continued by IMS Connect and passed to IMS on any of the sockets that are available to the remote client.

Figure 8-7 illustrates the new capability. When receiving a transaction reply from OTMA, IMS Connect checks to see whether the reply message contains the conversational transaction indicator, the Transaction Instance Block (TIB) token, and the new conversational option flag. If so, IMS Connect passes the conversation token back to the IMS TM Resource Adapter and cleans up all the internal knowledge about this conversational iteration. The Adapter then returns the conversation token back to the Java client. The Java client maintains responsibility for sending the conversation token back in the subsequent iterations of a conversation or in the SYNC_END_CONVERSATION interaction to end a conversation. The conversation token is composed of the 8-byte timestamp of the TIB token, the conversational transaction indicator, and the new conversational option indicator. The conversation token is passed back and forth between the Java client, the IMS TM Resource Adapter, IMS Connect, and OTMA. IMS Connect uses the conversation token to track the conversation within a single iteration.

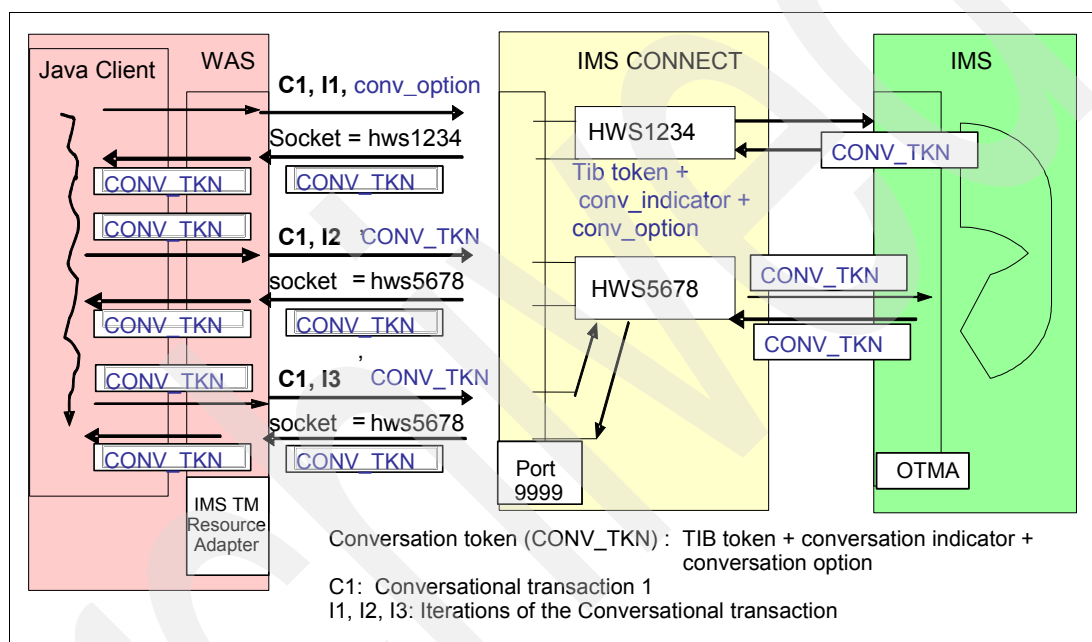


Figure 8-7 The new capability

In a complex environment, such as one that uses the z/OS Sysplex Distributor or distributed WebSphere Application Servers (WAS) in a cloned environment, the request from a Java client can be distributed through multiple WAS instances or multiple IMS Connects to a single IMS. Figure 8-8 on page 233 shows one possible configuration.

The current architecture of IMS Connect's conversational transaction support uses the TIB token to keep track of multiple iterations for a given conversational transaction. Because this unique token is not shared between multiple IMS systems, even with Shared Queues, the function does not support an environment of multiple IMSs.

The function does not support user-written IMS Connect applications. The new conversation token is not passed in the IRM header.

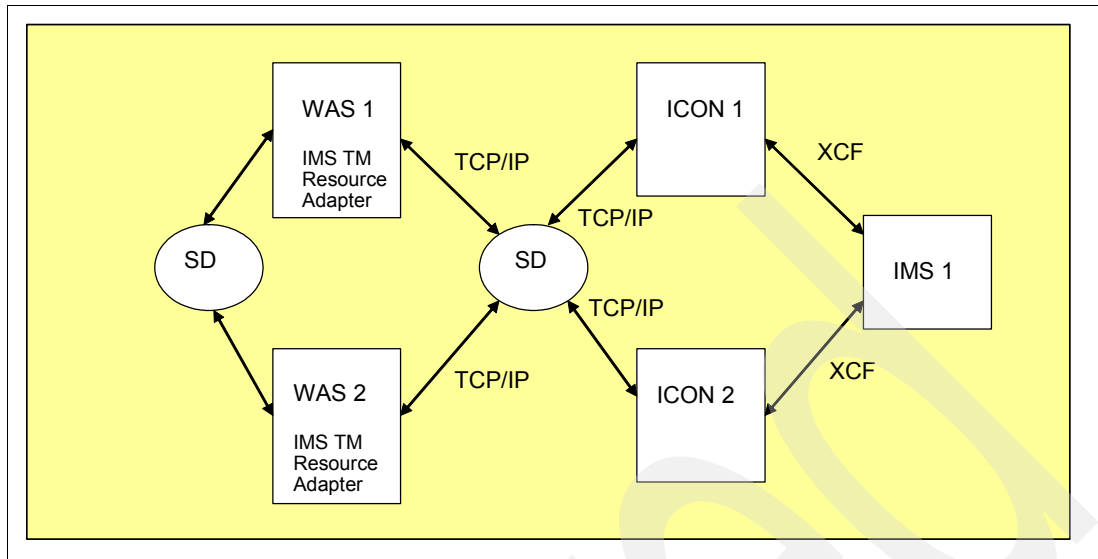


Figure 8-8 A configuration with cloned WAS and SDs accessing a single IMS

Figure 8-9 shows an example of an environment that has cloned distributed WebSphere Application Servers and multiple instances of IMS Connect accessing a single IMS system.

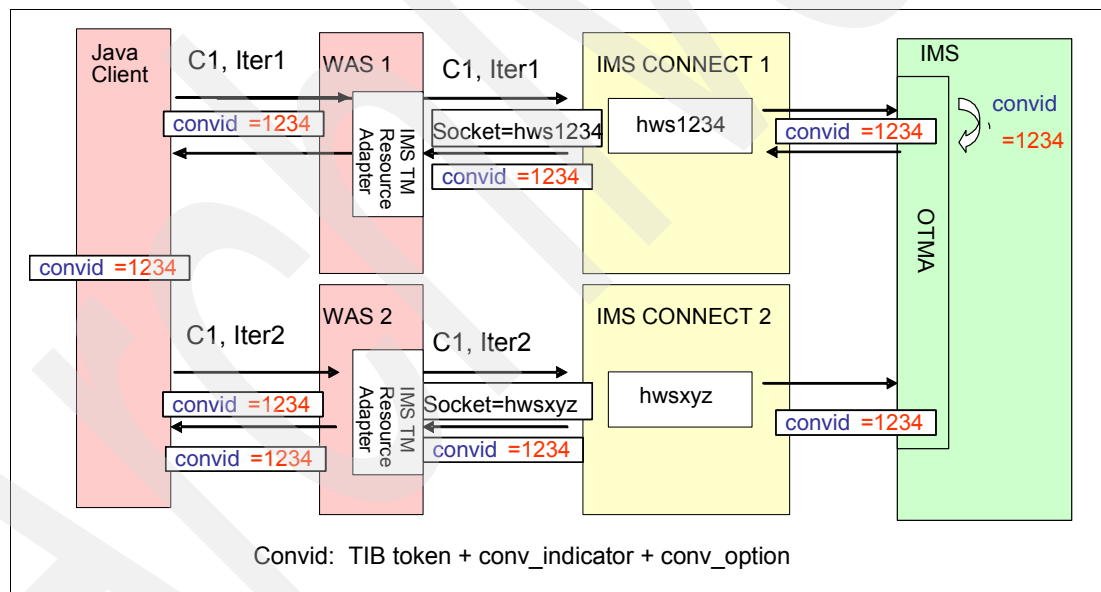


Figure 8-9 A message flow with the new conv option

If the new conversational option flag and other conversational information are set in an input message, IMS Connect assumes that this is a subsequent iteration of a conversational transaction and uses the conversational information to resume the conversation. If IMS Connect determines that the conversational transaction information from the input message does not conflict with any conversation it manages, IMS Connect continues processing the input. Otherwise, IMS Connect issues the HWSP1510E error message:

HWSP1510E MESSAGE CONTAINS INVALID CONVERSATION INFORMATION C=clientid, M=mc

This scenario can happen when the conversational option flag is changed from FALSE in one conversational iteration to TRUE in a subsequent one.

The IMS Connect commands, VIEWHWS and QUERY MEMBER TYPE(IMSCON), display the CONV status if a client is running an IMS conversational transaction. When the new conversational option is specified, IMS Connect no longer keeps track of the conversational status of the client across iterations of the conversational transaction. The output of the commands, therefore, will not be able to display the CONV status. Only when the display command is entered during the instance that a conversational iteration is in progress will the CONV status be displayed.

8.5 IMS Connect enhancements benefits

This section summarizes the benefits of IMS Connect enhancements in IMS Version 10:

- ▶ Better control and management of IMS Connect environment
HWSCFGxx parameters for ACEE aging value, CM1 Timeout value, and Message Flood Control
- ▶ Improved security capability
Ability to change passwords and support mixed case
- ▶ Integrated XML support
XML Adapter opens the IMS environment to XML applications
- ▶ IMS SOA Composite Business Application support
Expands conversational processing to environments with connection pooling, sysplex distributor, and cloned application servers

Integration enhancements

The IMS Integration Suite is a collection of IMS middleware functions and tools that support your IMS on demand systems and your distributed IMS application environment. Some of these middleware functions and tools are generally available, whereas others might be part of an open beta program or technology preview program. All of these functions and tools require IMS Version 9 (or later) and are not supported for previous releases of IMS.

The IMS Integration Suite includes the following IMS middleware functions and tools:

- ▶ IMS TM Resource Adapter (formerly know as IMS Connector for Java)
- ▶ IMS DB Resource Adapter (formerly known as IMS JDBC Connector)
- ▶ IMS DLIModel Utility
- ▶ IMS SOAP Gateway
- ▶ IMS XML DB and XQUERY
- ▶ IMS Message Format Services (MFS) Web Support

In this chapter, we discuss each of these functions with the exception of IMS MFS Web Services.

IMS MFS Web Support provides two solutions to reuse existing MFS-based IMS business logic:

- ▶ IMS MFS Web Services support allows you to submit MFS-based IMS transactions as Web services. IMS MFS SOA support replaces IMS MFS Web services with a broadened support for development tooling supporting the latest J2EE™ Connector (J2C) programming model and the Service Component Architecture (SCA) programming model.
- ▶ IMS MFS Web Enablement support replaces the MFS online processor and allows you to seamlessly navigate MFS-based Web pages that all have a similar look and feel.

Because IMS MFS Web Support solutions are not directly related to IMS Version 10, but have their own evolution path, we do not discuss them in this book. You can find the latest information about these solutions via the IMS home page:

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

9.1 IMS TM Resource Adapter

IMS TM Resource Adapter is a J2EE Connector Architecture (J2C) resource adapter that supports IBM Java integrated development environments (IDEs) to create Java applications that can access IMS transactions. The IDEs import C, COBOL, and PL/I source to generate the Java data bindings for the IMS transaction input/output message segments.

Figure 9-1 shows the dual functions of IMS TM Resource Adapter to provide IDE support and to provide runtime environment support to access IMS applications using IMS Connect as the interface to IMS via OTMA.

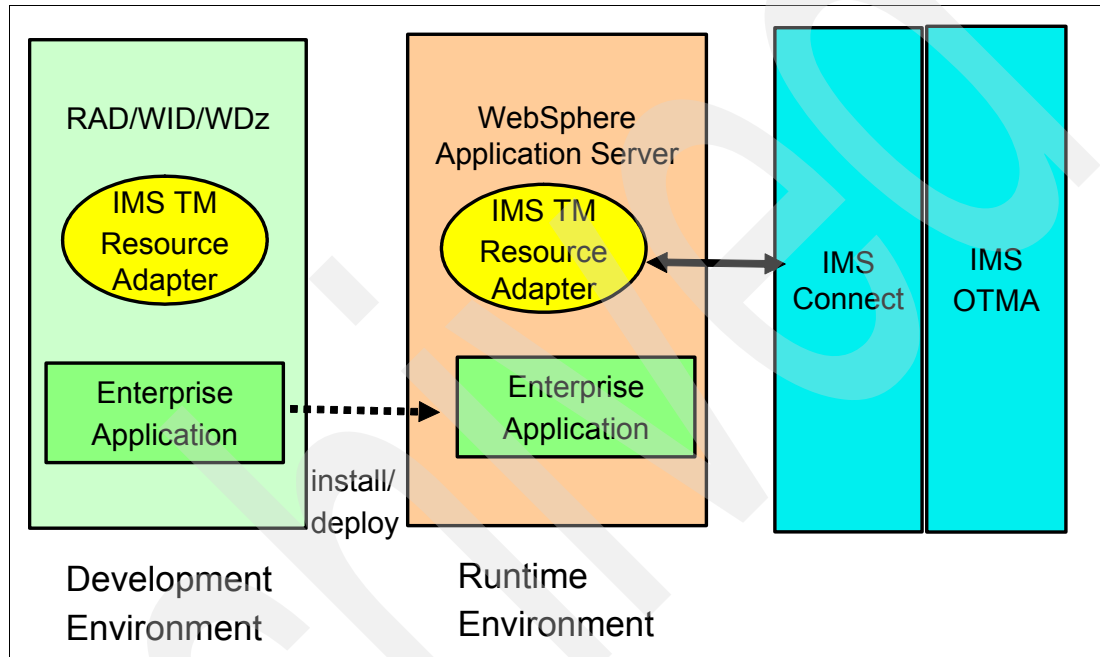


Figure 9-1 IMS TM Resource Adapter overview

The following IDEs are supported by IMS TM Resource Adapter:

- ▶ Rational Application Developer (RAD)
- ▶ Rational Software Architect
- ▶ WebSphere Integration Developer (WID)
- ▶ WebSphere Developer for zSeries (WDz)
- ▶ WebSphere Studio Application Developer Integration Edition (WSAD-IE)

The IMS TM Resource Adapter also provides a plug-in for a J2EE application server by using the standardized system contracts. IMS TM Resource Adapter provides runtime support for WebSphere Application Server on distributed and z/OS platforms.

A J2EE Application Server provides the runtime environment for application code. WebSphere Process Server (WPS) extends an application server by providing a business process engine that executes business processes. WID is the development tool that is used to create standards-based business processes that are deployed to WPS.

9.1.1 Pre-IMS Version 10 enhancements

IMS TM Resource Adapter has been enhanced independently between IMS Version 9 and IMS Version 10. Let us have a quick look at these changes first. This is the list of

enhancements that IMS TM Resource Adapter provided after IMS Version 9 and before IMS Version 10:

- ▶ PL/I source message structures

Rational Application Developer imports C, COBOL, or PL/I definitions of the IMS transaction input and output messages and creates Java data bindings for the input and output messages. Prior to this support, the client had to create a COBOL copybook version of the IMS PL/I input/output message structure.

- ▶ Commit Mode 1, Sync Level CONFIRM

Commit Mode One Synclevel None has the potential to discard an output message without a ROLLBack of the database in the event that IMS Connect is not able to deliver the output message to IMS TM Resource Adapter. As a result, a compensation transaction might have to be executed to reverse the database changes. Adding the Commit Mode One Synclevel Confirm can reduce the necessity of using a compensation transaction, because an ACK will be performed by IMS TM Resource Adapter.

APPC provides CM1 sync levels NONE,CONFIRM, and SYNCPT. Clients, who have APPC client applications accessing IMS using CM1 sync level CONFIRM, can now provide IMS TM Resource Adapter clients with the same sync levels.

- ▶ Retrieve output using alternate client ID on shareable persistent sockets

IMS Connect APAR PK24907 supports a new protocol “Resume tpipe with alternate client ID” that provides the IMS TM Resource Adapter client application the ability to specify an alternate clientID in the SYNC_RECEIVE_ASYNCPUT interaction.

If IMS Connect cannot send the output message to IMS TM Resource Adapter, IMS Connect will respond NACK to OTMA. OTMA will not dequeue the output message.

- ▶ Enhanced SSL cipher configurability

The Secure Socket Layer (SSL) protocol ensures that the transfer of sensitive information over the Internet is secure.

Currently, IMS TM Resource Adapter allows clients to specify only “strong” or “weak” as the SSLEncryptionType for all connections created by an IMS connection factory. Enhanced SSL cipher configurability allows clients to specify a new value, “ENULL”, for SSLEncryptionType. When “ENULL” is specified, IMS TM Resource Adapter will attempt to use a cipher spec whose name contains the string “NULL.”

Null encryption will allow authentication to take place during the SSL handshaking process, which is currently the case. After the handshaking process for a socket has completed, including authentication as required, all messages will flow in the clear over that socket.

An example is the passing of non-confidential data between IMS TM Resource Adapter and IMS Connect, both of which are located behind a firewall.

9.1.2 Enhancements for IMS Version 10

The following is a list of IMS TM Resource Adapter enhancements provided with IMS Version 10:

- ▶ Asynchronous Callout to EJB™ and message-driven beans (MDBs)
- ▶ Change RACF password using IMS TM Resource Adapter and IMS Connect
- ▶ IMS TM Resource Adapter Enterprise Workload Manager (EWLM) policy-based performance management
- ▶ IMS service-oriented architecture (SOA) Composite Business Application support
- ▶ CM0 multiple PURG calls

Asynchronous callout to EJB and MDB

IMS TM Resource Adapter has been enhanced to support access to Enterprise Java Beans (EJBs) from an IMS application program. Refer to Figure 9-2 for the processing flow overview.

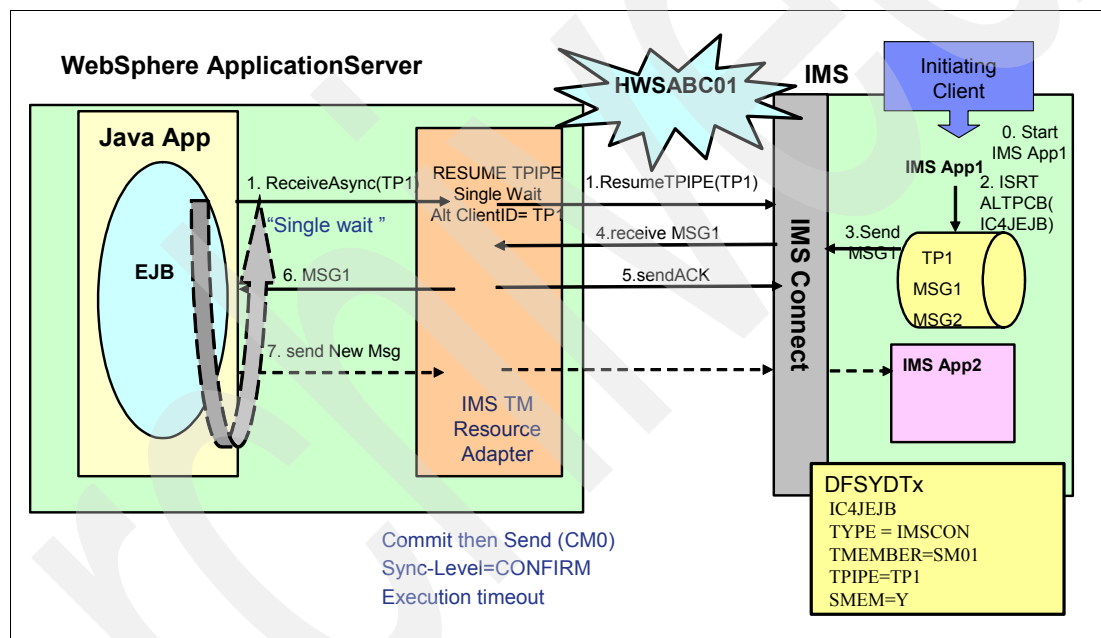


Figure 9-2 Asynchronous callout to EJB

In this scenario, an EJB uses the SYNC_RECEIVE_ASYNC_OUTPUT_SINGLE_WAIT (1) alt-clientID=TP1 to request a message queued to TP1. IMS TM Resource Adapter uses the IMS Connect “Resume tpipe with alternate clientID” protocol to wait for a message to retrieve (step 1). IMS Connect receives the request and notifies OTMA to send a message. An already scheduled (step 0) IMS Application program does an insert to an alternate program communication block (ALTPCB) for a destination defined in an OTMA destination routing descriptor (step 2). The OTMA destination routing descriptor is used to define the OTMA TPIPE for the enqueue of the message (TP1). OTMA sends the message to IMS Connect (step 3) and IMS Connect will return the message to IMS TM Resource Adapter (step 4). IMS TM Resource Adapter receives the message and notifies IMS Connect with an ACK (step 5). IMS Connect will forward the ACK to OTMA and the message will be dequeued.

IMS TM Resource Adapter receives the message and returns it to the EJB (step 6).

An optional step is for the EJB to create a response to the message request. IMS TM Resource Adapter sends the response to IMS (step 7). This results in the scheduling of an IMS application program to process the response.

IMS TM Resource Adapter also supports access to MDBs from an IMS application program. The J2EE Connector Architecture specifications V1.5 describes how an enterprise application can access a J2EE application environment. This is known as *Inbound processing*. IMS TM Resource Adapter provides inbound processing for IMS application programs. From an IMS application perspective, this is Asynchronous Callout processing. Refer to Figure 9-3 for a callout scenario to a MDB.

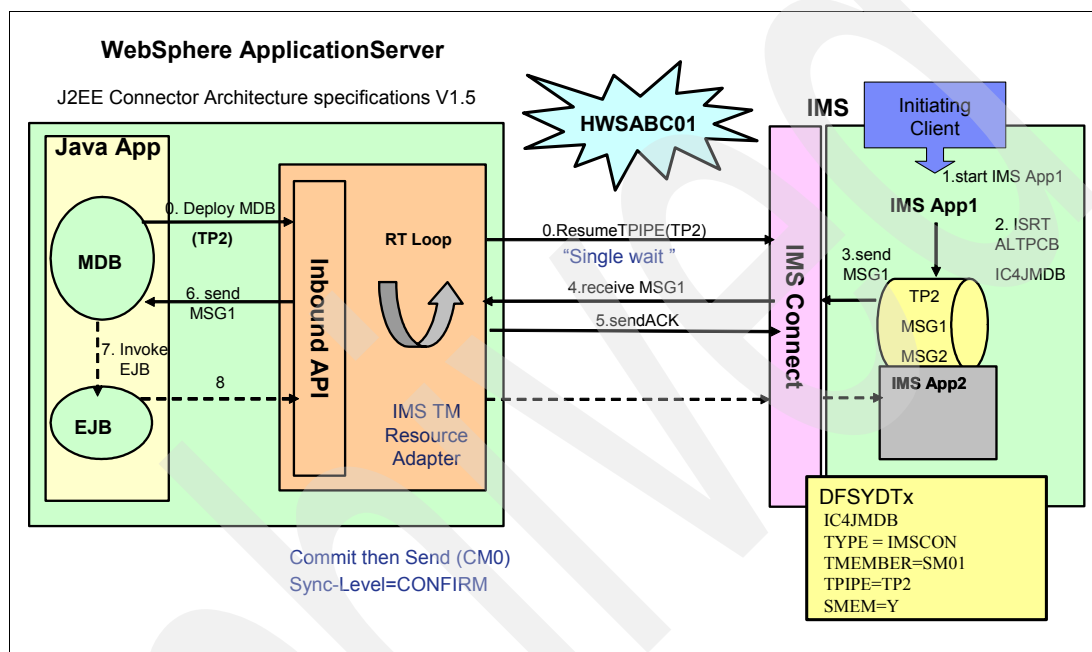


Figure 9-3 Asynchronous callout to MDB

In this scenario, IMS TM Resource Adapter uses the IMS Connect “Resume tpipe with alternate clientID” protocol to wait for a message to retrieve (step 0).

An already scheduled (step 1) IMS Application program does an insert to an ALTPCB for a destination defined in an OTMA Descriptor (step 2). The OTMA Descriptor is used to define the OTMA TPIPE for enqueue of the message.

OTMA sends the message to IMS Connect (step 3) and IMS Connect will return the message to IMS TM Resource Adapter (step 4). IMS TM Resource Adapter receives the message and notifies IMS Connect with an ACK (step 5). IMS Connect will forward the ACK to OTMA and the message will be dequeued.

A MDB running inside WebSphere Application Server will use the standard JCA 1.5 interfaces to listen to a callout request from IMS applications. IMS TM Resource Adapter receives the message and sends it to the MDB (step 6).

The MDB will invoke the EJB (step 7) to process the message.

An optional step is for the EJB to create a response to the message request. IMS TM Resource Adapter sends the response to IMS (step 8). This results in the scheduling of an IMS Application program to process the response.

RACF password change

IMS TM Resource Adapter has been enhanced to change a RACF password during interactions with IMS Connect.

When a J2EE application requires access to an IMS transaction that is password-protected, the RACF password might have expired. Using this facility, the J2EE application can prompt the client for an new password and provide the new password to facilitate completion of the business process. The RACF password change is implemented by IMS Version 10 IMS Connect change password request message, which has the following format:

LLLL | IRM | OTMA | LLZZ HWSPWCH *old-password/new-password1/new-password2* | EOM

Example 9-1 provides sample code for the RACF password change.

Example 9-1 Sample Java code for changing the RACF password

```
package sample.ims.hwspwcdh;
import com.ibm.connector2.ims.ico.*;

public class HWSPWCHIMSConnect
{
    public static void main(String[] args)
    {
        try
        {
            INMSG input = new INMSG();
            input.setInll((short) 31);
            input.setInzz((short) 0);
            input.setIncmd("HWSPWCH CURPSWD/NEWPSWD/NEWPSWD");
            HWSPWCHIMSProxy proxy = new HWSPWCHCmdIMSProxy();
            proxy.setINMSG(input);
            // Run the IMS service.
            proxy.execute();
        }
    }
}
```

IMS TM Resource Adapter EWLM performance management

The Enterprise Workload Management (EWLM) support requires the EWLM CICS/IMS Support that is built into z/OS Version 1 Release 8 or provided in APAR OA12935 for z/OS Version 1 Release 7.

IMS TM Resource Adapter obtains the EWLM correlator from WebSphere Application Server and then passes the correlator to IMS by inserting the correlator into the OTMA prefix for the transaction. The general format of the message is:

LLLL | IRM | OTMA *User Data (EWLM 64 byte correlator)* | LLZZ *TRANCODE data* | EOM

When the transaction returns to IMS TM Resource Adapter after processing is complete, IMS TM Resource Adapter reports the status of the transaction to WAS.

If the incoming transaction does not contain an EWLM correlator, the existing function is unchanged.

CM0 Send Only flows will not be instrumented by EWLM. IMS TM Resource Adapter does not receive a correlator for CM0 send only transactions. Therefore, the transaction will be sent to IMS with the correlator field zeroed out, and IMS will call the existing non-EWLM version of the WLM services.

If the correlator is provided, IMS TM Resource Adapter will pass it to IMS. If the correlator is not provided, the OTMA prefix field will be set to zero.

The implementation of this capability requires IMS Version 10 OTMA. If an EWLM-enabled transaction is routed from a IMS Version 10 to a back-level IMS, the transaction passes through the back-level system with the EWLM data unchanged and the existing WLM function is unchanged on the back-level system. No APAR is required for prior versions of IMS.

IMS SOA Composite Business Application support

IMS SOA Composite Business Application support provides multi-connection IMS conversational transaction support. It supports WebSphere Process Server business process flows, and it is only available for IMS TM Resource Adapter clients. Figure 9-4 shows a sample process flow.

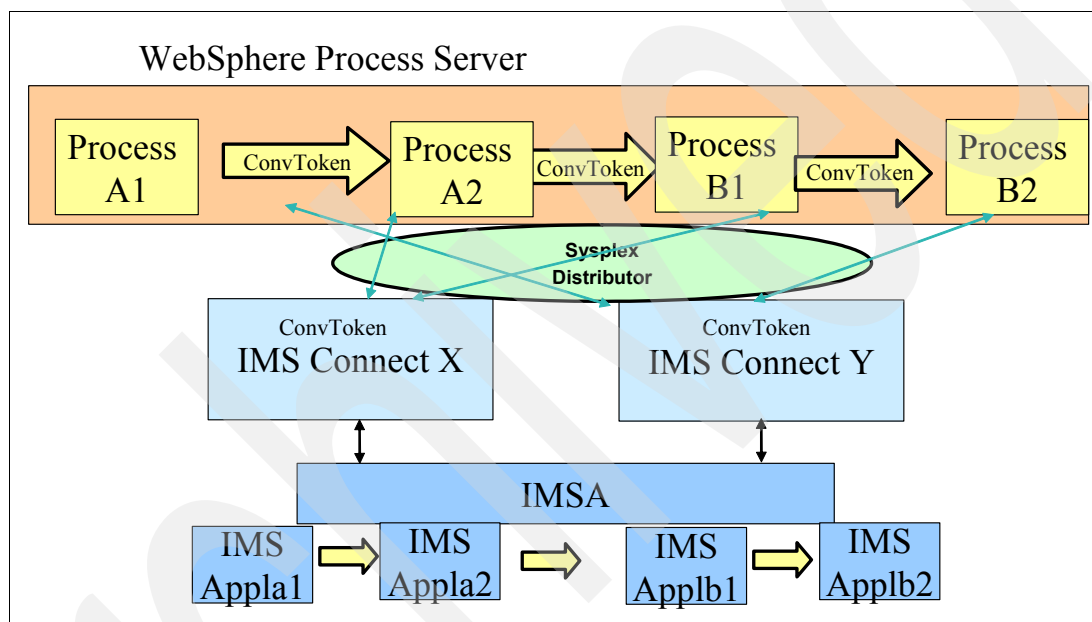


Figure 9-4 An example of a business process flow accessing IMS applications

In this scenario, Process A1 starts a conversational transaction for program App1a1 running in IMSA. Program App1a1 processes the request, updates the scratch pad area (SPA), and responds to Process A1. Process A1 passes the ConvToken to Process A2, which continues the IMS conversation and invokes program App1a1.

Program App1a1 immediately does a program to program switch to App2a2 running in IMSA. App2a2 responds to Process A2 and does a deferred program to program switch to Appl1b1 running in IMSA.

Process A2 passes the ConvToken to ProcessB1. ProcessB1 continues the IMS conversation and invokes Appl1b1 running on IMSA. PGM Appl1b1 processes the request, updates the SPA, and responds to Process B1. ProcessB1 passes the ConvToken to ProcessB2.

ProcessB2 invokes Appl1b1. Appl1b1 does an immediate program to program switch to Appl2b2. Appl2b2 responds to ProcessB2 and ends the conversation.

Because an IMS Conversational transaction is a commit mode 1 type of transaction, IMS Connect only supports commit mode 1 conversational input messages with a synchronization level of none or confirm from IMS TM Resource Adapter client applications.

CM0 multiple PURG calls

When there are multiple ISRT and PURG calls in the IMS application, OTMA sends one response message with multiple output segments if the client application uses Commit Mode 1 interaction. If the client application uses Commit Mode 0, OTMA will send multiple output response messages, one for each PURG call. Figure 9-5 shows this support prior to the IMS Version 10 enhancement.

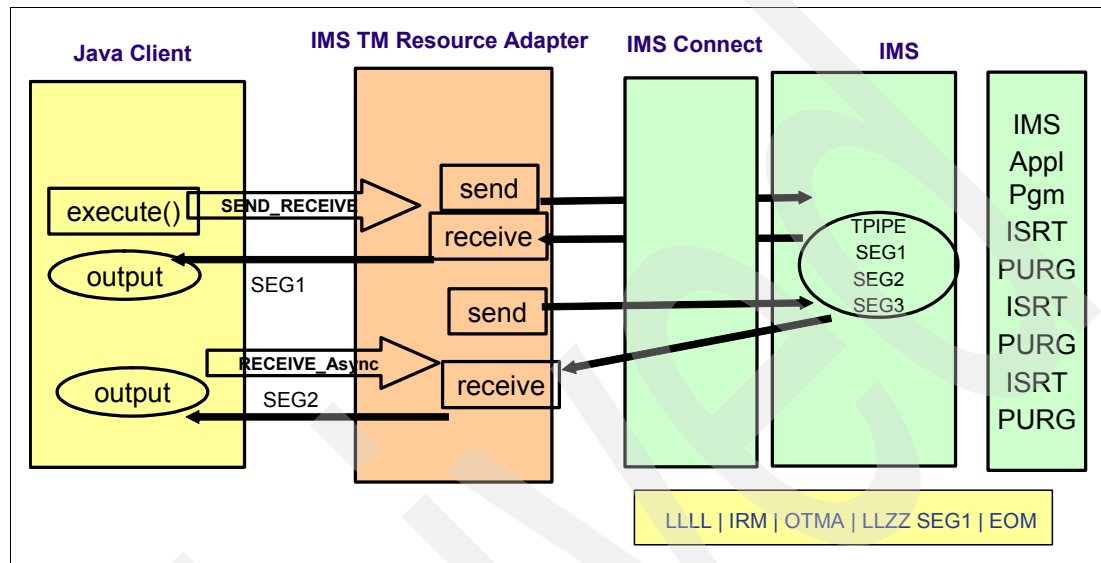


Figure 9-5 Multiple PURG call support prior to IMS Version 10

OTMA introduces an enhancement so that the client using CM0 with multiple PURG calls in the TP PCB can ignore the PURG calls to generate one output message with multiple segments. This provides consistent output message processing for CM0 and CM1.

A new property called `ignorePURGCall` will be added to the `IMSInteractionSpec`.

Commit Mode 1 is only supported using a shareable persistent socket.

IMS TM Resource Adapter uses persistent socket connections to communicate with IMS Connect. A persistent socket connection prevents the overhead of opening and closing the socket for each use by an application. IMS TM Resource Adapter provides two types of persistent socket connections:

- **Dedicated persistent**
A *dedicated persistent socket connection* is restricted to Commit Mode 0 (Commit-then-send protocol) and the client application provides the client ID to be used as the socket identifier.
- **Shareable persistent**
A *shareable persistent socket connection* supports Commit Mode 0 and Commit Mode 1 (Send-then-commit protocol). However, the client application does not provide a client ID for a socket identifier. IMS TM Resource Adapter generates a client ID.

The support for CONFIRM provides the ability for IMS TM Resource Adapter to respond with an ACK to the output message, which activates the IMS sync point process. In this case, the client application is not running under a Global Transaction. This provides the ability for IMS TM Resource Adapter to notify IMS that it has received the IMS application output message without requiring a Global Transaction.

Figure 9-6 presents the new optional support. In this example, the Java client is using a shareable persistent socket Commit Mode 1 Sync Level CONFIRM. IMS TM Resource Adapter generates HWSABC01 for a socket ID and sends the input message to the IMS application program via IMS Connect and OTMA. To provide a message queue anchor for the Commit Mode 1 output message, OTMA creates a transaction pipe (TPIPE) and will use the TCP/IP-generated Client Port number provided by IMS Connect for the TPIPE name. When IMS TM Resource Adapter receives the output message from IMS Connect, it responds with an ACK. IMS Connect will send the ACK to OTMA to activate the IMS sync point process.

Note the Java Client application cannot respond with an ACK.

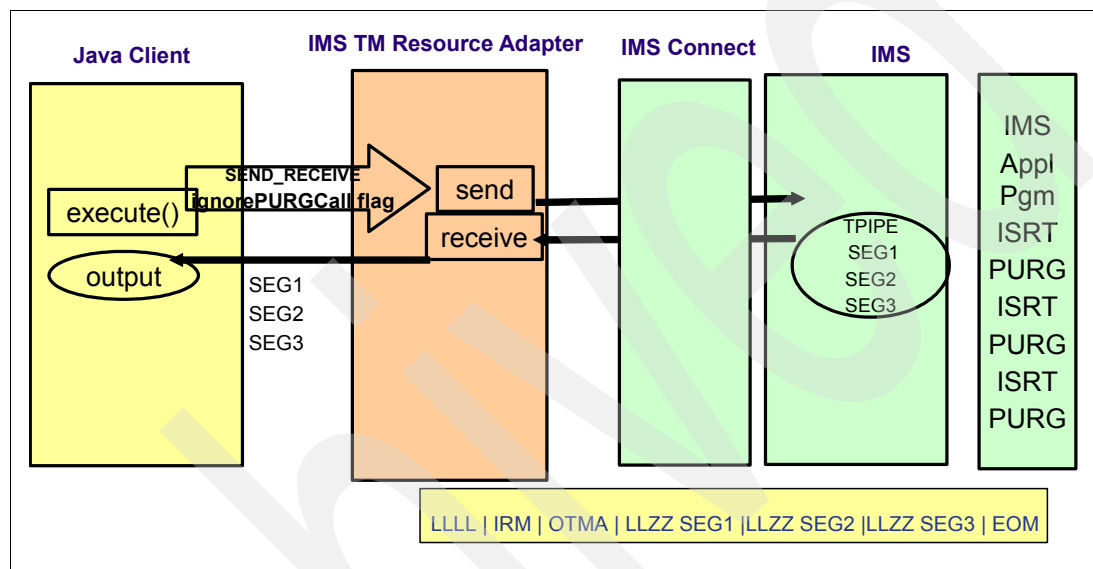


Figure 9-6 The new optional support

When there are multiple ISRT and PURG calls in the IMS application, OTMA sends one response message with multiple output segments if the client application uses Commit Mode 1 interaction. If the client application uses Commit Mode 0, OTMA will send multiple output response messages, one for each PURG call. OTMA introduces an enhancement so that the client using CM0 with multiple PURG calls in the TP PCB can ignore the PURG calls to generate one output message with multiple segments. This provides consistent output message processing for CM0 and CM1.

A new property called ignorePURGCall will be added to the IMSInteractionSpec.

9.2 IMS DB Resource Adapter (IMS JDBC connector)

The *IMS DB Resource Adapter* is a function of IMS that enables a programmer with minimal IMS knowledge to write Java application programs that access IMS databases. The IMS DB Resource Adapter enables JDBC access to IMS DB from IMS TM JMP/JBP environments, CICS Java application, DB2 Java Stored procedure, and Enterprise Java Beans running on WebSphere distributed and z/OS environments.

The IMS DB Resource Adapter has been enhanced to support DB2 Java Stored Procedures returning DB2 result sets instead of returning IMS result sets. The generalized sequential access method (GSAM) has been enhanced to use IMS Java Metadata generated by the

DLIModel Utility. Changes were also made to support IBM Software Development Kit (SDK) V5 for z/OS.

9.2.1 DB2 Java Stored Procedures returning DB2 result sets

The IMS DB Resource Adapter can be utilized in DB2 stored procedures on z/OS to access IMS database information. However, the current version only supports the specification of individual fields containing IMS data to be returned to the stored procedure client. Therefore, when the stored procedure is developed, it must know the IMS data type and amount of IMS data that will be returned from a stored procedure to the client. Any changes in the database or query can require changes to the stored procedure.

For IMS Version 10, the IMS DB Resource Adapter provides the capability to return IMS data from a DB2 Java stored procedure encapsulated in a DB2 result set. This capability removes IMS individual field processing and client dependency on IMS database structure.

Example 9-2 is an example of a client application calling a DB2 stored procedure that returns a DB2 result set. This shows how the client application can process the individual fields of IMS data from the DB2 result set.

Example 9-2 DB2 Java Stored Procedure client API using DB2 result set

```
String sql = "CALL SYSADM.AUTOLISTMODELS(?,?)";

// register the output parameters
callStmt.registerOutParameter(2, Types.CHAR);

// execute the statement and print results

ResultSet results = callStmt.getResultSet();

System.out.println("DealerName: " + dealername);
System.out.println("DealerPhone:      " + dealerphone);
System.out.println("OrderLastName:      " + orderlastname);
```

By embedding the IMS data within a DB2 result set, the type of data and the amount of data do not need to be known when the stored procedure is developed.

The DB2 global temporary table (GTT) is either created or declared to be used in the process of converting an IMS result set to a DB2 result set.

The IMS DB Resource Adapter `ResultSetConverter` class is used to convert an IMS result set instance into a DB2 result set instance. This method takes in an instance of `com.ibm.ims.db.DLIResultSet` object, which is IMS DB Resource Adapter's implementation of the `java.sql.ResultSet` interface, and converts it into an instance of the DB2 implementation of the `java.sql.ResultSet` interface.

The IMS DB Resource Adapter has been enhanced to create a DB2 GTT in the DB2 Stored Procedure Address Space and moves the IMS data returned from the IMS `ResultSet` into the DB2 GTT. The DB2 Result Set returned to the client application will iterate the DB2 GTT for result set processing.

The stored procedure clients are `READ_ONLY`. This means that the client cannot update or delete IMS data using the DB2 result set returned to it by the stored procedure, because it is not possible to send a change in a DB2 temp table back to an IMS database.

Existing DB2 stored procedures using the IMS Version 10 DB Resource Adapter can continue to run without modifications.

To update existing stored procedures to return DB2 result sets instead of individual fields, you need to update the stored procedure to specify the use of a created or declared type of DB2 GTT. After the Stored Procedure application queries the IMS database, the IMS DB Resource Adapter API is used to convert the IMS ResultSet to a DB2 Result Set using the GTT.

The output parameters defined for the stored procedure need to be changed to return a result set instead of individual fields.

The client application needs to be modified to accept a DB2 result set instead of IMS individual fields.

The use of DB2 result sets provides a consistent and easier programming interface for processing output from a DB2 stored procedure that uses the (IMS Version 9) IMS DB Resource Adapter to access IMS data.

9.2.2 GSAM metadata support

IMS DB Resource Adapter support for GSAM was enabled by Small Programming Enhancement PQ93785/UQ93241. This provided a Java Batch Processing application program with a Java API to open, read, write, and close GSAM databases. However, the application was required to know the complete details of the GSAM database.

IMS Version 10 GSAM enables all of the supported data types to be stored into and read from a GSAM database record and provides automatic data conversion on behalf of the application. This reduces the specific GSAM database knowledge for an application developer. GSAM database support is now consistent with the IMS Java class library support for all other IMS database types.

Given a DLIDatabaseView name, the IMS Java class library will use the metadata information to capture the correct sequence of bytes in the record, read that sequence of bytes, and convert that sequence of bytes into the appropriate data type as defined by the metadata.

Example 9-3 demonstrates the use of the GSAMConnection class, the GSAMRecord class, and the DLIDatabaseView class. It shows how to obtain a connection to a GSAM DLIDatabaseView, open access to the GSAM file using a GSAM program communication block (PCB), read a record from the GSAM file, get data from the record using a label, and close the GSAM connection.

Example 9-3 Using the GSAMConnection class

```
// Create a IMS Java JDBC GSAM connection
    GSAMConnection gsamConnection = GSAMConnection.createInstance
    ("GSAMDLIIDatabaseView");

// Open a IMS Java JDBC GSAM connection
    this.gsamConnection.open("GSAMPCB",ioarea);

// Process IMS Java JDBC GSAM connection
    gsamRecord=gsamConnection.getNext("GSAMPCB");
gsamRecord.getString("LastName");

// Close IMS Java JDBC GSAM connection
    this.gsamConnection.close("GSAMPCB",ioarea);
```

In error situations, the application program will get the exception and it can choose to send it to:

- ▶ JBP //JAVAOUT DD PATH='/path/JVM™.out'
- ▶ JBP //JAVAERR DD PATH='/path/JVM.err'
- ▶ JBP job log

The new messages are:

- ▶ INVALID_DBVIEW_URL
DLIDatabaseView metadata class URL is not correct.
- ▶ INVALID_GSAM_PCB_NAME
GSAM PCB name is not correct.
- ▶ INVALID_GSAM_RECORD
GSAMRecord object reference is null.
- ▶ INVALID_GSAM_RSA
GSAM Record Search Argument (RSA) is null.

GSAM metadata migration considerations

Java Batch Processing applications that use the IMS Version 9 GSAM database support will not be affected. To convert an existing application:

- ▶ Use IMS Version 10 DLIModel Utility to create the GSAM DLIDatabaseView.
- ▶ Use the new GSAM fields generated by DLIModel Utility.
- ▶ Use the new GSAM API.

The DLIModel Utility for IMS Version 10 provides new GSAM record and field options, which are described in 9.3, “DLIModel Utility” on page 247.

IBM SDK V5 for z/OS

z/OS delivers a complete Java 2 Software Developer Kit (SDK). Previous versions of the SDK provided a reusable function to support transactional runtime environments, such as IMS. This capability allowed the Java Virtual Machine (JVM) to be initialized during IMS Java dependent region startup and to be “reset” after the IMS application program completed processing. This avoided the overhead of loading the JVM for each IMS application program schedule.

The new SDK provides a Class Sharing capability to replace the persistent reusable function. z/OS APAR OA11519 is recommended for cache class sharing.

IMS message GU/CHKP processing was altered for Java Dependent Regions (JDRs) when JDRs were initially implemented. For JDRs, an IMS message GU meant to get the message only and do not perform sync point processing. CHKP processing meant perform sync point processing only and do not get the next message. In the standard IMS model, message GU and CHKP meant perform sync point and retrieve the next input message.

For IMS Version 10, IMS JDRs GU/CHKP processing is consistent with the IMS standard model. Prior to IMS Version 10, IMS Java Dependent Region programming required an explicit checkpoint/rollback call using the IMS DB Resource Adapter Transaction class:

- ▶ IMSTransaction.getTransaction().commit()
- ▶ IMSTransaction.getTransaction().rollback()

For IMS Version 10, the IMS Java application does not need to perform an explicit commit before obtaining the next input message. An IMS Java application that does not issue explicit

ROLLBACK but uses U118 abend for rollback processing must be changed. An IMS Java application that uses the IMS Java hierarchical database interface for CHKP call processing must be able to process the next message off the message queue. Figure 9-7 presents the comparison between the old and new programming model support.

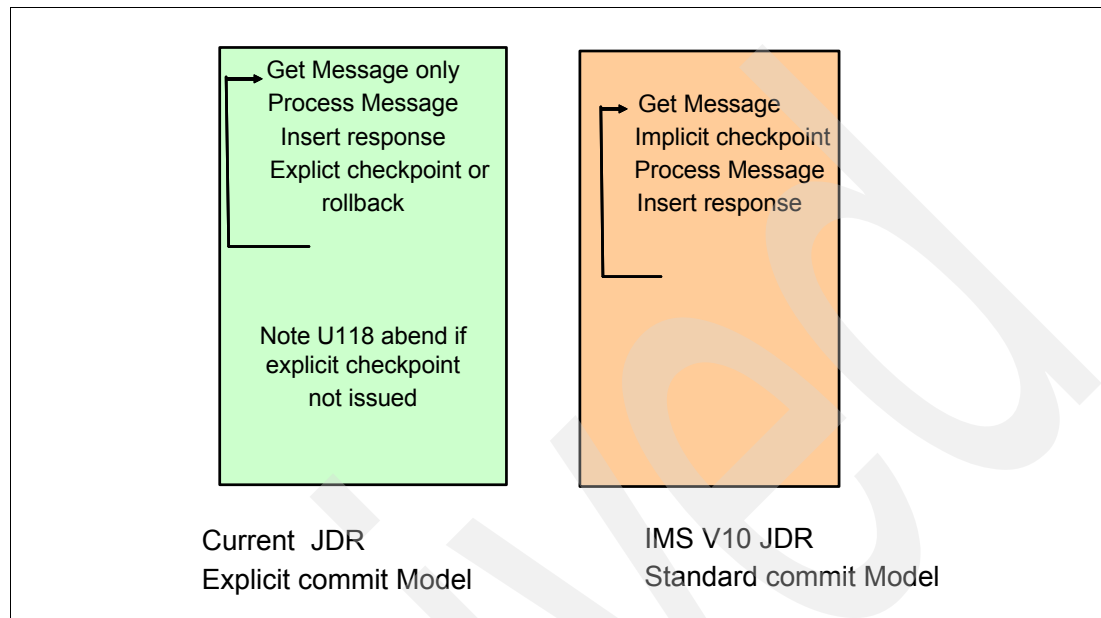


Figure 9-7 JDR commit model comparison

The IMS DB Resource Adapter class `com.ibm.ims.application.IMSApplication` is deprecated in IMS Version 10. A class marked as *deprecated* is considered obsolete but still works in IMS Version 10. You do not have to change your applications, but we highly recommend that you do.

The `com.ibm.ims.base.DLISecondaryIndexInfo` class has been removed from the library. This will only impact you if you did not use `DLIModel` to generate the metadata classes (the database view) or if you use the `DLISecondaryIndexInfo` class explicitly in your code.

The `com.ibm.ims.db.SecondaryIndexInfo` class has been renamed to `com.ibm.ims.base.SecondaryIndexInfo`.

You will be impacted only if you use the class directly in your code. The metadata that is generated by the `DLIModel` Utility is not affected.

9.3 DLIModel Utility

The IMS Version 10 `DLIModel` Utility adds support for GSAM databases. It is now able to parse program control blocks (PCBs) that refer to GSAM databases and generate the appropriate IMS Java `DLIDatabaseView` class. In addition, the IMS Version 10 `DLIModel` Utility adds support for XQuery, including new XML Schema annotations. (The information about XML schemas for the `DLIModel` Utility is updated as described in the topic 9.5, “XML database and XQUERY” on page 259). The utility now also generates separate program specification block (PSB) and database definition (DBD) XML Metadata Interchanges (XMIs), as well as the capability to import the PSB and DBD XMIs.

The `DLIModel` Utility GUI enhancements are available as a plug-in to Eclipse.

Eclipse is a platform-independent software framework that provides a foundation for Integrated Development Environments (IDE). The IBM Rational Application Developer (RAD) and WebSphere Developer for z/OS (WDz) Integration Development Environments are built on the Eclipse platform.

Note: The previous version of the DLIModel Utility could only import COBOL that had been converted to the XML format.

The main path to launching the DLIModel Utility is as a new eclipse project. The DLIModel New Project Wizard is invoked through the Eclipse file menu system.

After the DLIModel wizard has completed and the panel is closed, the Eclipse workspace will default to a new perspective called the IMS perspective.

DLIModel uses the Eclipse editor and will display each PCB in the order that it is specified on the PSB. Segments and fields are displayed in the IMS hierarchical format.

Two versions of the IMS DLIModel Utility are available:

- ▶ An IMS-shipped version that runs from System Services or from the z/OS BPXBATCH utility.
- ▶ A technology preview IMS Web no charge download version that runs as a plug-in to Eclipse. The GUI can be installed in an Eclipse 3.0.1 or 3.0.2 level tool. It can also be installed in WebSphere Developer for z IDE.

You can download an Eclipse-based DLIModel Utility GUI plug-in at:

<http://www.ibm.com/software/data/ims/toolkit/dlimodelutility>

In Figure 9-8, you can see the versions of the IMS DLIModel Utility.

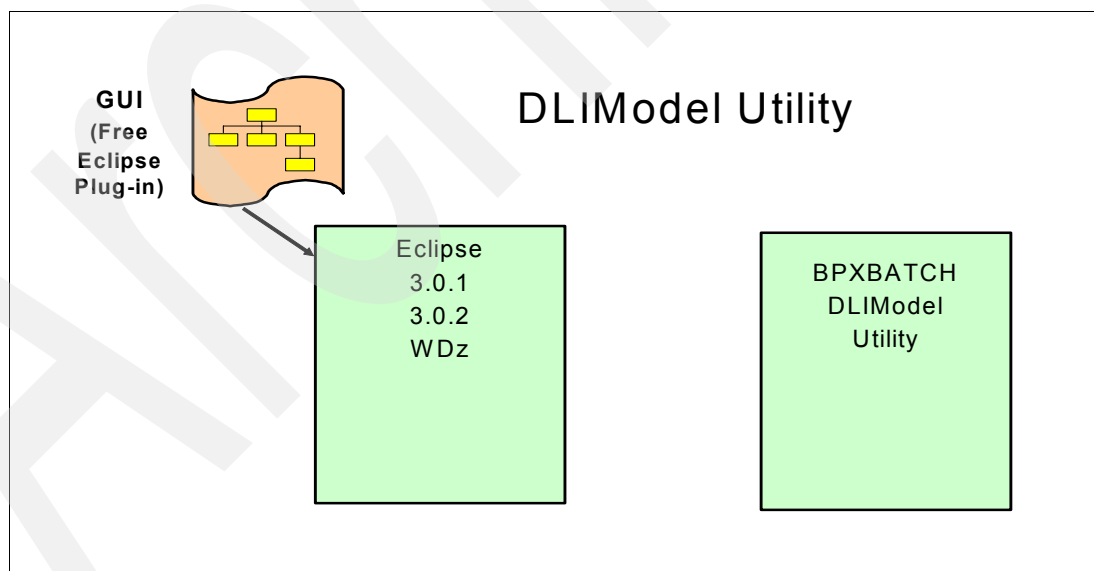


Figure 9-8 Versions of the DLIModel Utility

Use the DLIModel Utility to parse program specification blocks (PSBs), database definitions (DBDs), and COBOL copybook information and to construct an internal model of IMS metadata.

The constructed metadata model serves two purposes: to construct a visual representation of IMS resources for programming and DBA reference and to construct the required metadata for Java or IMS XML Database application development.

In addition to creating metadata, the DLIModel Utility also:

- ▶ Generates an annotated XML schema for IMS databases based on selected PCB definitions. These XML schema define the XML layout used when retrieving XML data from or storing XML data in IMS databases.
- ▶ Incorporates additional field information from XMI input files that describe COBOL copybooks
- ▶ Incorporates additional PCB, segment, and field information or overrides existing information
- ▶ Generates a database report, which is designed to assist application developers. The report is a text file that describes a particular PSB and all PCBs. This report includes segment names, field names, field types, database hierarchies, and other database specific information.
- ▶ Generates individual XMI descriptions of the PSB and its databases

The DLIModel Utility can process:

- ▶ All database organizations except MSDB, HSAM, and SHSAM
- ▶ All types and implementations of logical relationships
- ▶ Secondary indexes, except for shared secondary indexes
- ▶ Secondary indexes that are processed as stand-alone databases
- ▶ PSBs that specify field-level sensitivity

Generate or read PSB and DBD XMI

GenDBDXMI is an optional parameter to generate a DBD XMI file (named dbdname.xmi) that describes the database model based on the DBD. If GenDBDXMI is not specified, no DBD XMI file is generated. GenDBDXMI is mutually exclusive with either GenPSBXML or GENXMI.

GenDBDXMI=NO is equivalent to not including the GenDBDXMI parameter.

GenPSBXML is an optional parameter to generate a PSB XMI file (named psbname.xmi) that describes the database model based on the PSB. If GenPSBXML is not specified, no PSB XMI file is generated. GenPSBXML is mutually exclusive with either GenDBDXMI or GENXMI. If GenDBDXMI is not specified, the default is GenPSBXML.

GenPSBXML=NO is equivalent to not including the GenPSBXML parameter.

The current XMIPath=path optional parameter specifies the HFS directory where the DLIModel Utility writes the generated XMI.

DBDXMIPath is an optional parameter that specifies the HFS directory where the DLIModel Utility reads the input DBD XMI.

PSBXMIPath is an optional parameter that specifies the HFS directory where the DLIModel Utility reads the input PSB XMI. The current PSB statement has a new Source=ds or xmi optional parameter that specifies the location from which to import the PSB. The default search order is for XMI and then PSB source.

A new DbdName control statement has been added. The DbdName statement is optional and is used to define which DBD the utility will use when generating DBD XMI. The DbdName statement has the following parameters:

- ▶ Name - Process the DBD with the name.
- ▶ Nameprefix - Process all DBDs with the nameprefix in the specified DBD data set input file.
- ▶ * - Process all DBDs in the specified data set input file.
- ▶ Source=ds | xmi is an optional parameter that specifies the location to which to import the DBD. If not specified, the default search order is for XMI and then DBD source.

In Figure 9-9, you see the new option parameters.

Generate or Read PSB and DBD XMI

- New Option Parameters
 - GenDBDXMI
 - dbdname.xmi
 - GenPSBXML
 - psbname.xmi
 - DBDXMIPath= name
 - input DBD XMI
 - PSBXMIPath= name
 - input PSB XMI
 - PSB Statement
 - Source=ds | xmi
 - DbdName= name | nameprefix | *
 - Source=ds | xmi

Figure 9-9 Generate or read PSB and DBD XMI: new option parameters

The IMS DLIModel Utility has been enhanced to generate XMI from PSB and DBD source. The generated XMI can also be used as input to the DLIModel Utility.

GSAM now uses the GSAMDLI DatabaseView IMS Java class for metadata information about the GSAM database. The DLIModel Utility now supports GSAM databases.

Input

Shows the inputs and outputs of the DLIModel Utility. The actions of the utility are directed by control statements that you supply. PSB and DBD source members are read from their PDS or PDSE data sets and parsed by the utility to build an in-memory object model of the database structure and the PSB's view of that structure.

Note: IMS COBOL copybooks can only be processed by the GUI, and the BPXBATCH utility can only process COBOL XMI representations of the COBOL copybooks.

Output

The utility generates various outputs that were requested through control statements. You can request to have an IMS Java metadata class generated for the PSB processed, together with a corresponding easy-to-read DLIModel Java Report for the Java programmer to use.

You can request an XML description of the entire in-memory model (one description covers the PSB and all DBDs processed in the run).

You can also request a detailed trace file of the utility execution if a detailed trace file is necessary for problem resolution.

IMS Java metadata classes

The DLIModel Utility produces the necessary metadata classes needed to develop IMS Java applications. However, the Java developer only needs to reference the DLIModel Java Report for information about the classes.

DLIModel Java Report

The DLIModel Java Report summarizes the structure of the IMS databases in a way that allows you to create IMS Java applications and to code SQL queries against the databases. With the DLIModel Java Report, you do not have to interpret the syntax of the IMS Java classes or refer to the DBD or PSB source.

XML description of databases

An XML file, written in UTF-8 encoding, is produced by the utility if you specify `genXML=YES` in the `OPTIONS` control statement. It describes all of the PCBs and their referenced DBDs processed in the entire run of the utility. The XML that is produced by the utility is based on a metamodel of the IMS database that is defined in the Uniform Modelling Language (UML). This model is a package with a number of inheritance relationships to the Object Management Group (OMG) Common Warehouse Metamodel (CWM). However, only the IMS package itself is included and used in the DLIModel Utility.

XML schema

The generated XML schema, written in UTF-8 encoding, is an XML document that describes an IMS database based on a PCB. For more information about XML schemas, refer to 9.3.2, “XML schemas” on page 252.

In Figure 9-10 on page 252, you see the input to and outputs from the DLIModel Utility.

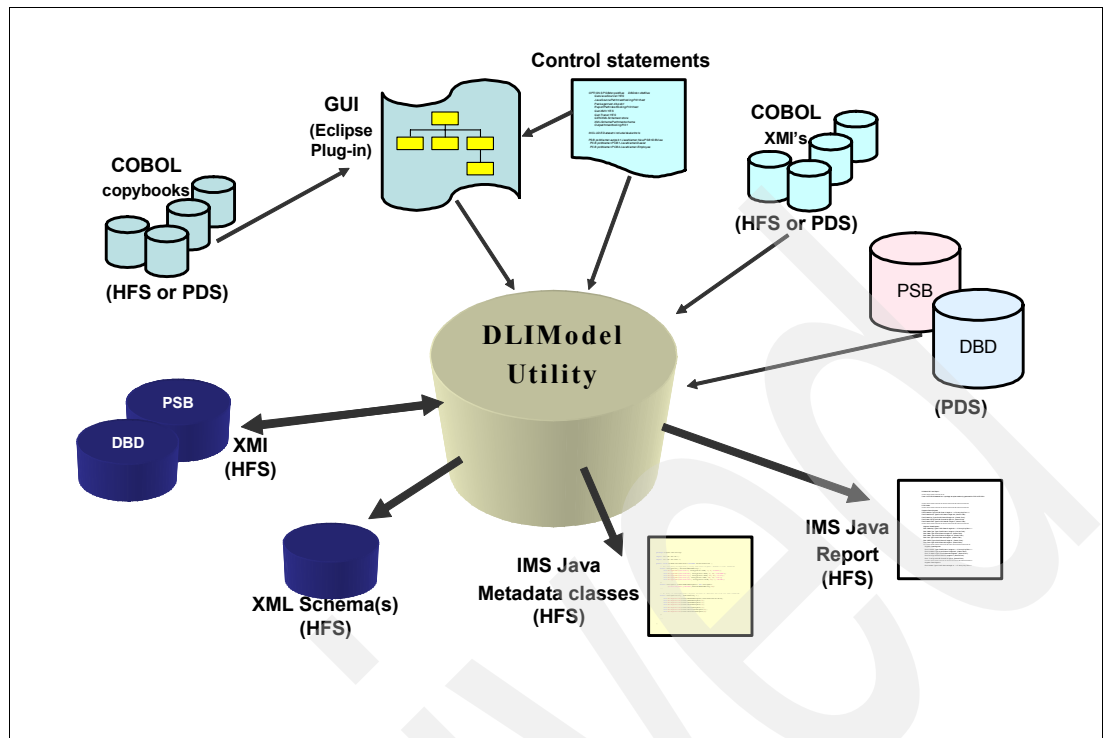


Figure 9-10 DLIModel Utility input and output

If you plan to use existing control statements as input to the DLIModel Utility GUI, you have to:

- ▶ Change the PSBDS, DBDDS, and INCLUDE paths to paths that are relative to their control statement files
- ▶ Verify that either Package or JavaSourcePath is specified, not both. If both are specified, Package value will take precedence

Additionally, you need to remove any COBOL XMI imports from their control statements. Then run the DLIModel Utility GUI with the modified control statements and use the Editor's context menu to import COBOL copybooks directly.

9.3.1 Java tooling

The DLIModel Utility automatically constructs the required IMS Java “metadata class” from PSB and DBD source. In IMS Version 7, the user was responsible for creating this class manually. The utility allows information about additional fields, long Java-style names, and data types to be supplied from user-coded control statements and from XMI descriptions (or both) of COBOL copybook members. The utility can produce XMI descriptions of databases that conform to the Object Management Group's Common Warehouse Metamodel 1.1. This utility eases development of Java applications and JDBC access to the IMS DB.

9.3.2 XML schemas

An XML schema is required to retrieve XML documents from or store XML documents into IMS. IMS uses an XML schema to validate an XML document that is being either stored into or retrieved from IMS. The generated XML schema, written in UTF-8 encoding, is an XML document that describes the XML view of an IMS database based on a PCB, segment, or a

field. The XML schema, not the application program, determines the structural layout of the parsed XML document in the database during storage and the generated XML document during retrieval.

Do not edit the XML schema. Make changes to the schema only by editing the control statements and running the DLIModel Utility.

Example 9-4 is an annotation from a schema that can be used for both storage and retrieval.

Example 9-4 Schema for storage and retrieval

```
<xsd:annotation>
  <xsd:appinfo>
    <ims:DLI mode="store"/>
  </xsd:appinfo>
</xsd:annotation>
```

If you are using the generated schema for storage, the schema also specifies whether the XML is stored intact or decomposed. An element can have an annotation that specifies that the element and all of its nested elements are to be stored intact. Example 9-5 is an annotation from an element that is stored intact.

Example 9-5 Element to store intact

```
<xsd:annotation>
  <xsd:appinfo>
    <ims:intact />
  </xsd:appinfo>
</xsd:annotation>
```

An annotation can coordinate the PCB with an element or complex type. Example 9-6 is an annotation from an element that is defined by the PCB.

Example 9-6 Element defined by PCB

```
<xsd:annotation>
  <xsd:appinfo>
    <ims:pcb alias="PCBAliasName" name="PCBNAME"/>
  </xsd:appinfo>
</xsd:annotation>
```

An annotation can coordinate the segment with an element or complex type. Example 9-7 is an annotation from an element that is defined by the segment.

Example 9-7 Element defined by the segment

```
<xsd:annotation>
  <xsd:appinfo>
    <ims:segment alias="SegmentAliasName" name="SEGNAME"/>
  </xsd:appinfo>
</xsd:annotation>
```

In certain cases, an annotation will coordinate a simple data type with the existence and value of a field. Example 9-8 on page 254 is an annotation from an element that is defined by the field.

Example 9-8 Element defined by the field

```
<xsd:annotation>
  <xsd:appinfo>
    <ims:field alias="FieldAliasName" name="FLDNAME"/>
  </xsd:appinfo>
</xsd:annotation>
```

The DLIModel Utility generates an XML schema from the following input:

- ▶ DBDs
- ▶ One PSB
- ▶ COBOL copybook XMI (optional)
- ▶ Control statements

In a single run of the DLIModel Utility, you can generate schemas from all PCBs, segments, or fields in the PSB that is to be processed by the utility or for only the PCBs, segments, or fields that you specify.

The generated XML schemas are stored in the path that you specify in the control statements. The file names have the following format:

PsbName-pcbJavaName.xsd

An XML schema of a sample dealership database is provided in the HFS directory pathprefix/usr/lpp/ims/imsjava91/samples/dealership/AUTPSB11-Dealer.xsd.

Table 9-1 lists the control statements and parameters that are required and optional for generating an XML schema.

Table 9-1 Control statements and parameters to generate an XML schema

Control statement	Required parameters	Optional parameters
OPTIONS	GenJavaSource GenXMLSchema	XMLSchemaPath
PSB	PSBName	none
PCB (optional)	PCBName	GenXMLSchema XMLRootElement
FIELD (optional)	FieldName	XMLType XMLStorageType Overflow
SEGMENT (optional)	SegmentName	none
SIDASEG (optional)	Xpath Source Field	none

For more information about the DLIModel Utility, refer to *IMS Version 10: System Utilities Reference*, SC18-9968.

9.4 IMS SOAP Gateway

IMS SOAP Gateway is a separate product that is available at no charge from the IMS Web site.

The IMS SOAP Gateway is an XML-based connectivity Web service solution that integrates IMS applications in an SOA environment. This diagram shows a SOAP client accessing IMS via the IMS SOAP Gateway Server.

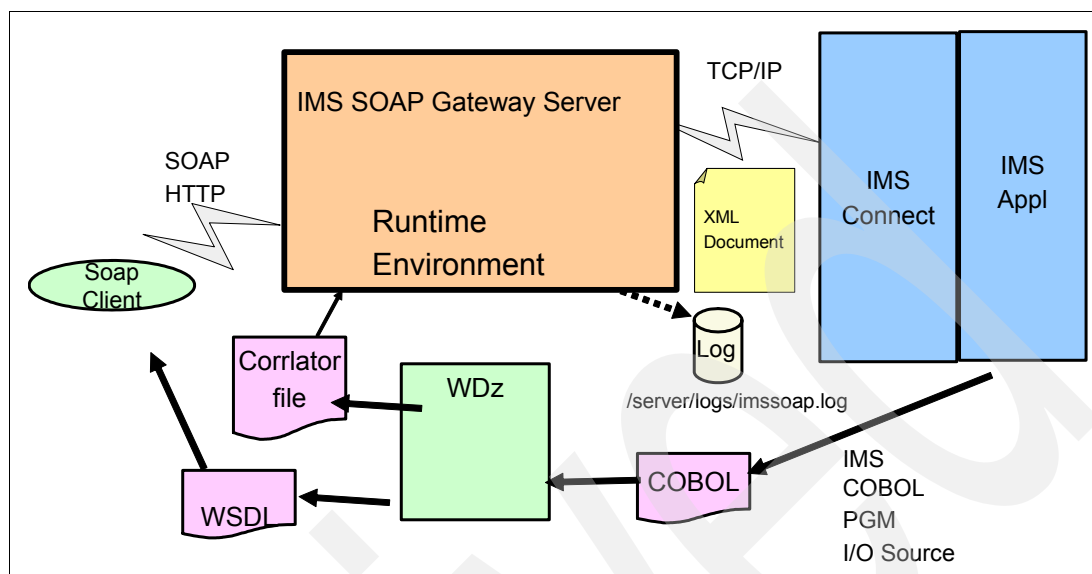


Figure 9-11 IMS SOAP Gateway overview

IMS SOAP Gateway Server is the HTTP endpoint for the SOAP Client. *SOAP* is a protocol for exchanging XML-based messages over a network using HyperText Transfer Protocol (HTTP). HTTP is a protocol used to transfer information over TCP/IP. To deploy an IMS application as a Web service, you need to create a Web service description language (WSDL) file.

A WSDL file is an XML document that describes a Web service. WSDL files are used by others (for example, the client that invokes the service) to discover the service and to understand how to invoke the service. It specifies the location of the service and the operations that the service exposes. WebSphere Developer for zSeries (WDz) can be used to generate a WSDL file.

After its introduction with IMS Version 9, IMS SOAP Gateway has been enhanced prior to IMS Version 10 by a new version and a fix pack. This is the list of enhancements IMS SOAP Gateway provides in Version 9.2:

- ▶ **SSL/HTTPS** - HTTPS provides secure connection between the SOAP Client and the IMS SOAP Gateway. SSL provides secure connection between the IMS SOAP Gateway and IMS Connect.
- ▶ The XML Adapter supports existing IMS application programs without modifications. XML Adapter requires PTF UK17000 (PK24912 and PK29938) applied to IMS Version 9 and provides a new object code only IMS Connect user message exit HWSSOAP1. It is included in IMS Version 10.
- ▶ The Windows runtime SDK fix supports Daylight Saving Time.
- ▶ zLinux runtime consolidates server support on the z platform and the SDK fix supports Daylight Saving Time.
- ▶ **Accessibility**

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products successfully. The IMS SOAP

Gateway Administrative Console, documentation, and Deployment Utility are all accessible and include the following accessibility features:

- Keyboard-only operation
- Interfaces that are commonly used by screen readers
- Keys that are tactilely discernible and do not activate just by touching them
- Industry-standard devices for ports and connectors
- The attachment of alternative input and output devices

The fix pack contains the following enhancements:

- ▶ AIX runtime SDK Fix supports Daylight Saving Time
- ▶ Dynamic Load Connection Bundle
- ▶ zLinux documentation update

9.4.1 IMS SOAP Gateway enhancements for IMS Version 10

IMS SOAP Gateway supports asynchronous callout to Web Services capability, which utilizes a number of enhancements in IMS Version 10:

- ▶ IMS Connect Resume TPIPE with Alternate ClientID
- ▶ IMS OTMA Resume TPIPE Security
- ▶ IMS OTMA destination routing descriptors

We have described these features in Chapter 8, “Connectivity enhancements” on page 203.

Asynchronous Callout to Web Service with XML Adapter

Figure 9-12 shows an asynchronous callout scenario to a Web Service with the XML adapter in use.

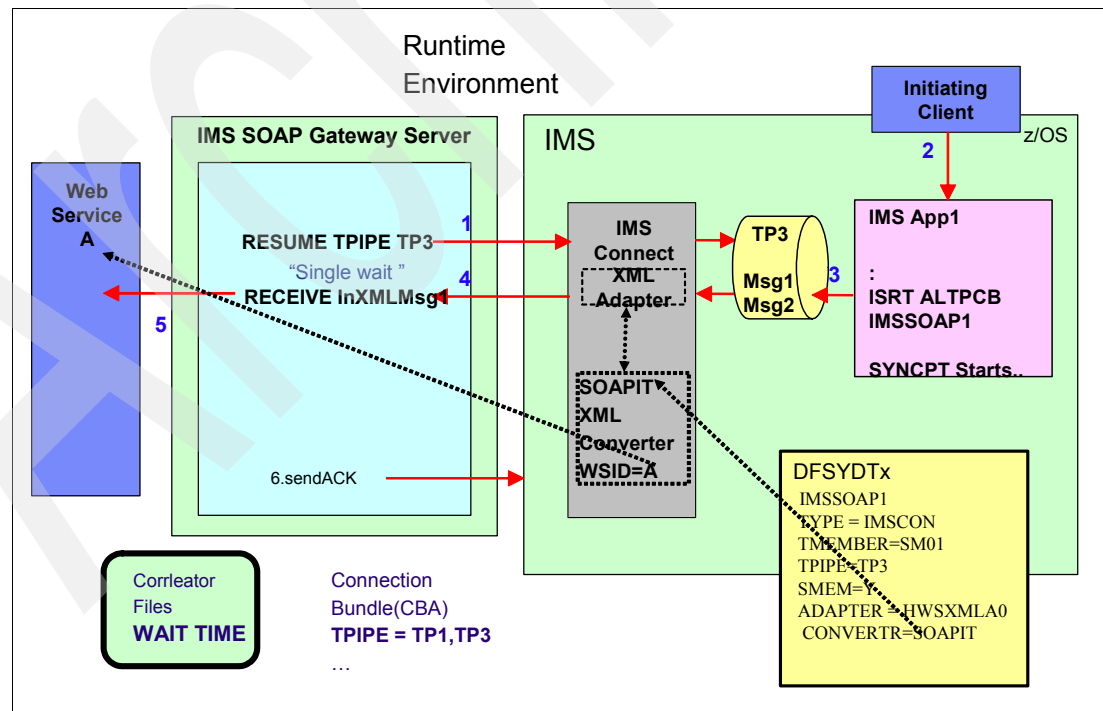


Figure 9-12 Asynchronous Callout to Web Service with XML Adapter

In this scenario, an IMS Application program does an insert to an ALTPCB whose destination is an OTMA Descriptor. The OTMA Descriptor is used to define the OTMA TPIPE for the enqueue of the message.

Steps 1, 2, and 3 schedule the IMS application, which results in the insert to ALTPCB defined by the OTMA Descriptor that represents IMS SOAP Gateway.

IMS SOAP Gateway will use the Resume TPIPE ALT-ClientID protocol to retrieve the message (1). IMS Connect will invoke the COBOL Adapter to perform an XML transformation of the message and send the message to the IMS SOAP Gateway (4). IMS SOAP Gateway receives the messages and sends it to the Web Service (5).

An optional step is that the Web Service creates a response to the message request. IMS SOAP Gateway sends the response to IMS (6). This results in the scheduling of an IMS Application program to process the response.

Asynchronous callout to Web Service with XML Adapter and a response

Figure 9-13 shows an asynchronous callout scenario to a Web Service with the XML Adapter in use and with a response.

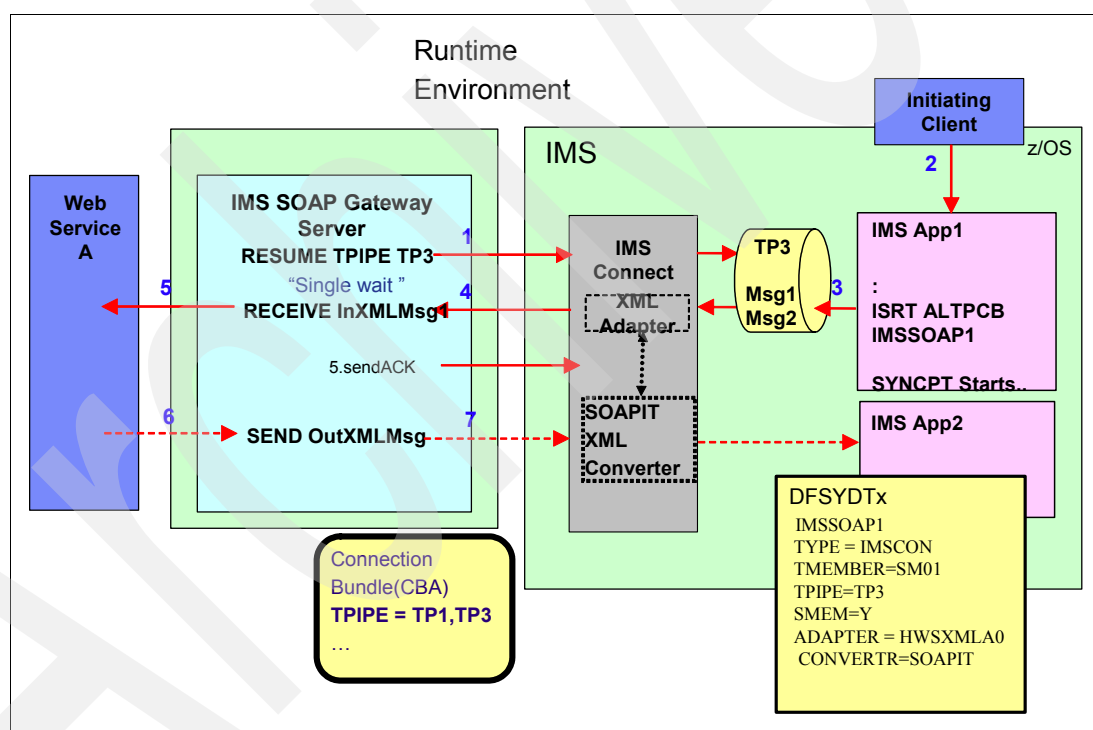


Figure 9-13 Asynchronous callout to a Web Service with the XML Adapter with a response

In this scenario, an IMS Application program does an insert to an ALTPCB whose destination is an OTMA Descriptor. The OTMA Descriptor is used to define the OTMA TPIPE for the enqueue of the message.

Steps 1, 2, and 3 schedule the IMS Application, which results in the insert to ALTPCB defined by the OTMA Descriptor that represents IMS SOAP Gateway.

IMS SOAP Gateway will use the Resume TPIPE ALT-ClientID protocol to retrieve the message (1). IMS Connect will invoke the COBOL Adapter to perform an XML transformation

of the message and send the message to the IMS SOAP Gateway (4). The IMS SOAP Gateway receives the message and sends it to the Web Service (5).

An optional step is for the Web Service to create a response to the message request. The IMS SOAP Gateway sends the response to IMS (6). This results in the scheduling of an IMS Application program to process the response.

Asynchronous callout to Web Service without the XML Adapter

Figure 9-14 shows an asynchronous callout to a Web Service without the XML Adapter scenario.

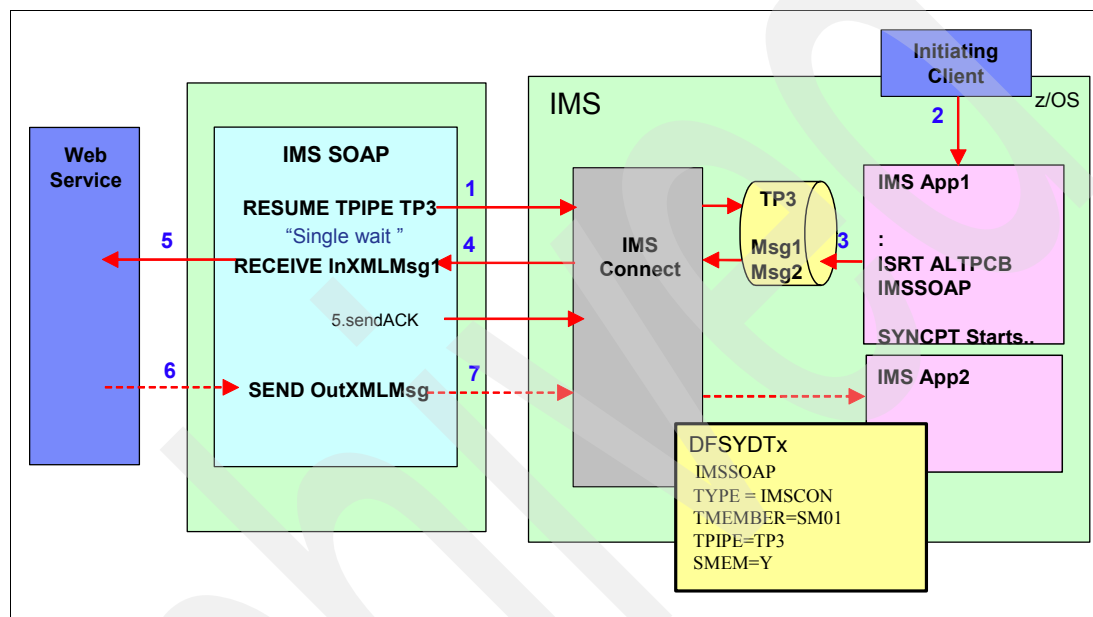


Figure 9-14 Asynchronous callout to a Web Service without an XML Adapter

In this scenario, an IMS Application program does an insert to an ALTPCB whose destination is an OTMA Descriptor. The OTMA Descriptor is used to define the OTMA TPIPE for enqueue of the message.

Steps 1, 2, and 3 schedule the IMS Application, which results in the insert to the ALTPCB defined by the OTMA Descriptor that represents IMS SOAP Gateway.

The IMS SOAP Gateway will use the Resume TPIPE ALT-ClientID protocol to retrieve the message (1). IMS Connect will invoke the COBOL Adapter to perform the XML transformation of the message and send the message to the IMS SOAP Gateway (4). The IMS SOAP Gateway receives the message and sends it to the Web Service (5).

An optional step for the Web Service is to create a response to the message request. The IMS SOAP Gateway sends the response to IMS (6). This results in the scheduling of an IMS Application program to process the response.

Note: The IMS Application needs to create an XML data stream in this scenario.

9.5 XML database and XQUERY

Because XML and IMS databases are both hierarchical, IMS is a natural database management system for managing XML documents. First introduced with IMS Version 9, IMS allows you to easily receive and store incoming XML documents, as well as compose XML documents from existing information that is stored in IMS databases.

For example, you can:

- ▶ Compose XML documents from all types of existing IMS databases, to support, for example, business-to-business on demand transactions and intra-organizational sharing of data.
- ▶ Receive incoming XML documents and store them in IMS databases. These databases can be existing databases or new databases. XML documents are stored decomposed: the document is parsed and element data and attributes are stored in fields in segments as normal IMS data. This is appropriate for data-centric documents.

Note: IMS XML DB allows applications to view a traditional IMS database as an XML database and to use an IMS database to store XML documents.

You can store XML documents decomposed, intact, or in a combination of decomposed and intact. In *decomposed storage mode*, the incoming document is parsed and element data and attributes are stored in fields as normal IMS data. Decomposed storage is appropriate for data-centric documents. In *intact storage*, the incoming document, including its tags, is stored directly in the database without IMS being aware of its structure. Intact storage is appropriate for document-centric documents.

To store XML in an IMS database or to retrieve XML from IMS, you must first generate an XML schema and the Java metadata class for IMS using the DLIModel Utility. The metadata and schema are used during the storage and retrieval of XML. Your application uses the Java JDBC user-defined IMS functions `storeXML` and `retrieveXML` to store XML in IMS databases, create XML from IMS data, or to retrieve XML documents from IMS databases.

9.5.1 IMS DB Resource Adapter XML API

An XML document can be composed from an IMS database.

As shown in Figure 9-15 on page 260, the `retrieveXML` is used to access the IMS database and create an XML document based on the DLIModel Utility-generated XML schema. The XML root element segment is provided as an input parameter to the `retrieveXML` User Defined Function (UDF). The *FROM* clause identifies which PCB in the PSB is to be used and the lowest level segment in the path. The *WHERE* clause predicate qualifies which segments to use to build the XML document.

During XML retrieval, IMS database segments are retrieved, fields are converted to the destination XML encoding, tags and XML syntactic information defined in the DLIModel-generated XML schema are added, and the XML document is composed.

The *root* of an XML document can be any segment in the IMS hierarchy.

The `retrieveXML` is used to access the IMS DB and create an XML document based on the DLIModel Utility-generated XML schema, but you can only retrieve the selected IMS records and build an entire XML document from the IMS records.

In Figure 9-15, you see an example of *retrieveXML*.

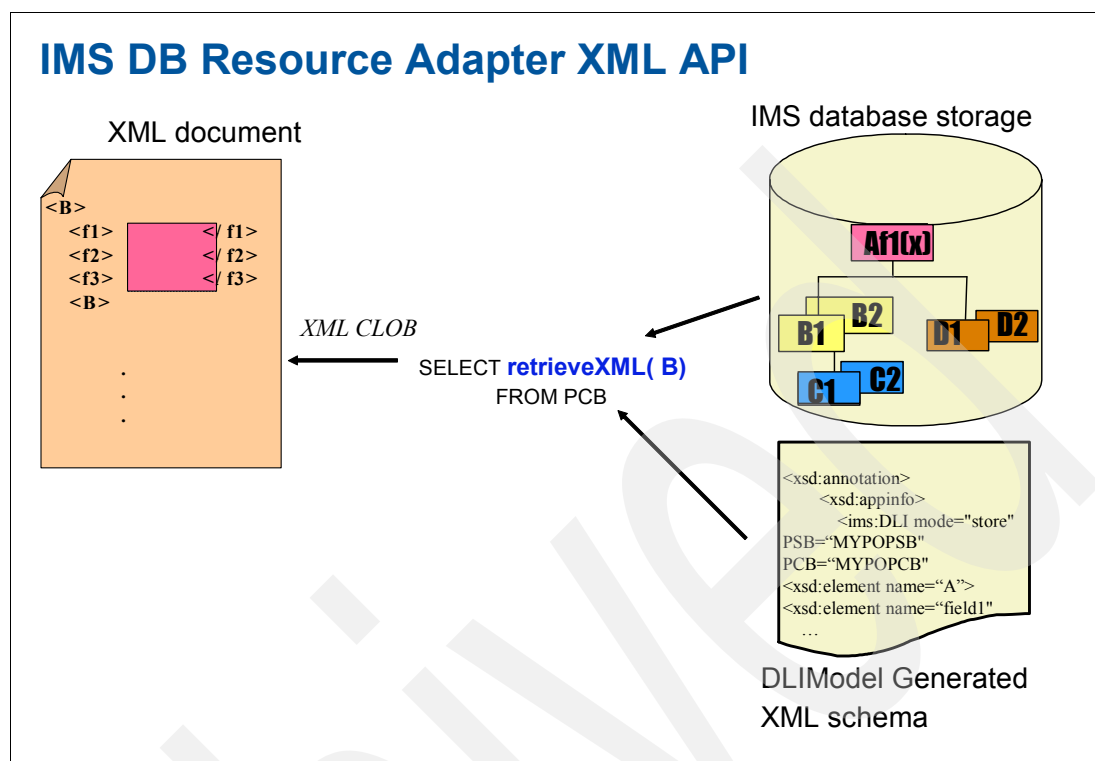


Figure 9-15 Example of *retrieveXML*

9.5.2 IMS DB Resource Adapter XQuery API

In IMS Version 10, the IMS Version 9 XML database support is extended by *XQuery*.

XQuery is a functional programming language that was designed by the World Wide Web Consortium (W3C) to meet specific requirements for querying XML data. XQuery is based on the structure of XML and leverages this structure to provide query capabilities for the same range of data that XML stores.

The IMS DB Resource Adapter XQuery support extends the *retrieveXML* user-defined function (UDF) by adding a second parameter. The second parameter allows the passing of an XQuery 1.0 expression. The expression is evaluated relative to the *retrieveXML* context and returned to the result set as a character large object (CLOB) value.

Figure 9-16 on page 261 shows you the entire IMS DB as an XML document and enables the return of specific IMS data based on the XQuery.

For IMS XQuery support, the XQuery 1.0 and XPath 2.0 Data Model serve two purposes:

- ▶ They define the information contained in the input to be used by the IMS XQuery processor.
- ▶ They define all permissible values of expressions in the XQuery and XPath languages that can be evaluated by the IMS XQuery processor.

The IMS DB Resource Adapter is packaged in `imsjava.jar`. The IMS XQuery function resides in a separate package (`imsxquery.jar`).

IMS DB Resource Adapter XQuery API

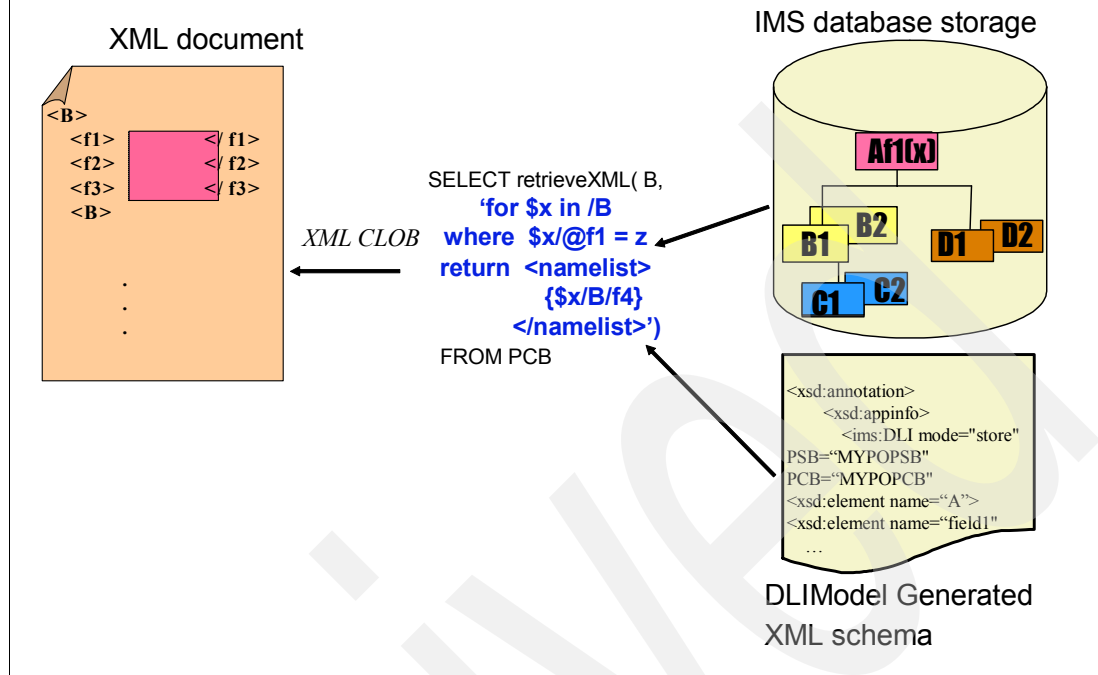


Figure 9-16 Example of retrieveXML with XQuery

XQuery provides a way to view a collection of XML documents and to be able to search, aggregate, evaluate, and essentially pick and choose the parts of an XML collection that are important and then convert that resulting data into XML.

Note: Because IMS XQuery is an extension to IMS XML DB, existing DLIModel-generated XML schemas can be used by the IMS XQuery processor to compose XML documents.

9.5.3 Migration considerations for XQuery

The IMS XQuery function uses features present in the IBM SDK Version 5 for z/OS runtime. As a result, you will need to migrate Java applications to IBM SDK Version 5 for z/OS support before you can use XQuery support, which should require no changes to the applications themselves because the SDK is backward compatible.

This function also introduces a new format for representing the XML structure of an IMS database in the defining IMS XML schema. However, it is not a requirement to migrate the XML schema generated under IMS Version 9. Older style XML schemas can be used seamlessly with applications written for IMS DB Resource Adapter XQuery API.

For more information about XML, refer to Chapter 8, "XML storage in IMS databases", in *IMS Version 10: Database Administration Guide*, SC18-9704, *IMS Version 10: System Utilities Reference*, SC18-9968, and *IMS Version 10: Application Programming API Reference*, SC18-9699.

Archived

Installation and migration considerations

This chapter provides the requirements for installing the major enhancements for IMS Version 10. It includes the following sections:

- ▶ Hardware and software requirements:
 - Packaging, prerequisites, and coexistence
 - Knowledge-Based Log Analysis (KBLA) coexistence
 - RECON upgrade
- ▶ IMS library changes
- ▶ IMS installation verification procedure (IVP) enhancements
- ▶ System definition and execution parameter changes
- ▶ Installation and migration tasks, including:
 - Replacing Security Maintenance Utility (SMU) security

10.1 Hardware and software requirements

This topic discusses the hardware and software requirements for IMS Version 10.

10.2 Hardware requirements

The processor on which IMS Version 10 runs must be:

- ▶ A System z or z/Series machine running in z/Architecture® mode (Enterprise Systems Architecture (ESA) mode is not supported by IMS Version 10.)
- ▶ Capable of running z/OS Version 1 Release 7 or later
- ▶ A 64-bit processor, such as those listed in Table 10-1

Table 10-1 IBM 64-bit processors

Machine name	Machine type
IBM eServer zSeries 900 (z/900)	2064
IBM eServer zSeries 800 (z/800)	2066
IBM eServer zSeries 990 (z/990)	2084
IBM eServer zSeries 890 (z/890)	2086
IBM System z9™ EC (formerly z9 109)	2094
IBM System z9 BC	2096

For more information about these processors, see:

<http://www.ibm.com/systems/z/hardware/>

10.2.1 System console

The console requirements of z/OS Version 1 Release 7 or later apply.

10.2.2 Tape units

IMS supports IBM 3480 and later tape units (or equivalent products) for installation and maintenance. IMS supports the tape block sizes greater than 32 760 bytes for the output of the following utilities:

- ▶ Database Image Copy utility (DFSUDMP0)
- ▶ Online Database Image Copy utility (DFSUICP0)

10.2.3 Coupling facilities

Sysplex data sharing (including data caching and Virtual Storage Option (VSO) data sharing) with internal resource lock manager (IRLM) V2.2 requires a Coupling Facility (CF) level 9 or later. Shared Queues and shared-expedited message handling (EMH) support require a Coupling Facility level 9 or later.

System-managed duplexing requires a Coupling Facility level 11 or later and bidirectional CF-to-CF links (such as HiPerLink, ICB link, or IC link).

For more information about system-managed CF structure duplexing, see:

<http://www.ibm.com/servers/eserver/zseries/library/techpapers/gm130103.html>

10.2.4 Direct access storage devices (DASD)

During the binding of the IMS control block's load modules (specifically during the bind of the IMS VTAM control block's load monitoring module), both the binder work data set SYSUT1 and IMS.SDFSRESL library must reside on a device that supports a record size of 18 KB or greater. For all other system libraries and working storage space, any device that is supported by the operating system is allowed.

For IMS database storage, any device that is supported by the operating system is allowed within the capabilities and restrictions of Basic Sequential Access Method (BSAM), Queued Sequential Access Method (QSAM), Overflow Sequential Access Method (OSAM), and Virtual Storage Access Method (VSAM).

Restriction: IMS does not support VSAM Extended Addressability (EA).

The Database Image Copy 2 utility (DFSUDMT0) requires DASD controllers that support either:

- ▶ The concurrent-copy feature of Data Facility Storage Management Subsystem (DFSMS)
- ▶ The FlashCopy feature of the IBM Enterprise Storage Server (ESS)
- ▶ The SnapShot feature of the IBM RAMAC Virtual Array (RVA) storage system

FlashCopy and SnapShot might require microcode from IBM to activate their functionality. Also, the source and target data sets (databases and image copies) must reside on the same ESS or RVA hardware.

Details about the DASD storage requirements for the following subjects are in *IMS Version 10: Program Directory for Information Management System Transaction and Database Servers*, I108-7540:

- ▶ SMP/E system entries
- ▶ SMP/E data sets
- ▶ Target libraries
- ▶ Distribution libraries
- ▶ Install process
- ▶ Optional machine-readable material

10.2.5 Large sequential data set support

No specific hardware is required for this enhancement, but to take advantage of this support, hardware that has more than 65 535 tracks must be used.

10.2.6 Multiple Systems Coupling

When the physical link is channel-to-channel (CTC) and is dedicated to IMS, Multiple Systems Coupling (MSC) requires the System/370™ CTC adapter or a logical channel on the IBM 3088, ESCON®, or Fibre Channel connection (FICON®). MSC FICON CTC support requires that at least one IMS system is installed on an IBM zSeries machine with the FICON

channel and FICON CTC microcode. The other side (IMS) can be any processor with a FICON channel.

10.2.7 Parallel RECON access

The parallel RECON access function requires a Parallel Sysplex environment.

10.2.8 Terminals that are supported by IMS Version 10

Refer to *IMS Version 10: Release Planning Guide*, GC18-9717, for information about terminals that are supported by IMS Version 10.

10.2.9 Remote Site Recovery

Remote Site Recovery (RSR) requires:

- ▶ A Sysplex Timer® (if either data sharing or workload is spread across multiple CPCs)
- ▶ A high bandwidth control unit (such as a 3172)
- ▶ At least one tape unit at the tracking site

Coordinated Disaster Recovery support for IMS and DB2 requires that the DB2 logs reside on devices that support eXtended Remote Copy.

10.3 Software requirements

The z/OS service levels that are required for installation and execution are listed in *IMS Version 10: Program Directory for Information Management System Transaction and Database Servers*, I108-7540.

Before you install IMS Version 10, check with your IBM Support Center or use either Information/Access or Service Link to see if there is additional preventive service planning (PSP) information of which you need to be aware.

The minimum level of z/OS for IMS Version 10 is z/OS V1R7.

As with previous IMS versions, the High Level Assembler Toolkit is required to provide the assembler macros that IMS uses.

10.3.1 Packaging, prerequisites, and coexistence

IMS Version 10 packaging is the same as that for IMS Version 9. Transaction Manager is a prerequisite for Extended Terminal Option (ETO). Recovery Level Tracking RSR is a prerequisite for Database Level Tracking RSR.

IRLM 2.2 is the only IRLM shipped and supported with IMS Version 10. If IRLM is already installed (for example, if it was installed with DB2), you do not need to reinstall it:

- ▶ When using multiple IMS systems of the same release level on the same z/OS system, you need only one IRLM. If two or more IMS systems share data at the block level, they must use the same IRLM.
- ▶ When using multiple IMS systems of different release levels on the same z/OS system, you can have one IRLM or you can choose to use two or more IRLM address spaces. If

two or more IMS systems share data and are running on the same z/OS system, they should use the same IRLM.

- ▶ When using multiple IMS systems on different z/OS systems for inter-processor block-level data sharing, you must have one IRLM on each z/OS system.
- ▶ IRLM 2.2 connections to IRLM 2.1 are supported:
 - IRLM 2.1 is supported with IMS Version 8 and IMS Version 9
 - IRLM 2.1 is not supported with IMS Version 10
 - IRLM 2.2 with IMS Version 10 can connect to IRLM 2.1 with IMS Version 9 or IMS Version 8

The IMS Version 10 product number is 5635-A01. Table 10-2 shows the function modification identifiers (FMIDs) of the IMS Version 10 features.

Table 10-2 FMIDs for IMS Version 10

FMID	Feature description
HMK1010	System Services
JMK1011	Database Manager
JMK1012	Transaction Manager
JMK1013	Extended Terminal Option (ETO)
JMK1014	Recovery Level Tracking (RSR)
JMK1015	Database Level Tracking (RSR)
JMK1016	IMS Java On Demand Features
HIR2220	IRLM 2.2

PTF OA12935 must be applied to z/OS 1.7 systems for the exploitation of EWLM.

z/OS V1R8 is required if the Fast Replication function of Image Copy 2 and the Database Recovery utility are used. In addition to z/OS, you must install RACF, or an equivalent security product, in order to use security with IMS Version 10.

Database Recovery Control (DBRC) parallel RECON access requires z/OS DFSMSStvs, which is a separately orderable feature of z/OS V1R7 or z/OS V1R8. Special bids will be considered for IMS clients using the parallel RECON access function, who do not already have DFSMSStvs, to acquire DFSMSStvs for use restricted to IMS.

DFSMS APAR OA11468/PTF UA18949 is a prerequisite for Fast Path clients running at z/OS 1.7 or higher.

The manual *z/OS MVS Setting Up a Sysplex*, SA22-7625, has additional information about requirements when using particular functions in IMS Version 10.

10.3.2 Supported connections

Intersystem Communication (ISC) is supported with:

- ▶ IMS Version 10, IMS Version 9, and IMS Version 8
- ▶ CICS Transaction Server V3.2, V3.1, V2.3, and V2.2
- ▶ User-written software

IMS Multiple Systems Coupling (MSC) is supported with:

- ▶ IMS Version 10, IMS Version 9, and IMS Version 8

Shared Queues is supported with:

- ▶ IMS Version 10, IMS Version 9, and IMS Version 8

DB2 connections are supported with:

- ▶ DB2 Version 9, DB2 Version 8, and DB2 Version 7

Database Control (DBCTL) connections are supported with:

- ▶ CICS Transaction Server V3.2, V3.1, V2.3, and V2.2

10.3.3 Global Online Change coexistence

The Global Online Change coexistence small program enhancements (SPEs) allow lower level IMS systems to share an OLCSTAT data set with IMS Version 10. IMS Version 10 makes changes to the header record. The SPEs allow IMS Version 8 and IMS Version 9 to implement and tolerate these changes.

The change to the OLCSTAT data set is made by executing the Global Online Change utility (DFSUOLC0) with the appropriate parameters and SYSIN input. The parameters must include FUNC=INI (initialization) and other parameters to specify the currently active data sets. Statements in the SYSIN data set are used to specify the IMS subsystem. The values in the OLCSTAT data set should be noted so that these parameters and input statements can be created to match them. This initialization can be done while IMS systems are active.

10.3.4 OTMA coexistence SPEs

The Open Transaction Manager Access (OTMA) coexistence SPEs (PK47171 for IMS Version 8 and PK47172 for IMS Version 9) are required if *all* of the following conditions are true:

- ▶ OTMA Shared Queues solution for CM1 or CM0 is being used
- ▶ The Shared Queues front-end IMS is IMS Version 10 and the back-end IMS is IMS Version 9 or IMS Version 8
- ▶ The back-end IMS is using ALT-program communication block (PCB) to generate asynchronous output for OTMA clients
- ▶ OTMA clients, such as WebSphere MQ or IMS Connect, connect to the back-end IMS to retrieve the asynchronous output

Without the coexistence SPEs, the error symptom is that the OTMA client connected to the Shared Queues back-end IMS will fail to retrieve the ALT-PCB output messages.

10.3.5 IMS Connect coexistence

The integrated IMS Connect function that is provided with IMS Version 9 and IMS Version 10 can coexist with the IMS Connect 2.2 product (running with IMS Version 8). However, the IMS Connect 2.2 product cannot be installed in the same SMP/E zones as IMS Version 9 or IMS Version 10. If you install IMS Version 9 or IMS Version 10 in the same SMP/E zones as the IMS Connect 2.2 product, the integrated IMS Connect function will overwrite the IMS Connect 2.2 installation.

For communications between the integrated IMS Connect and IMS Version 8, you must install the following IMS Version 8 APARs:

- ▶ PQ87087
- ▶ PQ59431
- ▶ PQ63231

If you plan to use the IMS Version 10 IMS Connect XML-to-COBOL conversion support while coexisting with IMS Version 9, you must install the following prerequisites:

- ▶ Apply IMS Version 9 APARs PK24912 and PK29938.
- ▶ You must have the copybook for the COBOL IMS application program that processes the input messages from IMS SOAP (an XML-based messaging protocol) Gateway.
- ▶ You must increase the IMS Connect region size to accommodate the storage used by the XML converters. The XML converters run in a z/OS Language Environment enclave in the IMS Connect region and use about 33 MB of storage.
- ▶ You must install IMS SOAP Gateway Version 9.2 or higher. You can download IMS SOAP Gateway online at:

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

For the technical details of the IMS Connect XML-to-COBOL conversion support, see the *IMS Version 10: Communications and Connections Guide*, SC18-9703.

10.3.6 Other coexistence maintenance

Here, you find the coexistence maintenance that is required on earlier releases when they coexist with IMS Version 10:

- ▶ The System Management Enhancement Coexistence SPEs allow lower level IMS systems to coexist with IMS Version 10. Lower level IMS systems that process transactions submitted from the Operations Manager (OM) API will receive an *AD* status code if they reply to the input/output program communication block (IOPCB). System Management Enhancement Coexistence SPEs are:
 - IMS Version 8 - PK30188
 - IMS Version 9 - PK30189
- ▶ The Operations Manager (OM) Coexistence SPEs allow OM address spaces and IMSs at lower level of IMS to coexist in a Common Service Layer (CSL) environment with IMS Version 10 OM. Operations Management Coexistence SPEs are:
 - IMS Version 8 - PK27279
 - IMS Version 9 - PK27280

Resource consistency checking for global online change is not done by IMS Version 10. This function was available in IMS Version 8 and IMS Version 9. It checks to ensure that the data set names used for the active ACBLIB, FORMAT, MODBLKS, and MATRIX data sets are the same for all of the IMS systems. It is only done when global online change is be used (*OLC=GLOBAL* is specified in the *DFSCGxxx* member) and an RM structure is used. It can be disabled in IMS Version 8 and IMS Version 9 by specifying values for the *NORSCCC=* parameter in the *DFSCGxxx* member. You should disable resource consistency checking, because the use of resource consistency checking created a single point of failure. For example, the loss of the ACBLIB data set would affect all systems, because all systems use the same data set.

Note: IMS Version 10 does not do resource consistency checking. The NORSCC= parameter in the DFSCGxxx member can be specified for compatibility, but it is ignored by IMS Version 10.

If your IMSplex has a mixture of IMS Version 10 and IMS Version 9 or IMS Version 8 systems, resource consistency checking will apply only to the IMS Version 8 and IMS Version 9 systems. When resource consistency checking is used, only the IMS Version 8 and IMS Version 9 systems must use the same data sets.

The Resource Consistency Checking SPEs must be applied for the IMS systems to use resource consistency checking when an IMS Version 10 system is part of the IMSplex. Resource Consistency Checking Coexistence SPEs are:

- ▶ IMS Version 8 - PK32969
- ▶ IMS Version 9 - PK32970

For more detailed information about the coexistence, refer to *IMS Version 10: Release Planning Guide*, GC18-9717.

10.3.7 Remote Site Recovery (RSR) coexistence

IMS Version 10 RSR tracking system can process logs created by IMS Version 10, IMS Version 9, or IMS Version 8.

IMS Version 10 RSR Isolated Log Sender can send logs created by IMS Version 10, IMS Version 9, or IMS Version 8.

Logs created by IMS Version 10 cannot be processed by IMS Version 8 or IMS Version 9 tracking system or Isolated Log Sender.

10.3.8 Log records

Among the log records that have been changed for IMS Version 10 are the following types:

- ▶ 07, 08, 22, 4001, 4004, 4006, 4007, 4012, 4083, 45, 56FA, 5950, 5951, 5953, 5954, 5955, 5956, 5957, 5958, 56FA, and 7205

Note: This is *not* an exhaustive list of changed log records.

If you have application programs that process IMS log records, you need to examine them to see if they are affected by the changes to the log records. You can assemble DSECTs for IMS log records by using the ILOGREC macro. Refer to *IMS Version 10: Release Planning Guide*, GC18-9717, for a list of new and changed log records.

10.3.9 Knowledge Based Log Analysis coexistence and migration

The IMS Knowledge-Based Log Analysis (KBLA) Migration panel facilitates migrating KBLA from IMS Version 9 to IMS Version 10 by prompting users to enter IMS Version 10-related ISPF variables. This panel appears the first time that you run a new version of KBLA.

Note: Whenever you install a new release of IMS, you must migrate KBLA.

Users specify the appropriate SDFSRESL library for an IMS version in the Default Parameter Maintenance panel (Option 0.1). An audit in KBLA restricts the specifiable version number to either 9 or 10. Different SDFSRESL libraries for a version can be specified, such as 9, 91, 10, 101, and so forth. KBLA attempts to find the best match of the IMS version specified on a processing panel to the SDFSRESL data set supplied on the Option 0.1 panel.

With IMS Version 9, if a value was not specified for the *KBLA Loadlib* parameter in the KBLA ISPF Option 0.1 panel IMS K.B.L.A - Define KBLA Environment, KBLA defaulted to the specification associated with the IMS Version 9 resource library. KBLA always included this data set as the first default subsystem name (DSN) in the JOBLIB DD concatenation sequence.

IMS Version 10 renames the variable from *KBLA Loadlib*, which previously implied that a value was required, to *KBLA Test Loadlib*, which is intended to imply that a value is optional and perhaps only used for testing purposes. If specified, the associated data set will be included as the first DSN in the JOBLIB concatenation sequence.

When you upgrade the KBLA utilities to support IMS Version 10, the following functions are added:

- ▶ KBLA uses the correct log record, DSECTS, to process logs that are created by either IMS Version 9 or IMS Version 10.
- ▶ Optionally, a user can concatenate (on the JOBLIB DD statement) an additional JOBLIB data set other than the IMS.SDFSRESL data set associated with the current release.
- ▶ If not specified, the IMS Version 10 SDFSRESL name is set to *hlq.SDFSRESL* where *hlq* is the high-level qualifier for the environment.
- ▶ Additionally, the panel automatically propagates the IMS Version 9 level *KBLA Loadlib* value into *KBLA Test Loadlib*. You can nullify or set the value to a different name. If the panel determines that the *KBLA Loadlib* data set is the same data set as the existing IMS Version 9 data set, this value will be removed. The panel will provide an opportunity to enter another data set name as the *KBLA Loadlib*, or this variable can be left blank. When the JCL for KBLA utilities is generated, the JOBLIB will contain only the *SDFSRESL* data set name for the selected IMS version if the *KBLA Loadlib* variable contains blanks.

Note: If you are an IMS Version 9 KBLA user, you need to note that the previous user-set parameter values are retained during the migration and any new parameters that have been added to the IMS Version 10 KBLA environment are implemented as “optional” parameters.

With KBLA, you can directly format SNAP subrecord types of X'67xx' log records, and control blocks in the record can span segments, which can cause byte misalignment in the formatted output. IMS Version 10 corrects the formatting for these records so that they retain the same byte alignment, even when they span segments, thus making the formatted output easier to read.

The next several figures provide an example of migration actions.

Figure 10-1 on page 272 shows the ISPF Menu 0.1 panel prior to migration.

```

IMS K.B.L.A. - Define KBLA Environment
Command ==>
Panel Successfully Processed

TIME....15:06:30
DATE....2005/01/20
JULIAN..2005.020

Fill out the following variables and press ENTER .

KBLA Test Loadlib. . . . . IMSBLD.I91RTS14.CRESLIB
IMS Version 9   Reslib DSN IMSBLD.I91RTS14.CRESLIB
IMS Version     Reslib DSN
IMS Version     Reslib DSN
IMS Version     Reslib DSN
COPY1 DSN. . . . . IMSVS.BTSARCNO.RECON1
COPY2 DSN. . . . . IMSVS.BTSARCNO.RECON2

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 10-1 The ISPF Menu 0.1 panel prior to migration

Figure 10-2 on page 273 shows the KBLA Migration Panel. This Panel provides the ability to define the IMS Version 10 SDFSRESL entry and to override the *KBLA Test Loadlib* data set name.

If the Menu 0.1 panel on the previous page (Define KBLA Environment) shows all four RESLIB entry slots as being filled in, the migration panel automatically switches back to that panel, accompanied by an error message indicating that Option 0.1 contains all the entries that it can handle. In that case, you need to enter the IMS Version 10 IMS.SDFSRESL entry on the panel in place of one of the existing SDFSRESL entries. Upon successful exit from Option 0.1, KBLA returns to the Primary Option Panel.


```

IMS K.B.L.A. - Migration Panel
Command ==>
(C) Copyright IBM Corp. 2003. All Rights Reserved.

                                     TIME....15:02:20
                                     DATE....2005/01/20
                                     JULIAN..2005.020

Supply the indicated values and press ENTER .

IMS Version 10 SDFSRESL DSN IMSBLD.IMS10A.SDFSRESL

KBLA Test Loadlib . . . . . IMSBLD.I91RTS14.CRESLIB
(If the KBLA Loadlib DSN is the same as the SDFSRESL for IMS Version 9
it will be set to a null value)

To Perform Migration, press ENTER .
To Exit panel, press END .

```

Figure 10-2 The KBLA migration panel

If the name for the *KBLA Test Loadlib* is the same as the existing IMS Version 9 resource library data set, this value will be removed as shown on Figure 10-3.

```

IMS K.B.L.A. - Migration Panel
Command ==>

                                     TIME....15:17:05
                                     DATE....2005/01/20
                                     JULIAN..2005.020

Supply the indicated values and press ENTER .

IMS Version 10 SDFSRESL DSN IMSBLD.IMS10A.SDFSRESL

KBLA Loadlib . . . . .
(If the KBLA Loadlib DSN is the same as the SDFSRESL for IMS Version 9
it will be set to a null value)

To Perform Migration, press ENTER .
To Exit panel, press END .

```

Figure 10-3 The KBLA migration panel

In Figure 10-4 on page 274, you will see the menu 0.1 panel after the migration.

```

IMS K.B.L.A. - Define KBLA Environment
Command ==>

                                                                    TIME....15:19:53
                                                                    DATE....2005/01/20
Fill out the following variables and press ENTER .                JULIAN..2005.020

KBLA Test Loadlib. . . . .
IMS Version 9   IMS.SDFSRESL DSN IMSBLD.I91RTS14.CRESLIB
IMS Version 10  IMS.SDFSRESL DSN IMSBLD.IMS10A.SDFSRESL
IMS Version     IMS.SDFSRESL DSN
IMS Version     IMS.SDFSRESL DSN
COPY1 DSN. . . . . IMSVS.BTSARCNO.RECON1
COPY2 DSN. . . . . IMSVS.BTSARCNO.RECON2

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 10-4 The ISPF menu 0.1 panel after the migration

For more information, see Chapter 3, “Migration to IMS Version 10”, in *IMS Version 10: Release Planning Guide*, GC18-9717.

10.3.10 RECON upgrade

IMS Version 8 and IMS Version 9 RECONS can be upgraded to IMS Version 10 by executing the DBRC utility (DSPURX00) and using the CHANGE.RECON UPGRADE command with an IMS Version 10 SDFSRESL library. Before doing the upgrade, apply the correct SPE to your IMS:

- ▶ IMS Version 8 SPE PK06145 allows IMS Version 8 to use IMS Version 10 RECONS. SPE PK06145 for IMS Version 8 also supplies compatibility support that allows IMS Version 10 to invoke HALDB Online Reorganizations (OLR) for partitions that are accessed by IMS Version 8. IMS Version 8 cannot invoke HALDB OLR, but when this maintenance is applied, it can access partitions for which OLR is used.
- ▶ IMS Version 9 SPE PK06147 allows IMS Version 9 to use IMS Version 10 RECONS.

For more information about migrating RECONS and DBRC, see Chapter 7, “DBRC enhancements” on page 157.

10.3.11 IMS utilities coexistence

The Batch Backout (DFSBB000), Log Archive (DFSUARC0), and Log Recovery (DFSULTR0) utilities access one log. The release level of the utility must match the IMS release that was used to create the log.

The IMS Version 10 Database Recovery utility (DFSURDB0) accepts Image Copies, hierarchical indexed sequential access method (HISAM) Unload data sets, Change Accumulation data sets, and IMS logs as inputs. These inputs can be created by IMS Version 10, IMS Version 9, or IMS Version 8.

The Change Accumulation utility (DFSUCUM0) accepts IMS logs and Change Accumulation data sets as inputs. These inputs can be created by IMS Version 10, IMS Version 9, or IMS Version 8.

10.4 IMS library changes

The organization of the IMS Version 10 library is significantly different from that of earlier versions. The library has been reorganized and streamlined and includes new and changed titles. Also, as of IMS Version 10, none of the IMS information is licensed.

The reasons for reorganizing and redesigning the IMS library so dramatically are to achieve the following goals:

- ▶ Group similar information together into a more intuitive organization.
- ▶ Rewrite information to better support user tasks.
- ▶ Separate information into three basic types of topics:
 - Task
 - Concept
 - Reference
- ▶ Utilize the Darwin Information Type Architecture (DITA) open source tagging language.

The IBM strategic direction is to deliver product information in Eclipse information centers. Information centers provide the following advantages over PDF books:

- ▶ Improved retrievability: Users can search across the entire information center or set filters to search categories of information.
- ▶ Accessibility support
- ▶ Multi-language support

Table 10-3 illustrates the new organization of the IMS Version 10 library.

Table 10-3 Information moved between publications

IMS Version 10 titles	IMS Version 9 titles
<i>Release Planning Guide</i>	<i>Release Planning Guide</i>
<i>Installation Guide</i>	<i>Installation Volume 1: Installation Verification</i>
<i>System Definition Guide</i> <i>System Definition Reference</i>	<i>Installation Volume 2: System Definition and Tailoring</i>
<i>Exit Routine Reference</i>	<i>Customization Guide</i>
<i>Database Administration Guide</i>	<i>Administration: Database</i>
<i>System Administration Guide</i>	<i>Administration: System</i>
<i>Communications and Connections Guide</i>	<i>Administration: Transaction Manager</i>
<i>Operations and Automation Guide</i>	<i>Operations Guide</i>

IMS Version 10 titles	IMS Version 9 titles
<i>Application Programming Planning Guide</i> <i>Application Programming Guide</i> <i>Application Programming API Reference</i> <i>System Programming API Reference</i>	<i>Application Programming: Design Guide</i> <i>Application Programming: EXEC DLI Commands for CICS and IMS</i> <i>Application Programming: Database Manager</i> <i>Application Programming: Transaction Manager</i> <i>IMS Java Guide and Reference</i>
<i>Command Reference, Volume 1; Command Reference, Volume 2; and Command Reference, Volume 3</i>	<i>Command Reference Volume 1</i>
<i>Database Utilities Reference</i>	<i>Utilities: Database and Transaction Manager</i>
<i>System Utilities Reference</i>	<i>Utilities: System</i>
<i>Diagnosis Guide (unlicensed)</i> <i>Diagnosis Reference (unlicensed)</i>	<i>Diagnosis Guide and Reference (licensed)</i>
<i>Messages and Codes Reference, Volume 1: DFS Messages</i> <i>Messages and Codes Reference, Volume 2: Non-DFS Messages</i> <i>Messages and Codes Reference, Volume 4: IMS Component Codes</i>	<i>Messages and Codes Volume 1: Non-DFS Messages</i> <i>Messages and Codes Volume 2: DFS Messages</i>
<i>Messages and Codes Reference, Volume 3: IMS Abend Codes (unlicensed)</i>	<i>Failure Analysis Structure Table (licensed)</i>

Two new IMS Version 10 titles include:

- ▶ *IMSplex Administration Guide*
- ▶ *System Programming API Reference*

The content of the single-topic books listed here were distributed throughout the IMS Version 10 library:

- ▶ *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: Database Recovery Control (DBRC) Guide and Reference*, SC18-7818
- ▶ *IMS Version 9: IMS Connect Guide and Reference*, SC18-9287
- ▶ *IMS Version 9: Open Transaction Manager Access Guide and Reference*, SC18-7829

When you order IMS Database Manager or IMS Transaction Manager, you will receive one copy of each of the following IMS publications:

- ▶ *IMS Version 10: Installation Guide*, GC18-9710
- ▶ *IMS Version 10: System Definition Guide*, GC18-9998
- ▶ *IMS Version 10: System Definition Reference*, GC18-9966
- ▶ *IMS Version 10: Release Planning Guide*, GC18-9717
- ▶ *IMS Version 10: Program Directory for Information Management System Transaction and Database Servers*, I108-7540
- ▶ *IMS Version 10: Licensed Program Specifications*
- ▶ *IMS Version 10 Softcopy Library*, SK5T-7327 (a CD-ROM that contains all of the IMS information)

Note: IMS Version 10 is the final version of IMS in which the IMS information will be available in BookManager® format.

All of the IMS Version 10 information continues to be available in the XHTML and PDF formats in the Information Management Software for z/OS Solutions Information Center at:

<http://publib.boulder.ibm.com/infocenter/imzic>

The Information Center has been updated to include the new organization of information about IMS Version 10. The topics on the left side of the figure map to the titles of the IMS Version 10 publications that are changed as shown in Figure 10-5.

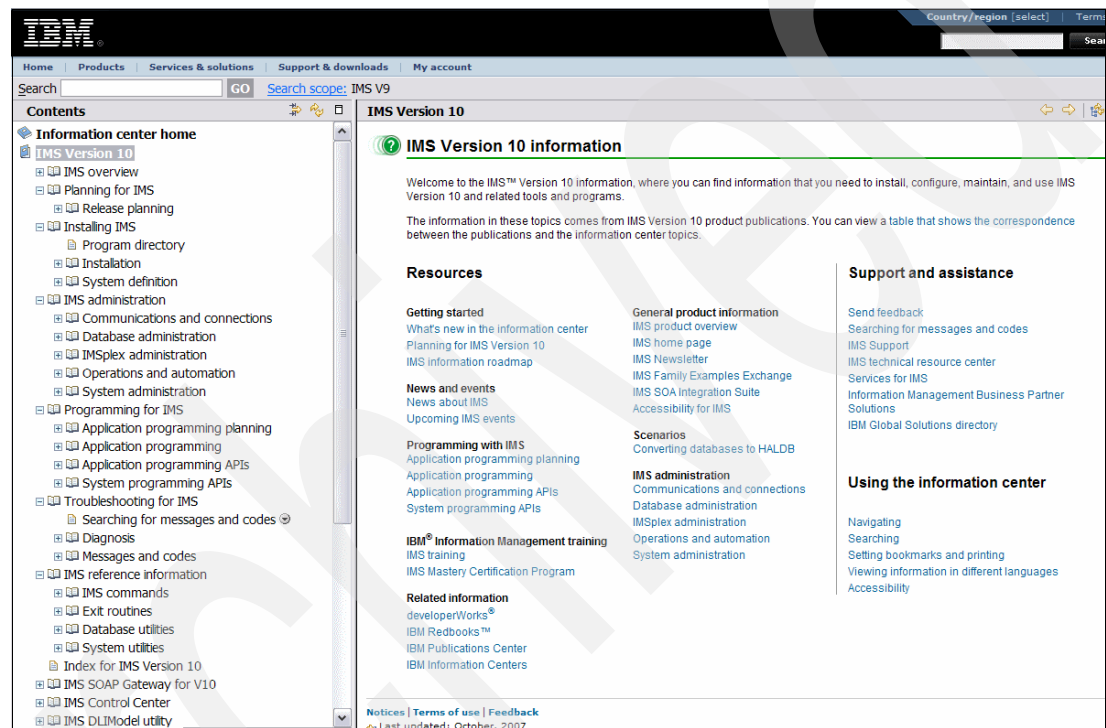


Figure 10-5 IMS Information Center

You can download or order additional copies of these publications or copies of any IMS publication from the IBM Publications Center at:

<http://www.elink.ibm.link.ibm.com/public/applications/publications/cgibin/pbi.cgi>

You can also download any of the IMS publications in softcopy format from the IMS Library page at:

<http://www.ibm.com/software/data/ims/library.html>

10.5 IMS IVP enhancements

The IMS Version 10 Installation Verification Program (IVP) provides new sample applications and examples to clarify existing functionality in IMS. Additionally, the IVP includes support for new functions that are introduced in IMS Version 10.

10.5.1 IVP enhancements for existing functions

Next, we list the IVP enhancements for existing functions.

Support of RACF implementation and SMU removal

To migrate current SMU security implementations to RACF (or an equivalent product), you might need to perform some or all of the following tasks:

- ▶ Review the SMU syntax conversion utilities provided by SPE APARs PK35433 and PK38522 for IMS Version 9 and PK49538 for IMS Version 10.
- ▶ Translate the current application group name (AGN) definitions into RACF only definitions.
- ▶ Add the four new security classes with the prefixes I, J, L, and M to the installation-defined class descriptor table if the classes are not already defined as default classes in the class descriptor table that is supplied by IBM. The four classes are:
 - IIMS - Program Specification Block (PSB)
 - JIMS - Grouping class for PSB
 - LIMS - Logical terminal (LTERM)
 - MIMS - Grouping class for LTERM
- ▶ Specify the new OPTIONS=SIGNON parameter on all applicable TYPE or TERMINAL system definition macros, or specify the new SIGNON=ALL I SPECIFIC parameter in the DFSDCxxx PROCLIB member for static terminals that are currently defined through SMU as terminals that are required to sign on.
- ▶ Specify the new type-1 Automated Operator Interface (AOI) definitions for the startup parameter AOI1= and on the system definition TRANSACT macro for type-1 AOI and type-2 AOI.
- ▶ Change the Time-Controlled Operation (TCO) and AOI definitions to use the System Authorization Facility (SAF) and DFSCCMD0 exit routine for security.
- ▶ Define the new RAS parameters in the IMS system definition SECURITY macro or the parameters on the existing parameter ISIS=.
- ▶ Change the Multiple Systems Coupling (MSC) link receive security definitions to use SAF and the DFSCCMD0 exit routine for security.
- ▶ Change the use of passwords for the /LOCK and /UNLOCK commands to use SAF and the DFSCCMD0 exit routine.
- ▶ Disable resource consistency checking for the MATRIX data set if you are going to have a mixed-version IMSplex that includes IMS Version 8 and do not intend to use SMU in that IMS Version 8 system. To disable resource consistency checking for the MATRIX data set, specify MODBLKS in the NORSCC= keyword of the DFSCGxxx PROCLIB member. This disables resource consistency checking for both the MATRIX and MODBLKS data sets.

Note: Ensure that your RACF definitions for the VSAM catalog data sets include IMS so that IMS can extend data sets and update the catalog with the new extent information.

New RACF security sub-options are added to the Sub-options panel of the Installation Verification Procedure (IVP). This panel provides the option of selecting whether RACF security is used. If this sub-option is selected, the IVP builds the necessary jobs and tasks to define resources to RACF and to set up the use of two IMS security user exit routines. This change is an enhancement to the C, D, and E steps of the IVP jobs and tasks.

For more information about replacing SMU security, see *IMS Version 10: Release Planning Guide*, GC18-9717.

A new RACF Security section is added to the Sub-options panel, which requires the user to specify whether RACF Security is to be used:

- The default is not to use this sub-option (no slash).

If the sub-option is selected, the IVP builds the necessary jobs and tasks to define resources to RACF and to set up the use of several IMS security user exit routines. The user can modify the sample RACF resource definition task. The sample user exit routines always authorize the user to the resources. This sub-option is not available for DB batch.

Figure 10-6 shows the new RACF Security options on Sub-Options panel.

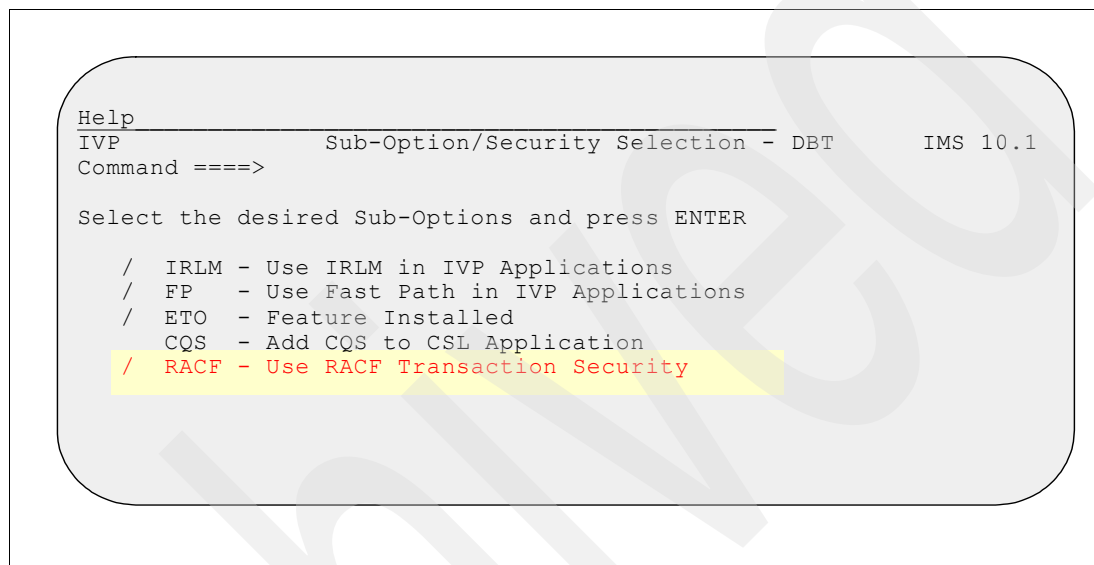


Figure 10-6 New RACF Security options on Sub-Options panel

IVP steps that implement RACF security

The steps to implement security are:

- IV_D218T
Includes actions that replace the IV_E316J actions in previous releases. The new step establishes IMS Security and provides an example of defining the IVP security definitions to RACF.
- IV_C105J
Provides a new step to replace the IV_E315J actions in previous releases. The new step assembles and binds the RACF security exits in *IMS.SDFSISRC* into *IMS.SDFSRESL* and replaces the uses of *DFSISIS0* with *DFSIRAS00*.
- IV_C210T
Browses the stage 1 source deck. Depending on the user security choice of using RACF or not, the SECURITY definition macro is also changed to specify whether the application resource access security authorization using RACF and exit routines are to be performed by IMS at execution time.

If the RACF Security option was checked on the Sub-option panel, the definition using RACF and exit routines is created as follows:

- SECURITY TYPE=(RAS,SIGNEXIT,TRANEXIT,RACFCOM),
SECLVL=(SIGNON,TRANAUTH)

If the security definition is to be done without using RACF, the definition is:

- ▶ SECURITY TYPE=(NORAS), SECLVL=(NOSIGN,NOTRAN)

IMS Java sample application

A new Java section is added to the Sub-options panel. If the sub-option is selected, the IVP adds the necessary jobs and tasks for the IMS JDBC application execution.

Figure 10-7 shows the new Java Application options on Sub-Options panel.

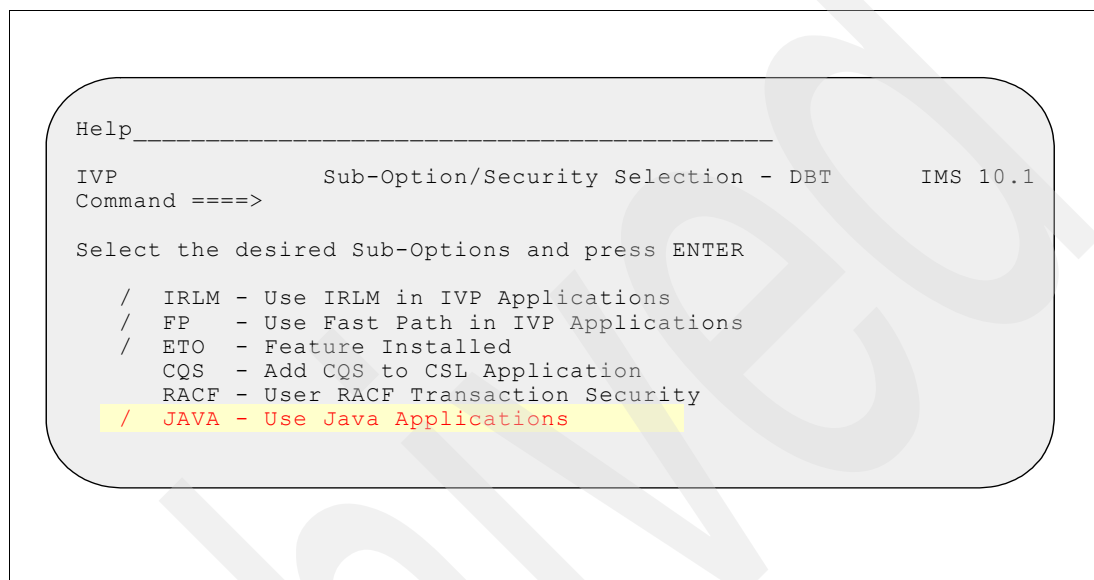


Figure 10-7 New Java Application options on Sub-Options panel

The Java Sample verifies that a Java workload can be run in an IMS Java Dependent Region environment. The changes in the IVP that support this capability are included in the “C”, “E”, “G”, and “H” steps.

The IVP provides the steps that:

- ▶ Add new database, application, and transaction definition statements to the IVP stage 1 source input.
- ▶ Add control statement members to the IMS.PROCLIB for Java virtual machine (JVM) initialization.
- ▶ Allocate the data set for the auto dealership database.
- ▶ Start an IMS control region.
- ▶ Start a JMP dependent region.
- ▶ Run a transaction in the JMP dependent region.
- ▶ Run a batch job in the JBP region to access databases.

Several new IVP variables have also been added to support the Java sample:

- ▶ IXUJPATH
Specifies the path name of the absolute Java libraries. The default value is /usr/lpp/ims/imsjava10. This substitutes a value for string *ImsjavaPath* in the IMS DFSJVMMS and DFSJVMMEV PROCLIB members.
- ▶ IXUSPATH
Specifies the path name of the sample Java libraries. The default value is

/usr/lpp/ims/imsjava10. This substitutes a value for string *SamplesPath* in the IMS DFSJVMMS and DFSJVMEV PROCLIB members.

- ▶ IXUJHOME
Specifies the path name of the Java installation libraries. The default value is /usr/lpp/java150/J1.5/bin. This substitutes a value for string *JavaHome* in the IMS DFSJVMMS and DFSJVMEV PROCLIB members.
- ▶ IXUJOUT
Specifies the path name of the HFS file for the output data. The default value is /tmp.
- ▶ IXUJERR
Specifies the path name of the HFS file for the error data. The default value is /tmp.

To support the IVP, new DBD and PSB sources have been added into the IMS.SDFSISRC library.

IMS Connect sample application

The IMS Connect sample application is available for DB/DC and XRF environments. The sample application is built in the Qx series of jobs and tasks.

To support the IMS Connect IVP, a new IVP variable and a new member in *IMS.SDFSISRC* have been added.

- ▶ New IVP variable:
 - IXUTCPIP: fully qualified DSNNAME for the TCP/IP Library
- ▶ New member in IMS.SDFSISRC:
 - DFSIVA9: sample IMS Connect client written in assembler that sends the PART transaction to IMS Connect and receives a response.

The IVP provides these modified steps:

- ▶ Step E206J
Assemble and bind the sample application program and user exits
- ▶ Step E302J
Add a configuration member in the PROCLIB data set to define the IMS Connect environment

The IVP provides these new steps:

- ▶ Step Q101J
Allocate the Recorder data set HWSRCRDR for output data capture
- ▶ Step Q203J
Start an IMS control region with OTMA enabled
- ▶ Step Q205J
Start the IMS Connect address space
- ▶ Step Q206J
Send a transaction message to IMS via IMS Connect and receive a response

10.5.2 IVP support for new functions

IVP has been enhanced to support various new features of IMS Version 10.

Operations Manager (OM) Subscribe/Unsubscribe sample

IMS Version 10 provides a TSO SPOC sample to view all messages and displays information from the audit trail.

The changes are delivered as enhancements to the CSL “E” steps of the IVP jobs and tasks.

Modified IVP steps:

- ▶ Step E303J
Add new parameter AUDITLOG= Audit Trail log data set name in the OM PROCLIB member CSLOI000
- ▶ Step E307T
Add CFRM and LOGR policy definitions for the new Audit Trail log data set

Dynamic Resource Definition User Interface (DRD UI)

Dynamic Resource Definition (DRD) is a new capability in IMS Version 10. The IVP provides a sample, using the DRD User Interface (DRD UI), that updates both a program and a transaction. The DRD User interface (DRD UI) is a general purpose interface that can be used to issue various resource definition and operational commands.

The IVP for DRD UI provides:

- ▶ A set of ISPF panels that provide direct support for DRD commands
- ▶ An IVP example of updating a program
- ▶ An IVP example of updating a transaction

A new panel provides you with options about actions that can be performed against resources. In the example shown in Figure 10-8, a request is made to update a resource by entering option 4.

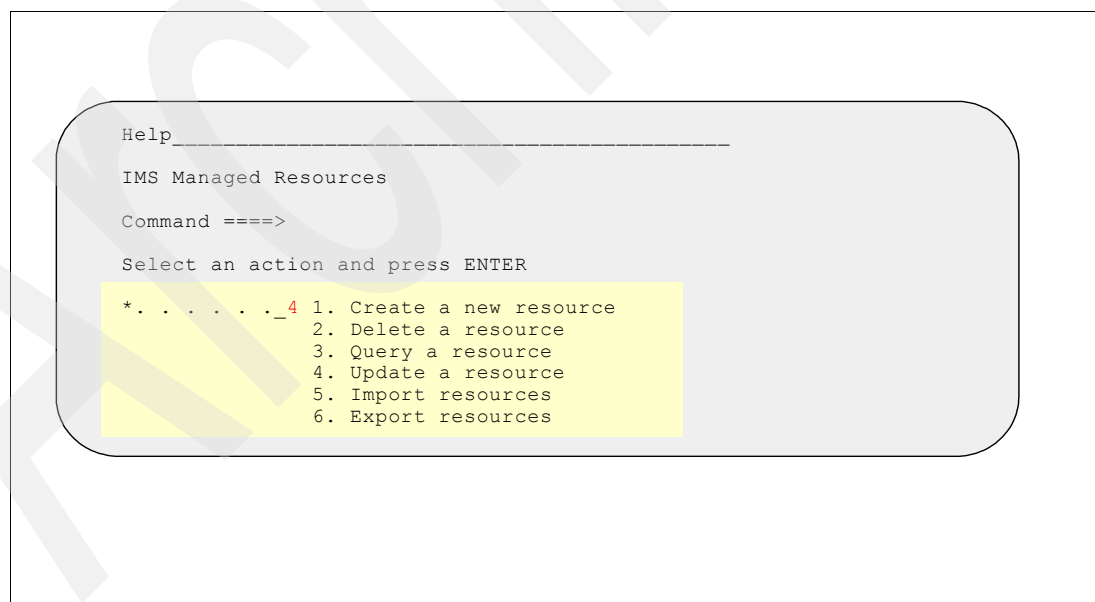


Figure 10-8 DRD Example: Update a resource

Figure 10-9 on page 283 shows the panel that allows you to select which resource is to be updated. The example on this visual shows a request to update a transaction.

```

File  Action  Manage Resources  SPOC  View  Options  Help
-----
Plex1          IMS Update a Resource
Command =====>
-----Plex. . Route. . Wait. .
-----

Select a resource type. To base the resource on a model, specify model info
Press enter to continue.

* Resource. . . . . 4 1. Database
                    2. Program
                    3. Routing Code
                    4. Transaction

Resource Name. . . . . APOL12, APOL13, SHUTTLE
Resource Type. . . . . 1. Resource
                    2. Descriptor

```

Figure 10-9 DRD Example: Update a resource: Transaction

Figure 10-10 shows the result in the panel that defines the parameter values applicable to a transaction. You can now provide the desired changes.

```

File  Action  Manage Resources  SPOC  View  Options  Help
-----
Plex1          IMS Update a Transaction
Command =====>
-----Plex. . Route. . Wait. .
-----

Press enter to continue.
More: +

*NAME Transaction Name. . . . . : APOL12, APOL13, SHUTTLE
Filter APOL12,
CLASS Current Class. . . . . 1-999

START                                     STOP
Enter "/" to select option               Enter "/" to select option
- Q Start queuing                        - Q Stop queuing
- SCHED Start scheduling                 - SCHED Stop scheduling
- SUSPEND Transfer to ready queue       - TRACE Stop trace
- TRACE Start trace

SET
AOCMD AOI command option. . . . . Yes, No, Tran
CLASS Set to class. . . . . 1-999
CMTMODE Commit mode. . . . . Single, Multiple
CONV Conversational. . . . . Yes, No
DCLWA Log Write-Ahead option. . . . . Yes, No
DEFAULT Default - descriptor only. . . . . Yes, No

```

Figure 10-10 DRD Example: Update a resource: Transaction: Changes

Chapter 6, "Dynamic Resource Definition" on page 133 in this book gives you more details about DRD.

10.5.3 Miscellaneous

This topic discusses miscellaneous changes in IVP.

IXUUPROC

The IVP variable *IXUUPROC* (User JES2 PROCLIB DDNAME or JES3 DDNAME suffix) was a required value in prior releases of IMS. In the IMS Version 10 IVP, it is optional. If the variable is not specified, the JES2 */*JOBPARM PROCLIB=xxxxxxx* JCL statement or the JES3 */*MAIN PROC=xx* is not generated in the JCL. This allows you to specify the JES PROCLIBs using the *JCLLIB ORDER* statement in one of the JESx variables: *IXUJESC1* to *IXUJESC5*.

New option to access the IVP Variable Export Utility from the IVP panel

As shown in Figure 10-11, the Variable Export Utility can be directly accessed as an option from the IVP Phase Selection panel.

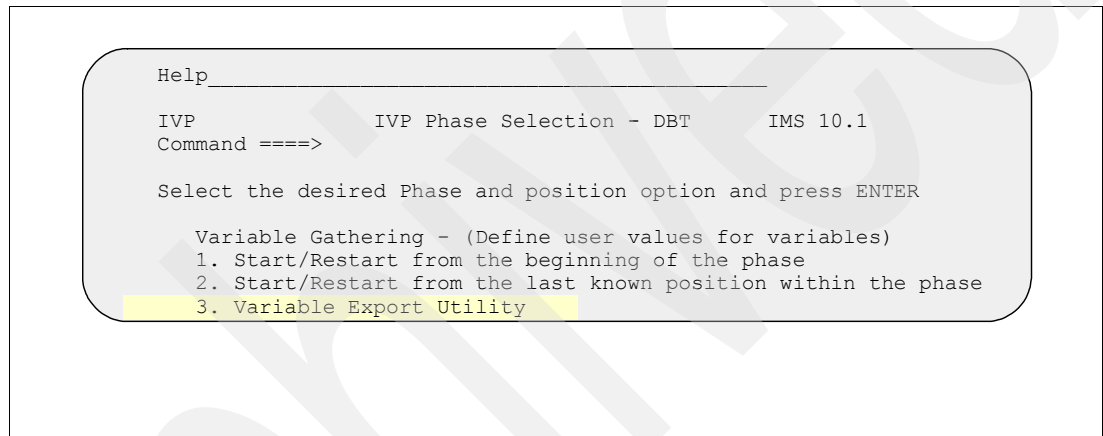


Figure 10-11 Selecting the Variable Export Utility

IV_E204J

The IMS IVP job IV_E204J has been changed to provide a DEV statement default of *DEV TYPE=(3270,2),FEAT=IGNORE* rather than *DEV TYPE=3270-A02,FEAT=IGNORE*. This is used for all terminals defined to IMS with *DEV TYPE=3270-An* and *SIZE=(x,80)* where *x* is greater than or equal to 24 (in addition to using it for terminals specifically defined as *DEV TYPE=(3270,2)*).

The change makes the IVP easier to use, because it eliminates the need for you to edit the MFS source to match the system.

SYSDA

The hard coded unit type *SYSDA* is changed to *SYSALLDA* in all the IVP JCL. *SYSDA* is not a required unit type. *SYSALLDA* is a required unit type.

10.6 System definition and execution parameter changes

This section describes changes to IMS system definition and execution parameters that affect functions available in previous releases. Definition and execution changes for new functions are discussed where the new functions are explained.

10.6.1 Transaction scheduling

In the previous releases of IMS, the *SCHD*= parameter on the system definition TRANSACT macro specified the scheduling option used when the transaction defined on the TRANSACT macro could not be scheduled for internal reasons (database intent or no more space in PSB pool or DMB pool to bring in needed blocks). IMS Version 10 always uses the *SCHD*=3 option. The default in previous releases was *SCHD*=1:

- ▶ *SCHD*=1

Specified that only transactions of equal or higher priority in the selected class would be scheduled. Five consecutive intent conflicts are allowed within a class before IMS starts scheduling the next eligible class.

- ▶ *SCHD*=3

Specified that any transaction in the selected class can be scheduled. IMS starts scheduling the next eligible class after attempting to schedule all the transactions in the current class.

Two other options were available in previous releases:

- ▶ *SCHD*=2

Specified that only higher-priority transactions in the selected class could be scheduled.

- ▶ *SCHD*=4

Specified that IMS should skip to the next class and attempt to schedule the highest-priority transaction in that class.

Scheduling failures for the “internal reasons” are rare. Intent conflicts only occur when PROCOPTs with the *E* (exclusive) option are used.

Note: The PSB and DMB pools must always be large enough to hold the currently required PSBs and DMBs.

10.6.2 FPCTRL system definition macro eliminated

IMS Version 10 has simplified the implementation of Fast Path. It does not have to be specified at system definition time. The FPCTRL system definition macro is ignored. In previous releases, it was required to enable Fast Path capabilities. It also was used to specify the default values for some Fast Path parameters. Fast Path capabilities are always generated for DB/DC and DBCTL systems. They are enabled by a parameter at execution time.

You must specify *FP*=Y at execution time for an online system to enable Fast Path capabilities in IMS Version 10. The default for the parameter is *FP*=N. Because defaults for Fast Path parameters cannot be specified at system definition time, they need to be specified at execution time. These parameters are *DBBF*=, *DBFX*=, *BSIZ*=, and *OTHR*=. The default values for these parameters are the same as the default values for the FPCTRL macro in previous releases. These are:

- ▶ *DBBF*=10
- ▶ *DBFX*=4
- ▶ *BSIZ*=2048
- ▶ *OTHR*=2

It is highly unlikely that these values are appropriate for most Fast Path users.

10.6.3 DFSDFxxx PROCLIB member

IMS Version 10 has a new PROCLIB member, which is used to consolidate execution definitions that otherwise might be in several PROCLIB members. The *DFSDFxxx* member can be used to define parameters for several components of IMS.

DFSDFxxx eliminates the need to use a *DFSSQxxx* member for Shared Queues parameters or a *DFSCGxxx* member for Common Service Layer (CSL) parameters. Instead, these parameters can be specified in the *DFSDFxxx* member. If a parameter is specified in both members, the values specified in *DFSSQxxx* and *DFSCGxxx* override those specified in *DFSDFxxx*.

Note: *DFSDFxxx* is required to specify some IMS parameters.

The *DFSDFxxx* member used by IMS is specified in the *DFSDF=* execution parameter. The *DFSDFxxx* member has up to five sections of definitions. Each section begins with a section identification statement. Dynamic resource definition parameters are defined only in *DFSDFxxx*. They are in the *DYNAMIC_RESOURCES* and *COMMON-SERVICE_LAYER* sections. Each section begins with a section header:

- ▶ <SECTION=DYNAMIC_RESOURCES>
- ▶ <SECTION=SHARED_QUEUES>
Same parameters as are used in *DFSSQxxx* member
- ▶ <SECTION=USER_EXITS>:
 - Associates an exit routine type with a list of one or more exit routine modules to be called during IMS restart
 - The modules are called in the order listed.
- ▶ <SECTION=COMMON_SERVICE_LAYER>
Same parameters as are used in *DFSCGxxx* member
- ▶ <SECTION=DIAGNOSTICS_STATISTICS>:
 - Transaction level statistics
 - Abend search and notification
 - BPE external trace

In Figure 10-12 on page 287, you see an example of a *DFSDFxxx* member. It includes parameters for DRD, traces, transaction level statistics, abend search and notification, Shared Queues, and CSL. Shared Queues and CSL parameters have the same meaning as they had in the *DFSSQxxx* and *DFSCGxxx* members for previous releases.

```

<SECTION=DYNAMIC_RESOURCES>
DCLWA=Y                                /* Log write-ahead */
RDDSDSN=(IMS10.RDDSDSN1,IMS10.RDDSDSN2,IMS10.RDDSDSN3,IMS10.RDDSDSN4),
AUTOEXPORT=AUTO,
AUTOIMPORT=AUTO,
IMPORTERR=ABORT,
RDDSERR=ABORT
<SECTION=DIAGNOSTICS_STATISTICS>
TRANSTAT=Y                            /* Transaction Level
Statistics */
IASNPROC=IMSASN                        /* Abend Search &
Notification PROC */
<SECTION=SHARED_QUEUES>
CQS=CQSCW1,                           /* CQS name */
CQSSN=CQS1,                           /* CQS subsystem name */
EMHQ=IMSEMHQ01,                       /* EMHQ structure name */
MSGQ=IMSMMSGQ01,                     /* MSGQ structure name */
SQGROUP=GRUP1                         /* XCF group name */
<SECTION=COMMON_SERVICE_LAYER>
CMDSEC=N,                             /* No cmd authorization
checking */
IMSPLEX=PLEX1,                        /* IMSplex name */
OLC=GLOBAL,                           /* GLOBAL online change */
OLCSTAT=IMSTESTS.IMS01.OLCSTAT        /* OLCSTAT data set name */
MODBLKS=DYN                           /* DRD ENABLED; OLC
DISABLED */
PLEXPARM=()                           /* GLOBAL resource status
*/

```

Figure 10-12 Example of a DFSDFxxx member

10.6.4 Enhanced display of system parameters

IMS issues message DFS1929I to display the IMS System Parameters. Because the message is issued early in the initialization process, the values displayed reflect those provided during system generation or overridden through startup parameters. Every parameter is shown. If a value has not been provided by the user, the default, if one exists, is displayed. The values shown, however, might not reflect the actual values that are used during online execution. The discrepancy between the actual and requested values can be minor, but in some situations it can be major.

IMS Version 10 continues to issue the DFS1929I message early in initialization, for compatibility, but issues it once again after restart is complete and the log has been read so that the values displayed reflect the actual values in effect for the IMS execution. The first instance of the message has “initial” in the header. For example:

```
DFS1929I * IMS SYSTEM PARAMETERS INITIAL FOR THIS V10.1 DBDC EXECUTION:*
```

The DFS1929I message has been enhanced to include the IMS Control Region type along with the IMS Version number (shown as *Vnn.n*). The remainder of the message is compatible with the format of previous releases.

The first form of the message that is printed during IMS initialization contains all the IMS parameters with user-specified or default values.

In Figure 10-13 on page 288, we show you an example of the enhancements to the message header.

```

DFS1929I * IMS SYSTEM PARAMETERS ACTIVE FOR THIS V10.1 DBDC EXECUTION:
DFS1929I *      ALOT      = 1440      *SYS3
DFS1929I *      AOIP      = 2147483647 *SYS3
DFS1929I *      AOIS      = N        *SYS3

```

Figure 10-13 Example of the DFS1929I message header

The second DFS1929I message, which is produced after the log has been read, has been enhanced to display:

- ▶ Only the parameters that are applicable to the control region type
- ▶ Only the parameters associated with active components

For example, Fast Path parameters will be shown only if activated. Prior releases always displayed the parameters regardless of whether Fast Path was enabled. The new IMS Version 10 parameter *FP=Y | N* will always be displayed.

- ▶ Actual values that will be used during execution

For example, the CPLOG value displayed in the first DFS1929I message during initialization might not be valid after a warm or emergency restart, because its value was changed by a command.

Note: As a result, the value in the first DFS1929I message does not help in diagnosing problems.

The problem encompasses other parameters, such as APPC, APPCSE, OTMA, OTMASE, and so forth, which can also be changed by command and are recoverable during an IMS restart. The actual values, therefore, are provided in the second DFS1929I message after the restart is complete.

When migrating to IMS Version 10, user-written routines or procedures that look for the DFS1929I message should recognize the header format change to include the IMS version and Control Region type.

Additionally, you need to modify these routines to recognize that there are two versions of the message.

10.7 Installation and migration tasks

This is an overview of the tasks for migration to IMS Version 10:

- ▶ Eliminate BTAM and SMU use.

Remember that IMS Version 10 is the first release of IMS not to support BTAM or the Security Management Utility (SMU). Their use should be discontinued before migrating to IMS Version 10.

- Check the Preventive Service Planning (PSP) bucket.

As with all installations of new products, you need to review the Preventive Service Planning (PSP) bucket and the Program Directory for the product before you begin the migration. The upgrade name for IMS Version 10 is IMS 1010.

- Review the Program Directory.
- Review the installation information in Chapters 1, 2, and 3 of the installation guide:

The *IMS Version 10: Installation Guide*, GC18-9710, has content similar to the *Installation Volume 1: Installation Verification manual for IMS Version 9*. You need to read this guide before beginning the migration process. Review Chapters 1, 2, and 3 and Appendix G for installation information.

- Review the z/OS interface information in Chapter 5 of the *IMS Version 10: System Administration Guide*, SC18-9718.

- Install prerequisite software and maintenance.
- Check your IMS tools and related products.
- Apply coexistence maintenance to other lower level IMS systems:

- DBRC coexistence SPEs
- System Management Enhancements coexistence SPEs
- Global Online Change coexistence SPEs
- Operations Management coexistence SPEs
- OTMA and IMS Connect coexistence SPEs
- The Resource Consistency Checking Coexistence SPE

- Evaluate and update IMS exit routines.

If you use the RECON I/O Exit Routine (DSPCEXT0), you need to examine it for required changes due to the change in RECON records. The routine needs to be changed if you implement parallel RECON access.

The Queue Space Notification Exit (DFSQSSP0) is changed in IMS 10 and has added two fields; you must evaluate whether you need to make changes.

- Install IMS Version 10 using SMP/E installation process.
- Use the Custom-Built Product Delivery Offering (CBPDO) or ServerPac if you want.

The CBPDO product package consists of one logical tape (multiple volumes). A CBPDO package that includes IMS can also include other products in the same System Release (SREL). CBPDO also provides service for the products included with the product order. The service includes all PTFs available within one week of order fulfillment. All PTFs are identified by one or more SOURCEIDs, including PUTyymm, RSUyymm, SMCREC, and SMCCOR.

ServerPac is an entitled software delivery package. It consists of products and service for which IBM has performed the SMP/E installation steps and some of the post-SMP/E installation steps. To install the package on your system and complete the installation of the software that it includes, use the CustomPac Installation Dialog, which is the same dialog used for all CustomPac offerings, including SystemPac® (dump-by-data-set format), ProductPac®, and RefreshPac. For IMS, ServerPac allocates, catalogs, and loads all the data sets; sets up the SMP/E environment; supplies a job to update PARMLIB (IEFSSNxx, PROGxx, IEASVCxx, and SCHEDxx); and directs you to start the IVP.

- Run the IVP.

Running the IVP is optional, but recommended. All required installations tasks are done outside of the IVP. The IVP verifies that the installation is correct.

- **System definition**

System definition is required as with previous IMS releases. Most system definition statements from previous IMS releases are compatible with IMS Version 10.

- **Install the Type 2 and Type 4 SVCs**

The system definition creates the Type 2 and Type 4 SVC modules which must be installed in the z/OS system. A z/OS IPL is not required. These modules can be installed by running DFSUSVC0 and specifying SVCTYPE=(2,4).

- **Review IMS Resource Cleanup Module considerations**

In IMS Version 9 and IMS Version 10, use a dynamic resource cleanup module (DFSMRCL0). No user setup is required; you do not need to install the static resource cleanup module (DFSMRCL0) on the host z/OS system for IMS Version 9 or IMS Version 10. IMS Version 8 uses the static resource cleanup module (DFSMRCL0). Do not delete it from your z/OS system while there is still a possibility that you will need to run IMS Version 8.

- **Upgrade RECONs**

Upgrade the RECONs by using the CHANGE.RECON UPGRADE command using the IMS Version 10 release of the DBRC utility. If you have an IMSplex name in the RECON, PK55384 is needed on IMS Version 10 prior to the upgrade.

- **ACBGEN**

An ACBGEN is required for use with the online system or any batch DBB jobs.

10.7.1 Remote Site Recovery (RSR) migration

The migration of systems using RSR is similar to migrations for previous IMS Versions. IMS Version 10 tracking systems can process logs produced by lower releases. The IMS Version 10 Isolated Log Sender (ILS) function of the Transport Manager System (TMS) can process logs created by lower releases. However, IMS Version 8 and IMS Version 9 tracking systems cannot accept logs produced by IMS Version 10, and the IMS Version 8 and IMS Version 9 ILSs cannot accept logs produced by IMS Version 10. Of course, you can migrate all of the RSR components at the same time. More likely, you might prefer to migrate them in stages. The tracking system must be migrated before or at the same time as the ILS at the active site. The ILS at the active site must be migrated before or at the same time as the active IMS system. The RECONs must be upgraded to IMS Version 10 before the systems that use them are migrated to IMS Version 10.

Migration steps

The migration steps are:

1. Upgrade the RSR tracking system RECONs to IMS Version 10.
2. Migrate RSR tracking system to IMS Version 10.
3. Upgrade the active system RECONs to IMS Version 10.
4. Migrate active system Transport Manager Subsystem (TMS) running Isolated Log Sender to IMS Version 10.
5. Migrate active IMS to IMS Version 10.

Tools support for IMS Version 10

In this appendix, we discuss the following topics:

- **IMS tools migration and coexistence**

IMS tools migration and coexistence shows how the IBM IMS tools support IMS Version 10. As you can see, most of the tools will support IMS Version 10 by implementing the named PTFs or by replacing the tool with a newer version/release.

Check the list in Table 10-4 on page 292 to see which tool version you need to use to run with IMS Version 10.

- **IMS tools enhancements**

IMS tools enhancements give you a list of all available IBM IMS DM tools with a brief description of the functions of these tools.

IMS tools migration and coexistence considerations

Note: There is a special consideration for IMS Queue Control Facility (QCF) users:

There are APARs for IMS Version 9 and IMS Version 8 to support fallback from IMS Version 10 to these releases. The fixes for these APARs are needed if messages created under IMS Version 10 need to be requeued to an IMS Version 9 or IMS Version 8 system after a fallback to these releases. These fixes are not required to take messages from an IMS Version 8 or IMS Version 9 system and requeue them on an IMS Version 10 system.

The following APARs for QCF are required for fallback from IMS Version 10 to IMS Version 8 or IMS Version 9 to let them recognize IMS Version 10 messages:

- ▶ PK29667 (IMS Version 9)
- ▶ PK29666 (IMS Version 8)

These APARS are not required for migration from IMS Version 8 or IMS Version 9 to IMS Version 10.

Queue Control Facility (QCF) V2.1 requires the fix for APAR P28050 for IMS Version 10 support.

Table 10-4 lists all the IMS tools and how they support IMS Version 10. The tool requirements sometimes change during the early days of an IMS release, and this table includes the latest information available when this book was written. You can find the latest information from the following Web site:

<http://www.ibm.com/support/docview.wss?rs=688&uid=swg21249278>

or

<http://tinyurl.com/3852cs>

The “tinyurl” was created to simplify the typing.

Table 10-4 IMS tools support for IMS Version 10

IMS tools	VRM	IMS Version 10 support			Comments
		No support	Tolerates	Exploits	
Application Recovery Tool	1.2.0		X		PK33728
Data Refresher	1.1.0		X		No change required
Data Encryption	1.1.0		X		No change required
DB/DC Data Dictionary	1.6.0		X		No change required
IMS ADF II	2.2.0		X		No change required
IMS Audit Management Expert	1.1.0		X		PK33279
IMS Batch Backout Manager	1.1.0		X		APAR PK32686
IMS Batch Terminal Simulator	3.1.0		X		APAR PK33113
IMS Buffer Pool Analyzer	1.1.0		X		APAR PK33270
IMS Checkpoint Wrapper	1.1.0		X		APAR PK33922

IMS tools	VRM	IMS Version 10 support			Comments
		No support	Tolerates	Exploits	
IMS Command Control Facility	2.1.0		X		APAR PK32798
IMS Connect Extensions	1.2.0		X		APAR PK33967
IMS Connect Extensions	2.1.0		X		No change required
IMS Database Control Suite	3.1.0	X			
IMS Database Control Suite	3.2.0		X		APAR PK33307
IMS Database Recovery Facility	2.1.0		X		APAR PK31981
IMS Database Recovery Facility	3.1.0		X		APAR PK32674
IMS Database Repair Facility	1.2.0		X		No change required
IMS DataPropagator™	3.1.0		X		APAR PK32840
IMS DEDB Fast Recovery	2.2.0		X		APAR PK33437
IMS ETO Support	3.1.0		X		APAR PK32876
IMS Fast Path Basic Tools	1.2.0	X			Replaced by IMS HP Fast Path Utility V3.1
IMS Fast Path Online Tools	2.1.0	X			Replaced by IMS HP Fast Path Utility V3.1
IMS HALDB Conversion and Maintenance Aid	2.1.0	X			Replaced by IMS HALDB Conversion and Maintenance Aid V3.1
IMS HALDB Conversion and Maintenance Aid	3.1.0		X		
IMS HD Compression Extended	2.2.0		X		APAR PK32180
IMS HP Change Accumulation	1.2.0	X			
IMS HP Change Accumulation	1.3.0		X		APAR PK31045
IMS HP Change Accumulation	1.4.0		X		
IMS HP Fast Path Utility	2.1.0	X			Replaced by IMS HP Fast Path Utility V3.1
IMS HP Fast Path Utility	2.2.0	X			Replaced by IMS HP Fast Path Utility V3.1
IMS HP Fast Path Utility	3.1.0		X		
IMS HP Image Copy	3.1.0	X			
IMS HP Image Copy	3.2.0	X			
IMS HP Image Copy	4.1.0		X		APAR PK33115
IMS HP Load	2.1.0		X		APAR PK33129
IMS HP Pointer Checker	2.1.0	X			
IMS HP Pointer Checker	2.2.0		X		APAR PK33114

IMS tools	VRM	IMS Version 10 support			Comments
		No support	Tolerates	Exploits	
IMS HP Prefix Resolution	3.1.0		X		APAR PK33116
IMS HP Sysgen Tools	1.1.0	X			Replaced by IMS HP Sysgen Tools V2.2 (with APAR PK34032)
IMS HP Sysgen Tools	1.2.0	X			Replaced by IMS HP Sysgen Tools V2.2 (with APAR PK34032)
IMS HP Sysgen Tools	2.1.0	X			Replaced by IMS HP Sysgen Tools V2.2 (with APAR PK34032)
IMS HP Sysgen Tools	2.2.0		X		APAR PK34032
IMS HP Unload	1.1.0		X		APAR PK33118
IMS HP Unload	2.1.0		X		No change required
IMS Index Builder	2.3.0		X		APAR PK29311
IMS Library Integrity Utilities	1.1.0		X		APAR PK33141
IMS Library Management Utilities	1.1.0	X			Replaced by IMS Library Integrity Utilities
IMS MFS Reversal Utilities	1.1.0		X		No change required
IMS Network Compression Facility	1.1.0		X		APAR PK33275
IMS Online Reorganization Facility	1.1.0	X			Replaced by IMS Online Reorganization Facility V1.2.
IMS Online Reorganization Facility	1.2.0		X		
IMS Parallel Reorganization	3.1.0	X			
IMS Parallel Reorganization	3.2.0		X		APAR PK33131
IMS Parameter Manager	1.1.0		X		APAR PK32900
IMS Performance Analyzer	4.1.0		X		APAR PK32899
IMS Problem Investigator	2.1.0		X		APAR PK32902
IMS Program Restart Facility	2.1.0		X		APAR PK32684
IMS Queue Control Facility	2.1.0		X		APAR PK28050
IMS Queue Control Facility	3.1.0		X		No change required
IMS Sequential Randomizer Generator	1.1.0		X		No change required
IMS Sysplex Manager	1.2.0		X		APARs PK31950 and PK34236

IMS tools	VRM	IMS Version 10 support			Comments
		No support	Tolerates	Exploits	
IMS Tools Knowledge Base	1.1.0		X		
IMS Workload Router	2.4.0	X			Replaced by IMS Workload Router V2.5
IMS Workload Router	2.5.0		X		

10.8 IMS tool enhancements

The IBM DB2 and IMS tools are specifically designed to enhance the performance of IMS and DB2. These tools have been upgraded and enhanced to work with IMS Version 10.

The tools are categorized by function:

- ▶ IMS database administration tools
- ▶ IMS application management tools
- ▶ IMS performance management tools
- ▶ IMS recovery management tools
- ▶ IMS information integration management tools
- ▶ IMS utilities management tools
- ▶ IMS TM management tools
- ▶ Miscellaneous IMS tools

For complete information about these tools, go to:

<http://www.ibm.com/software/data/db2imstools>

10.8.1 IMS database administration tools

The following sections describe the IBM IMS database administration tools.

IBM Data Encryption for IMS and DB2 Databases, V1.1

The IBM Data Encryption for IMS and DB2 Databases tool:

- ▶ Provides user-customizable, precoded, exit routines for encryption of IMS and DB2 data
- ▶ Conforms to the existing OS/390® and z/OS security model and is implemented using standard IMS and DB2 exits
- ▶ Exploits System z and zSeries Crypto hardware features, which results in low overhead encryption and decryption
- ▶ Uses the ANSI Data Encryption Algorithm (DEA), which is also known as the U.S. National Institute of Science and Technology (NIST) Data Encryption Standard (DES) algorithm
- ▶ Works at the IMS segment level and can be customized at that level. For DB2, encryption and decryption are customizable at the row level.

Program number: 5655-P03

IBM IMS Audit Management Expert for z/OS, V1.1

The IBM IMS Audit Management Expert for z/OS tool:

- ▶ Collects and correlates data access information from a variety of IMS resources
- ▶ Provides centralized easy-to-use support for auditing needs through collecting data and saving it into an audit repository
- ▶ Provides a user interface (UI) with different options for examining the data in the repository
- ▶ Has a comprehensive reporting facility with multiple options to view and report on collected data
- ▶ Contains a separate administration user interface to be used in configuring the data query portion of the product
- ▶ Is designed to meet the needs involved in auditing IMS environments. Auditors use IMS Audit Management Expert to view, analyze, and report on IMS data.

Program number: 5655-R23

IBM IMS Database Repair Facility for z/OS, V1.2

The IBM IMS Database Repair Facility for z/OS:

- ▶ Interactively and in batch mode repairs VSAM- and OSAM-organized IMS databases that contain pointer or data errors
- ▶ Performs repairs quickly, thereby reducing the amount of time that the affected database is taken offline
- ▶ Supports IMS Fast Path DEDB databases
- ▶ Features VSAM and OSAM pointer repair, pointer navigation, a backout safety feature, and an ISPF front end
- ▶ Runs in interactive mode, in which you can view entire blocks of data or individual IMS segments and you can navigate to other segments. Any changes made are tracked and can be undone.
- ▶ Runs in batch mode, in which you can dump blocks from the data set or submit changes to the block data

Program number: 5655-E03

IBM IMS High Availability Large Database (HALDB) Conversion and Maintenance Aid for z/OS, V3.1

The IBM IMS High Availability Large Database (HALDB) Conversion and Maintenance Aid for z/OS:

- ▶ Guides DBAs through HALDB conversions with ease
- ▶ Assists in transporting HALDB databases to IMS test systems
- ▶ Speeds the conversion process with partition modelling
- ▶ Enables conversion with minimal manual intervention
- ▶ Requires no HALDB experience and little IMS expertise
- ▶ Provides single-step conversion, which allows you to convert a database in batch mode (the database must have dynamic allocation member (MDA) descriptions)
- ▶ Allows merging of identical HALDB databases and can be used to combine “user-partitioned” databases into one multipartition HALDB

- ▶ Dynamically detects the existence of IMS HP Unload V1.1, IMS HP Load V2.1, IMS HP Image Copy V4.1, and IMS HP Pointer Checker V2.2 and utilizes them accordingly.
- ▶ Working in conjunction with IMS Online Reorganization Facility V1.2, you can perform near online conversion to HALDB databases while the non-HALDB database is still being updated and near online maintenance allowing you to split or combine partitions with a very short outage.

Program number: 5655-N46

IBM IMS Library Integrity Utilities for z/OS, V1.1

The IBM IMS Library Integrity Utilities for z/OS:

- ▶ Includes and replaces all of the features and functions of the IMS Library Management Utilities:
 - Managing IMS ACB, PSB, and DBD libraries
 - DBD/PSB/ACB Compare, DBD/PSB/ACB Reversal, and DBD/PSB/ACB Mapper and IMS Advanced ACB Generator
 - Providing a high-speed generation process that greatly reduces the time needed to process large volumes of IMS ACBs
 - Reporting information in a tabular form
 - Checking results from your desktop and ensuring that all PSBs and DBDs were processed as expected
- ▶ Includes an Integrity Checker to help prevent system outages caused by databases corrupted by using the wrong DBD
- ▶ Includes a Consistency Checker function to ensure that all the necessary definitions have been created for a database

Program number: 5655-I42

IBM IMS Online Reorganization for z/OS, V1.2

The IBM IMS Online Reorganization for z/OS tool:

- ▶ Enables reorganization of IMS full function and HALDB databases with a short outage in seconds rather than minutes or hours (a short period of time called *takeover time*)
- ▶ Reorganizes databases to shadow data sets, captures the changes during this period, and applies them to the reorganized shadow data sets
- ▶ Requires databases and their data sets to be registered with DBRC
- ▶ Reorganizes HISAM, HIDAM, HDAM, and SHISAM databases and recreates their associated index data sets
- ▶ Reorganizes PHIDAM and PHDAM databases as single partitions while making selected DBD changes during the process
- ▶ Supports batch message processing programs (BMPs) that allow you to pause BMPs during /DBRECOVERY commands
- ▶ Allows predefining the online reorganization “window” to schedule when you want your online databases reorganized
- ▶ Supports CICS and OTMA applications that quiesce database access during /DBRECOVERY commands
- ▶ Supports the reorganization of all HALDB partitions in a single job step

Program number: 5655-H97

IBM IMS Parameter Manager for z/OS, V1.1

The IBM IMS Parameter Manager for z/OS tool:

- ▶ Enables IMS administrators to control the specification and maintenance of IMS parameter library members
- ▶ Simplifies parameter management and provides increased levels of parameter control
- ▶ Supports the 19 startup parameter members in an IMS system startup
- ▶ Allows parameter specification by ISPF dialog panels or the integrated ISPF editor
- ▶ Contains extensive online help and parameter descriptions
- ▶ Provides parameter syntax and value checking, and automatic parameter statement construction
- ▶ Provides a view of active (current) parameter members by IMS system
- ▶ Enables the creation of a “backup” member immediately prior to applying changes
- ▶ Automatically collects previous version histories for altered parameter members
- ▶ Includes facilities to assist with IMS software version migration

Program number: 5655-L69

IBM IMS Sequential Randomizer Generator for OS/390, V1.1

The IBM IMS Sequential Randomizer Generator for OS/390 tool:

- ▶ Creates a randomizer that enables the user to access HDAM and DEDB database segments either directly or sequentially
- ▶ Allows access to HDAM and DEDB databases in logical key sequence without sacrificing their efficient direct access capabilities
- ▶ Optimizes the distribution of database records by adjusting the randomizing module if the number of synonyms, the control interval (CI), or block utilization exceeds the user-specified value

Program number: 5655-E11

IBM IMS Sysplex Manager for z/OS, V1.2

The IBM IMS Sysplex Manager for z/OS tool:

- ▶ Reduces operational complexity with a single-system image providing aggregate or local views of data relating to IMS operating in a Sysplex or an IMSplex, shared message queues, and data sharing
- ▶ Provides enhanced management with structured displays of IMS-related address spaces, Coupling Facility (CF) structures, and resources
- ▶ Allows quick responses to potential problems with recorded system exceptions and console alerts
- ▶ Streamlines operations using the Single Point of Control (SPOC) as well as providing local and global command support
- ▶ Provides the ability to query IMS resources across the IMSplex by status or attribute
- ▶ Contains the Enhanced Dependent Region display with more filtering criteria and the ability to issue commands

- ▶ Provides consolidated master terminal operator messages across an IMSplex in an audit log
- ▶ Contains an IMS command and response journal for audit and trouble-shooting purposes
- ▶ Includes an automatic archive function for log data

Program number: 5655-P01

IBM IMS Tools Knowledge Base for z/OS, V1.1

The IBM IMS Tools Knowledge Base for z/OS tool:

- ▶ Is a central report repository allowing access to historical report data
- ▶ Automatically captures reports to the central repository by any IMS tool that is enabled for the IMS Tools Knowledge Base environment
- ▶ Automatically retains reports with a retention policy that controls the number of days and the number of versions of a report to retain
- ▶ Provides ready access to find and view reports using an interactive interface, which provides extensive search capabilities to quickly locate any reports and display them from anywhere in the sysplex

Program number: 5655-R34

10.8.2 IMS application management tools

The following sections describe the IBM IMS application management tools.

IMS Batch Backout Manager for z/OS, V1.1

The IBM IMS Batch Backout Manager for z/OS tool:

- ▶ Helps automate the normally manual task of backing out database updates after an IMS batch application failure
- ▶ Reduces the downtime associated with an application failure and improves database availability
- ▶ Provides automation of the batch backout process after specific application abends
- ▶ Dynamically handles log close and allocation
- ▶ Forces allocation of batch SLDSs for jobs with no logs allocated (either by specific or by generic job name)

Program number: 5697-H75

IBM IMS Batch Terminal Simulator for z/OS, V3.1

The IBM IMS Batch Terminal Simulator for z/OS tool:

- ▶ Provides a comprehensive way to test and check IMS application program logic, IMS application interfaces, teleprocessing activity, 3270 format control blocks, and database activity
- ▶ Simulates the operation of IMS applications in TSO and batch environments
- ▶ Operates transparently to the applications, requiring no changes to IMS code, control blocks, libraries, or application load modules
- ▶ Accesses DL/I and DB2 databases from TSO terminals

- ▶ Provides a stable online system in test and production environments to execute applications properly before they are put online
- ▶ Provides audit reporting and Java language support

Program number: 5655-J57

IBM IMS Connect Extensions for z/OS, V2.1

The IBM IMS Connect Extensions for z/OS tool provides:

- ▶ Ability to analyze problems and optimize performance by recording key IMS Connect events
- ▶ Improve system security with flexible access control
- ▶ Simplify the deployment of WebSphere and SOAP clients
- ▶ Create detailed IMS Connect reports in conjunction with IBM IMS Performance Analyzer for z/OS
- ▶ Solve IMS Connect problems in conjunction with IBM IMS Problem Investigator for z/OS
- ▶ View graphical real-time reports of TCP/IP activity with IBM OMEGAMON® XE for IMS on z/OS

Program number: 5655-S56

IBM IMS Message Format Services Reversal Utilities for z/OS, V1.1

The IBM IMS Message Format Services Reversal Utilities for z/OS:

- ▶ Includes the MFS Reversal and MFS Compare utilities in the package
- ▶ Converts Message Format Services MID, MOD, DIF, and DOF control blocks back into Message Format Services utility control statements
- ▶ Helps you recover the source and compare deltas if you lose your MFS source library or suspect a difference between the generated control blocks and the source
- ▶ Provides useful summary reports of the IMS FORMAT library that show the relationships among the members

Program number: 5655-F45

IBM IMS Program Restart Facility for OS/390, V2.1

The IBM IMS Program Restart Facility for OS/390:

- ▶ Automatically assigns the most recent checkpoint ID to jobs that are restarted because of an earlier abnormal termination
- ▶ Forces an extended restart to occur for all jobs that require restart
- ▶ Helps prevent corruption resulting from restarting an abended job without specifying a restart checkpoint ID
- ▶ Saves costly and time-consuming database recoveries caused by manual restart errors
- ▶ Reduces the risk of data being unavailable or becoming corrupt

Program number: 5655-E14

10.8.3 IMS performance management tools

The following sections describe the IBM IMS performance management tools.

IBM IMS Buffer Pool Analyzer for z/OS, V1.1

The IBM IMS Buffer Pool Analyzer for z/OS tool:

- ▶ Provides modelling facilities to assist with making informed decisions about the addition of buffers to an existing pool or sizing requirements for a new buffer pool
- ▶ Helps you determine the impact of buffer pool changes before they are made to take the guesswork out of the process
- ▶ Analyzes IMS database buffer pools (OSAM and VSAM) to provide statistical analysis of the impact of changes that affect the buffer pools
- ▶ Provides I/O rates and buffering requirements for a specific database
- ▶ Allows for better allocation of real memory resources
- ▶ Identifies databases that most heavily use each database subpool
- ▶ Performs “what if” scenario analysis, such as identifying the impact of splitting a specific database into a new buffer pool
- ▶ Determines the performance effects for a given buffer pool when you add or reduce the number of buffer pools

Program number: 5697-H77

IBM IMS Network Compression Facility for z/OS, V1.1

The IBM IMS Network Compression Facility for z/OS:

- ▶ Provides several new options for compression of 3270 data streams:
 - Compresses all repeated characters, not just spaces, nulls, asterisks, and dashes
 - Allows 3270 Field Merge, blank elimination, and elimination of non-display fields
 - Eliminates redundant 3270 Set Buffer Address (SBA) commands
- ▶ Makes installation easier by eliminating the need for the XCM address space and subsystem and for SYS1.PARMLIB updates, except APF authorization of one data set
- ▶ Includes a utility to convert current options to the new PROCLIB member format
- ▶ Improves flexibility by allowing cross-MVS system inquiries and updates to Network Compression Facility information and options

Program number: 5655-E41

IBM IMS Performance Analyzer for z/OS, V4.1

The IBM IMS Performance Analyzer for z/OS tool:

- ▶ Provides comprehensive reporting for the IMS Log and IMS Monitor, and for IMS Connect Extensions for z/OS event collection
- ▶ End-to-end transit analysis for all transaction workloads, including Shared Queues
- ▶ End-to-end IMS Connect and IMS log reporting, revealing a transaction's life cycle through Connect and IMS
- ▶ Report forms allow you to design your own transit reports
- ▶ Provides a wide variety of reports that can help shorten transaction response times and increase resource availability
- ▶ DBRC Log selection for quick and easy log report requests
- ▶ Produces IMS Log and Monitor reports to “health check” your system
- ▶ Comprehensive IMS monitor reporting, including Fast Path

Program number: 5655-R03

IMS Problem Investigator for z/OS, V2.1

The IBM IMS Problem Investigator for z/OS tool:

- ▶ Provides an enhanced level of problem-determination services for IMS TM and IMS DB systems
- ▶ Includes navigation aids, formatted and personalized reporting, and investigative procedures for IMS log, monitor, user log, CQS, and IMS Connect records
- ▶ Offers powerful automated features to help reduce the amount of time required to identify and analyze defects or other events of interest in the IMS log, monitor, user log, CQS, and IMS Connect records
- ▶ Allows you to gain an end-to-end picture of transactions in an IMSplex

Program number: 5655-R02

IBM Tivoli OMEGAMON XE for IMS on z/OS, V3.1

The IBM Tivoli® OMEGAMON XE for IMS on z/OS tool:

- ▶ Proactively manages performance and availability of IBM IMS systems from a single, integrated interface
- ▶ Tracks and optimizes both resource usage and transaction processing
- ▶ Views Coupling Facility statistics to identify factors affecting the performance of IBM Parallel Sysplex environments
- ▶ Monitors workload balancing using Shared Queues support and data sharing to minimize the impact of locks on shared databases
- ▶ Integrates information from IBM Tivoli OMEGAMON XE monitors across multiple platforms and third-party software into a single view

Program number 5608-A39

10.8.4 IMS recovery management tools

The following sections describe the IBM IMS recovery management tools.

Application Recovery Tool for IMS and DB2 Databases, V1.2

The Application Recovery Tool for IMS and DB2 Databases:

- ▶ Includes enhancements for IMS applications
- ▶ Supports HALDB (High Availability Large Databases) by adding a HALDB (Y I N) parameter to all IMS-related functions
- ▶ Supports IMS sysplex data sharing
- ▶ Adds the new member IMSDSGS to the PARMLIB

Program number: 5697-F56

IMS Database Recovery Facility for z/OS, V3.1

The IBM IMS Database Recovery Facility for z/OS:

- ▶ Is a high performance, state of the art database recovery product supporting all recoverable IMS databases

- ▶ Provides a high performance tool for recovering IMS database data sets and Fast Path areas
- ▶ Works in an online environment, with a single pass of input data
- ▶ Speeds recovery by applying database changes to multiple data sets simultaneously
- ▶ Increases database availability by reducing recovery time
- ▶ Provides recovery to any prior point in time through a timestamp recovery function
- ▶ Creates image copy data sets during the recovery process, eliminating the need to follow recovery with a separate image copy invocation
- ▶ Creates index data sets during the recovery process including primary, secondary, and HALDB index list data sets, which eliminates the need to follow recovery with a separate index build invocation
- ▶ Creates offline, cloned copies of the database that can be used for application program testing, audit, and database recovery
- ▶ Validates data during the recovery process, eliminating the need to follow a recovery with a separate validation procedure
- ▶ Automatically deletes and creates database data sets during recovery, eliminating the need to precede recovery with a separate data set deletion and creation procedure
- ▶ Generates new image copy data sets by applying database updates from a log data set or change accumulation data set to existing image copy data
- ▶ Provides options for verifying data sets, including allocate and open to verify availability and usability of recovery input data sets
- ▶ Eliminates the need to issue the IMS /DBR DB and /DBR AREA commands before recovery and the IMS /STA DB and /STA AREA commands after recovery

Program number: 5655-N47

IBM IMS DEDB Fast Recovery for z/OS, V2.2

The IBM IMS DEDB Fast Recovery for z/OS tool:

- ▶ Assists in the operation and maintenance of data integrity of IMS databases
- ▶ Is designed as a fast alternative to emergency restart (ERE) failure recovery
- ▶ Corrects online log data sets (OLDSs) by invalidating logging for transactions that did not reach the synchronization point
- ▶ Significantly reduces the amount of time needed to recover DEDBs after an IMS failure
- ▶ Generates the JCL for the MSDB Dump Recovery utility to be processed before an IMS cold start for MSDB (main storage database) recovery
- ▶ Shortens the recovery time of an unscheduled IMS cold start while maintaining the integrity of the IMS databases
- ▶ Supports multiple IMS releases from a single Load Library

Program number: 5655-E32

IBM IMS High Performance Image Copy for z/OS, V4.1

The IBM IMS High Performance Image Copy for z/OS tool:

- ▶ Helps you process and manage image copies more efficiently

- ▶ Enables you to run the image copy function with the hash checking of the IMS High Performance Pointer Checker for z/OS tool, V2 under the control of the IMS Parallel Reorganization for z/OS tool, V3
- ▶ Reduces the run time of the reorganization process for an IMS full-function and HALDB database
- ▶ Allows accuracy checking of an image copy and supports processes in parallel within an IMS Parallel Reorganization
- ▶ Contains a new optional parameter for controlling the secondary image-copy data set
- ▶ Supports dynamic allocation for the specification of SMS classes for new output image-copy data sets
- ▶ Creation Image Copy function creates new copies from:
 - IMS standard image copy data set
 - ICE compressed image copy data set
 - IMS Online Image Copy data set
 - IMS Image Copy 2 data set
- ▶ Supports Concurrent Copy for databases residing on devices supporting this feature, providing you with point-in-time data consistency
- ▶ Supports SnapShot and FlashCopy providing you with rapid backup and rapid recovery of a database data set

Program number: 5655-N45

IBM IMS High Performance Change Accumulation for z/OS, V1.4

The IBM IMS High Performance Change Accumulation for z/OS tool:

- ▶ Merges most recent database changes quickly, safely, and accurately
- ▶ Saves database administrator resources, as well as supplementary software costs while protecting your IMS data
- ▶ Offers improved performance with single address space support and simplified setup and configuration
- ▶ Provides priority scheduling for change accumulation groups, improved elapsed time performance, and reduction or elimination of cross-memory data transfers
- ▶ Uses in-memory spill files for 64-bit addressing
- ▶ Accumulates Type 29 log records for Online Reorganization (OLR) processing
- ▶ Contains a standard procedure mode for processing JCL by using dynamic allocation and no requirement for write access to a procedure library
- ▶ Has a Started Task Control (STC) interface that merges the custom and standard procedure modes of JCL processing

Program number: 5655-F59

10.8.5 IMS information integration management tools

The following sections describe the IBM IMS Information Integration management tools.

IBM IMS DataPropagator for z/OS, V3.1

The IBM IMS DataPropagator for z/OS tool:

- ▶ Supports WebSphere MQ-based, asynchronous near real-time propagation that allows:
 - The Capture Component to capture IMS database changes performed by IMS Batch, IMS BMP, and IMS TM application programs
 - The Apply Component to transform the IMS database changes into relational format and apply them to DB2 tables on the same or on different systems from the IMS databases
 - The transmission of the database changes from the Capture to the Target Component through the recoverable, buffered, and queued asynchronous messaging services of WebSphere MQ
 - Reduces elapsed time between the IMS database updates and the DB2 table updates to as little as a few seconds
- ▶ Supports point-in-time propagation, allowing the content of the DB2 target tables to reflect clearly identified logical points in time of the IMS source databases, for example, the logical end of a business day
- ▶ Exploits the well established, high-performance, reliable queued messaging services of WebSphere MQ and allows enterprises to leverage their WebSphere MQ and MQSeries® administration and operations skills
- ▶ Improves performance by transmitting multiple IMS database changes within the same WebSphere MQ message, by compressing WebSphere MQ message data, and by exploiting the caching services of the Virtual Lookaside Facility (VLF) of z/OS

Program number: 5655-E52

IBM WebSphere Classic Data Event Publisher for z/OS, V9.1

The IBM WebSphere Classic Data Event Publisher for z/OS tool:

- ▶ Provides easy linking of data events with business processes
- ▶ Supports the capturing of changes made to IMS, VSAM, CA-IDMS, and Adabas data and the publishing of these changes to WebSphere MQ queues
- ▶ Automatically reformats captured data into a consistent relational format before packaging it into either XML or delimited value format for publishing

Program number: 5655-R54

IBM WebSphere Classic Replication Server for z/OS, V9.1

The IBM WebSphere Classic Replication Server for z/OS tool:

- ▶ Supports SQL-driven and near-real time changed-data feeds over WebSphere MQ from VSAM files, IMS, CA-IDMS, and Adabas
- ▶ Automatically reformats System z data into relational constructs before delivering it to WebSphere MQ
- ▶ Maintains transactional integrity and is fully recoverable
- ▶ Leverages a common infrastructure with WebSphere Classic Federation Server for z/OS, and WebSphere Classic Data Event Publisher for z/OS, including the Classic Data Architect GUI metadata management tool

Program number: 5655-R53

IBM WebSphere Classic Federation Server for z/OS, V9.1

The IBM WebSphere Classic Federation Server for z/OS tool:

- ▶ Provides SQL access to mainframe databases and files with transactional speed and enterprise scale without mainframe programming
- ▶ Allows applications and tools to issue SQL SELECT, INSERT, UPDATE, and DELETE commands using ODBC, JDBC, or a Command Level Interface (CLI) to access System z data
- ▶ Stores System z-accessed data in VSAM, IAM, and sequential files, as well as DB2 for z/OS, IMS, Software AG Adabas, and CA-Datcom and CA-IDMS databases all without mainframe programming

Program number 5655-R52

10.8.6 IMS utilities management tools

The following sections describe the IBM IMS utilities management tools.

These tools are grouped into the following categories:

- ▶ Fast Path utilities management
- ▶ Full function utilities management
- ▶ Administration utilities management

10.8.7 Fast Path utilities management

The following sections describe the IBM IMS utilities management tools for Fast Path databases.

IBM IMS High Performance Fast Path Utilities for z/OS, V3.1

The IBM IMS High Performance Fast Path Utilities for z/OS tool:

- ▶ Provides an efficient and easy way to create Job Control Language (JCL) programs to unload, reload, restructure, and analyze the IMS Fast Path Data Entry Database (DEDB)
- ▶ Is composed of IMS Fast Path Advanced Tool (formerly IMS High Performance Fast Path Reorganization Tool, IMS Fast Path Online Tools, and IMS Fast Path Basic Tools)
- ▶ The IMS Fast Path Advanced Tool contains:
 - New extract function, which retrieves segment data from multiple DEDB areas
 - A DMAC print utility that enables printing of the contents of the DMAC block for multiple DEDB areas
 - The ability to input or to generate image copies with the Analyze, Change, DMAC Print, Extract, Reload, and Unload functions
- ▶ Helps analyze, maintain, tune, and migrate a DEDB to one with better database definition
- ▶ Boosts system availability by performing a variety of key functions without having to take IMS databases offline
- ▶ Shortens the recovery time after an Emergency Restart (ERE) failure (DEDB Fast Recovery tool)
- ▶ Is enhanced with an Online Expert Reorganization utility that enables the reorganization of a set of units of work

Program number: 5655-R05

10.8.8 Full function utilities management

The following sections describe the IBM IMS utilities management tools for full function databases.

IBM IMS High Performance Load for z/OS, V2.1

The IBM IMS High Performance Load for z/OS tool:

- ▶ Provides a high performance database reloading capability for IMS full-function databases
- ▶ Supports IMS Parallel Reorganization for z/OS V3 capabilities, such as image copy creation during database reorganization
- ▶ Supports reorganization reload of HALDB partitions, including online-reorganization-capable HALDB partitions of IMS Version 9
- ▶ Supports various formats of the unloaded data sets
- ▶ Includes the Physical Sequence Sort for Reload (PSSR) utility
- ▶ Gives you the option to load compressed data that was previously unloaded by IBM IMS High Performance Unload for OS/390 in a compressed format
- ▶ Accepts multiple standard formats for load data sets
- ▶ Allows JCL streams from IMS High Performance Load for OS/390, V1 to run with IMS High Performance Load for z/OS, V2 without modification

Program number: 5655-M26

IBM IMS High Performance Unload for z/OS, V2.1

The IBM IMS High Performance Unload for OS/390 tool:

- ▶ Includes an enhancement to the Unload utility that provides JCL compatibility with the IMS HD Reorganization Unload utility (DFSURGU0)
- ▶ Adds a check record to an unload file, which is used by IMS High Performance Load V2 to check with DBRC for database updates done between unload and reload processing
- ▶ Creates unload files that can be used to migrate a non-HALDB database to HALDB
- ▶ Enhancements to the API that supports a Qualified Segment Search Argument (SSA) in a GN call to retrieve the root segment or the second level dependent segment
- ▶ Some of the other enhancements with IMS HP Unload V2.1 include improved documentation and performance improvements

Program number: 5655-E06

IBM IMS High Performance Prefix Resolution for z/OS, V3.1

The IBM IMS High Performance Prefix Resolution for z/OS tool:

- ▶ Enables you to resolve and update prefixes of IMS databases involved in logical relationships as a single job step
- ▶ Uses the HPPRPIPE data transfer service to eliminate the intermediate Work File 2 (WF2) and Work File 3 (WF3) data sets
- ▶ Helps you avoid much of the I/O, tape handling, and DASD requirements that are often associated with prefix resolution and prefix update
- ▶ Executes the prefix resolution and prefix update functions as replacements for the IMS Prefix Resolution and IMS Prefix Update utilities

- ▶ Supports IMS Parallel Reorganization for z/OS, V3 single job step execution of database reorganization, prefix resolution, and prefix update tasks

Program number: 5655-M27

IBM IMS High Performance Pointer Checker for z/OS, V2.2

The IBM IMS High Performance Pointer Checker for z/OS:

- ▶ Enables you to analyze corrupt databases quickly and reduce the amount of time spent to diagnose and repair them
- ▶ Generates reports that facilitate system tuning, report space utilization, and detect and report problems in primary and secondary indexes
- ▶ Provides the ability to set several new thresholds, including available extents, CA and CI splits, and database and data set last extents
- ▶ Has significant improvements in performance to Full Checking capabilities: improved usability with simplified setup and operation; parallel processing of databases; and improved reporting
- ▶ Provides ease of use and fast, easy detection of database status

Program number: 5655-K53

IBM IMS Index Builder for z/OS, V2.3

The IBM IMS Index Builder for z/OS:

- ▶ Offers several features that improve overall performance and enhance ease of use
- ▶ Builds or rebuilds primary and secondary indexes quickly
- ▶ Allows you to specify an optional output file where records that are needed for prefix resolution can be split off and written as they are read in
- ▶ Eliminates the need to image copy indexes
- ▶ Provides an ISPF front end for easy JCL generation
- ▶ Creates a separate address space for sorting index records
- ▶ Recognizes index records that have duplicate keys and writes the duplicate keys to a SYSOUT data set
- ▶ Supports building IMS HALDB primary indexes

Program number: 5655-E24

IBM IMS Parallel Reorganization for z/OS, V3.2

The IBM IMS Parallel Reorganization for z/OS tool:

- ▶ Provides infrastructure to integrate IBM IMS tools for database reorganization and to operate them in a single job step
- ▶ Improves administrative productivity for reorganizing IMS full-function databases
- ▶ Drives IMS HP Unload for z/OS, IMS HP Load for z/OS, V2, IMS Index Builder for z/OS, V2.3, IMS HP Image Copy for z/OS, V3.2, IMS HP Pointer Checker for z/OS, V2, and IMS HP Prefix Resolution for z/OS, V3
- ▶ Issues IMS commands to IMS online subsystems. The /STA DB command is used automatically after completion of the database reorganization, and the /DBD and the /DBR commands can be issued before starting the database reorganization.
- ▶ Shows the results of issued IMS commands on each subsystem in detail in a new report

- ▶ Enables the use of an unload data set as an input data set allowing you to use IMS Parallel Reorganization as a reload utility to which one or more optional functions can be added
- ▶ Executes the image copy task with optional hash pointer checking in a parallel reorganization job step, concurrently with the unload, reload, and index builder steps
- ▶ Executes the prefix resolution and update task in a parallel reorganization job step for a database that has internal logical relationships
- ▶ Supports the IMSDALIB DD statement used for specifying the library of dynamic allocation members
- ▶ Provides optional deletion of the old database data sets at the successful completion of parallel reorganization
- ▶ Supports parallel reorganization of HISAM and SHISAM databases
- ▶ Supports parallel reorganization of HALDB partitions, including online-reorganization-capable HALDB partitions of IMS Version 9
- ▶ Allows JCL streams from IMS Parallel Reorganization, V2, JCL streams to run with IMS Parallel Reorganization V3 without modification

Program number: 5655-M28

10.8.9 Administration utilities management

The following sections describe the IBM IMS utilities management tools for IMS administration.

IBM IMS Database Control Suite for z/OS, V3.2

The IBM IMS Database Control Suite for z/OS:

- ▶ Provides a day-to-day IMS database environment management tool for DBAs, system programmers, and application support specialists
- ▶ Enables you to collect and build on the data that is required to generate job control language (JCL) for database backup, recovery, and reorganization management using IMS Base Utilities and IBM High Performance Tools
- ▶ Provides significant improvements to DBDSGGRPs that include:
 - Loading pre-existing DBDSGGRPs into IMS DBCS from the RECON
 - Using your naming conventions rather than generated names
 - Creating DBDSGGRPs with a mix of HALDB and full-function databases
- ▶ Has a powerful batch collection function that enables you to enter a database description library (DBDLIB), which then triggers the tool to build the jobs for all logically related members in just one pass, saving time and steps and avoiding data redundancy errors
- ▶ Integrates IBM IMS tools and utilities that automate database backup, recovery, reorganization, and monitoring
- ▶ Facilitates database maintenance efficiency with single-step job control language (JCL) conversion
- ▶ Improves DBA productivity with an intuitive interface and ease-of-use enhancements
- ▶ Provides an ISPF interface for the IMS Library Integrity utilities
- ▶ Includes a simplified user interface; reduction and consolidation of the number of data sets, ISPF panels, and process steps; and more flexibility for local standards

- ▶ Automates tedious manual IMS maintenance functions to make database management easier
- ▶ Is a functional replacement for the IMS Database Integrity Control Facility (DBICF) tool
- ▶ Provides improved usability and productivity because the build process, during which JCL skeletons for utilities are built, is no longer required
- ▶ Provides improved product panels and panel navigation; field level help for all help panels; and fuller integration with DBRC terminology

Program number: 5655-L08

10.8.10 IMS TM management tools

The following sections describe the IBM IMS TM management tools.

IBM IMS Command Control Facility for z/OS, V2.1

The IBM IMS Command Control Facility for z/OS:

- ▶ Issues commands for DBCTL, DCCTL, or DB/DC regions
- ▶ Issues IMS commands from a batch utility or from a TSO session via an ISPF Interface or from a Callable Application Program Interface (API)
- ▶ Ensures successful processing of database START, STOP, DBR, and DBD commands
- ▶ Synchronizes online change and database commands across all regions in a sysplex using the batch interface or callable API
- ▶ Keeps members of an IMSplex synchronized by storing commands that failed due to an IMS being unavailable and issuing the stored commands at IMS startup
- ▶ Includes enhanced CCF Message Log improved problem determination and system operation whether running a single IMS or multi-IMSplex
- ▶ Contains added IMS Operations Manager command support that provides more flexibility by enabling users who do not want to use APPC to be able to use the tool
- ▶ Contains message disposition determination which enables users to improve or eliminate user-written code

Program number: 5655-R58

IBM IMS Extended Terminal Option Support for z/OS, V3.1

The IBM IMS Extended Terminal Option Support for z/OS tool:

- ▶ Provides a front end to the IMS Extended Terminal Option (ETO) feature
- ▶ Offers capabilities to help manage, implement, customize, and exploit the benefits of ETO in your systems environment to manage resources effectively
- ▶ Supports tailoring of all parts of ETO, including sign-on processing
- ▶ Lets you set global options for the entire user community and override options for specific terminals or user IDs
- ▶ Allows LTERM names that start with a numeric value
- ▶ Supports multiple versions of IMS
- ▶ Extends Shared Queues support to issue SYSTEMS level ENQ for all LTERM names that are associated with a user at sign-on

- ▶ Offers SLU type P support and SLU 1 console support that allow you to supply a variety of options on an LU-by-LU basis (for example, Logmode, Logon Description, ASOT, and ALOT)

Program number: 5655-L61

IBM IMS High Performance System Generation (SYSGEN) Tools for z/OS, V2.2

The IBM IMS High Performance System Generation (SYSGEN) Tools for z/OS:

- ▶ Offer an integrated solution that helps reduce the requirement for IMS system generations
- ▶ Provide DBAs and System Programmers with reliable, easy-to-use IMS management tools to make application resource changes faster using fewer resources
- ▶ Include IMSplex support, which provides the capability to simultaneously update multiple IMS control regions when executing a resource update list
- ▶ Provide additional options for reverse sysgen, including the capability to select either incore control blocks or MODBLKS data set resource definitions
- ▶ Contain IMS storage display and zap, which provides the capability to display and alter IMS control blocks and storage. A storage map provides z/OS virtual storage boundaries and presents utilization for common storage areas.
- ▶ Reload ACBLIB definitions for database descriptions (DBDs) and program specifications blocks (PSBs) without using online change for ACBLIB
- ▶ Generate a resource update list that will make IMS control blocks match IMS sysgen source macros

Program number: 5655-P43

IBM IMS Queue Control Facility for z/OS, V3.1

The IBM IMS Queue Control Facility (QCF) for z/OS:

- ▶ Allows different IMS Queue Control Facility functions to run on multiple central electronics complexes (CECs) within a sysplex with other IMS Queue Control Facility servers
- ▶ With a new Queue Space Utilization Notification mechanism, allows you to now define up to ten areas of the total queue space to monitor for small or large messages
- ▶ Automatically detects an IMS cold start and initiates the requeue of the messages that were in the queue before the cold start
- ▶ Automatically detects an IMS warm start and initiates the requeue of the messages or offloads the messages that were in the dead letter queue before warm start
- ▶ Automatically offloads any messages that were on the queue during message overflow
- ▶ Selects messages based on a data string for faster problem determination
- ▶ Provides a new filter for the dead letter queue created in an IMS Extended Terminal Option (ETO) environment to identify and manipulate messages marked for the dead letter queue
- ▶ Contains multiple new console commands to display the top number of destinations that are using the message queues; display the destinations using the message queues over the last number of minutes; and initiate the requeue or offload of the messages that were in the queue

Program number: 5655-N50

IBM IMS Workload Router for z/OS, V2.5

The IBM IMS Workload Router for z/OS tool:

- ▶ Works with IMS TM to provide transparent routing or balancing of a transaction workload among two or more IBM systems
- ▶ Uses IMS Multiple Systems Coupling (MSC)
- ▶ Is adaptable to a variety of system configurations

Program number: 5697-B87

10.8.11 Miscellaneous IMS Tools

The following older IMS tools also support IMS Version 10:

- ▶ IBM Debug Tool for z/OS V8.1, Program Number: 5655-S17
- ▶ IMS ADF, Version 2.2
- ▶ IMS Data Refresher, Version 1.1
- ▶ OS/VS DB/DC Data Dictionary, Version 1.6
- ▶ Tivoli Decision Support for z/OS V1.7, Program Number: 5698-A07
- ▶ Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS, V3.1, Program Number: 5655-P07
- ▶ Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS, V3.1, Program Number: 5655-P08
- ▶ Tivoli System Automation for z/OS, V3.1, Program Number: 5698-SA3

Abbreviations and acronyms

ACEE	Access Control Environment Element	ITSO	International Technical Support Organization
AGN	Application Group Name	IVP	Installation Verification Program
AIB	Application Interface Block	KBLA	Knowledge-Based Log Analysis
APF	Authorized Program Facility	J2C	J2EE Connector Architecture
APPC	Advanced Program-to-Program Communication	J2EE	JAVA 2 Platform Enterprise Edition
BPE	Base Primitive Environment	JBP	Java Batch Program
CCF	Common Connector Framework	JCA	J2EE connector architecture
CGI	Common Gateway Interface	JCL	Job Control Language
CI	control interval	JDBC	Java Database Connectivity
CICS	Customer Information Control System	JDK™	Java Development Kit
CQS	Common Queue Server	JMP	Java Message Program
CSL	Common Service Layer	JRE™	Java Runtime Environment
CSM	Complete Status Message	JSP™	Java Server Pages
CTG	CICS Transaction Gateway	JVM	Java Virtual Machine
DB2	Database 2	LAN	local area network
DBCTL	Database Control	LPAR	logical partition
DBD	Database Description	LTERM	logical terminal
DRA	Database Resource Adapter	LU	logical unit
DRD	Dynamic Resource Definition	LU2	logical unit 2
DVIPA	dynamic virtual IP addressing	MCI	message control information
EAB	Enterprise Access Builder	MFS	message format services
ECB	Event Control Block	MOD	message output descriptor
EMH	Expedited Message Handler	MPP	message processing program
EWLM	Enterprise Workload Manager	MSC	Multiple Systems Coupling
GUI	graphical user interface	MVS	Multiple Virtual System
HTML	Hyper Text Markup Language	ODBA	Open Database Access
HTTP	Hyper Text Transfer Protocol	OLDS	online log data set
IBM	International Business Machines Corporation	OM	Operations Manager
ILDS	indirect list data set	OO	object-oriented
IMS	Information Management System	OTMA	Open Transaction Manager Access
IMS TOC	IMS TCP/IP OTMA Connection	OTMA C/I	OTMA Callable Interface
IPCS	Interactive Problem Control System	PC	personal computer
IPL	Initial Program Load	PCB	program communication block
IRM	IMS Request Message	PPT	Program Properties Table
ISC	Intersystem Communication	PSB	Program Specification Block
ISPF	Interactive Systems Productivity Facility	PST	Partition Specification Table
ITOC	IMS TCP/IP OTMA Connection	RACF	Resource Access Control Facility
		RAR	resource archive
		RM	Resource Manager

RMM	Request MOD Message
RRS/MVS	Resource Recovery Services/MVS
RSM	request status message
SCI	Structured Call Interface
SDO	service data object
SGML	Standard Generalized Markup Language
SMB	scheduler message block
SMP/E	System Modification Program/Extended
SMU	Security Maintenance Utility
SNA	System Network Architecture
SOA	service-oriented architecture
SOAP	simple object access protocol
SPOC	single point of control
SSPM	Sysplex serialized program management
STSN	Set and Test Sequence Numbers
SVL	Silicon Valley Laboratories
TCB	task control block
TCO	Time-Controlled Operations
TCP/IP	Transmission Control Protocol/Internet Protocol
Tpipe	transaction pipe
TPIPE	transaction pipe
UOR	unit of recovery
W3C	World Wide Web Consortium
WAN	wide area network
WAS	WebSphere Application Server
VIPA	Virtual IP Addressing
WLM	workload manager
WSDL	Web Service Definition Language
WWW	World Wide Web
XCF	cross system Coupling Facility
XMI	XML metadata interchange
XML	Extensible Markup Language

Archived

Archived

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this IBM Redbooks publication.

IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 319. Note that some of the documents referenced here might be available in softcopy only:

- ▶ *CICS and VSAM Record Level Sharing: Implementation Guide*, SG24-4766
- ▶ *CICS and VSAM Record Level Sharing: Planning Guide*, SG24-4765
- ▶ *CICS and VSAM Record Level Sharing: Recovery Considerations*, SG24-4768
- ▶ *DFSMSHVS Application Migration Guide*, SG24-6972
- ▶ *DFSMSHVS Overview and Planning Guide*, SG24-6971
- ▶ *DFSMSHVS Presentation Guide*, SG24-6973
- ▶ *VSAM Demystified*, SG24-6105
- ▶ *Systems Programmer's Guide to: z/OS System Logger*, SG24-6898
- ▶ *Parallel Sysplex Application Considerations*, SG24-6523
- ▶ *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: Operations and Implementation*, 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
- ▶ *IMS Connectivity in an On Demand Environment: A Practical Guide to IMS Connectivity*, SG24-6794
- ▶ *IMS Performance and Tuning Guide*, SG24-7324
- ▶ *ABCs of z/OS System Programming Volume 2*, SG24-6982
- ▶ *ABCs of z/OS System Programming Volume 3*, SG24-6983
- ▶ *ABCs of z/OS System Programming Volume 5*, SG24-6985

- ▶ *ABCs of z/OS System Programming Volume 7*, G24-6987
- ▶ *ABCs of z/OS System Programming Volume 8*, SG24-6988
- ▶ *ABCs of z/OS System Programming Volume 9*, SG24-6989
- ▶ *ABCs of z/OS System Programming Volume 10*, SG24-6990
- ▶ *ABCs of z/OS System Programming Volume 11*, SG24-6327
- ▶ *z/OS Version 1 Release 3 and 4 Implementation*, SG24-6581

Other publications

These publications are also relevant as further information sources:

- ▶ *IMS Version 10: Program Directory for Information Management System Transaction and Database Servers*, I108-7540
- ▶ *IMS Version 10: Application Programming Planning Guide*, SC18-9697
- ▶ *IMS Version 10: Application Programming Guide*, SC18-9698
- ▶ *IMS Version 10: Application Programming API Reference*, SC18-9699
- ▶ *IMS Version 10: Command Reference, Volume 1*, SC18-9700
- ▶ *IMS Version 10: Command Reference, Volume 2*, SC18-9701
- ▶ *IMS Version 10: Command Reference, Volume 3*, SC18-9702
- ▶ *IMS Version 10: Communications and Connections Guide*, SC18-9703
- ▶ *IMS Version 10: Database Administration Guide*, SC18-9704
- ▶ *IMS Version 10: Database Utilities Reference*, SC18-9705
- ▶ *IMS Version 10: Diagnosis Guide*, GC18-9706
- ▶ *IMS Version 10: Diagnosis Reference*, GC18-9707
- ▶ *IMS Version 10: Exit Routine Reference*, SC18-9708
- ▶ *IMS Version 10: IMSplex Administration Guide*, SC18-9709
- ▶ *IMS Version 10: Installation Guide*, GC18-9710
- ▶ *IMS Version 10: Master Index and Glossary*, SC18-9711
- ▶ *IMS Version 10: Messages and Codes Reference, Volume 1: DFS Messages*, GC18-9712
- ▶ *IMS Version 10: Messages and Codes Reference, Volume 2: Non-DFS Messages*, GC18-9713
- ▶ *IMS Version 10: Messages and Codes Reference, Volume 3: IMS Abend Codes*, GC18-9714
- ▶ *IMS Version 10: Messages and Codes Reference, Volume 4: IMS Component Codes*, GC18-9715
- ▶ *IMS Version 10: Operations and Automation Guide*, SC18-9716
- ▶ *IMS Version 10: Release Planning Guide*, GC18-9717
- ▶ *IMS Version 10: System Administration Guide*, SC18-9718
- ▶ *IMS Version 10: System Definition Reference*, GC18-9966
- ▶ *IMS Version 10: System Programming API Reference*, SC18-9967
- ▶ *IMS Version 10: System Utilities Reference*, SC18-9968
- ▶ *IMS Version 10: System Definition Guide*, GC18-9998

- ▶ *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: Database Recovery Control (DBRC) Guide and Reference*, SC18-7818
- ▶ *IMS Version 9: IMS Connect Guide and Reference*, SC18-9287
- ▶ *IMS Version 9: Open Transaction Manager Access Guide and Reference*, SC18-7829
- ▶ *z/OS V1R9.0 Security Server RACF Security Administrator's Guide*, SA22-7683
- ▶ *z/OS V1R4.0 - V1R9.0 DFSMStvs Planning and Operating Guide*, SC26-7348
- ▶ *z/OS DFSMStvs Administration Guide*, GC26-7483
- ▶ *z/OS MVS Programming: Writing Transaction Programs for APPC/MVS*, SA22-7621
- ▶ *DFSMS Storage Administration Reference*, SC26-7402
- ▶ *DFSMSdss Storage Administration Guide*, SC35-0423
- ▶ *DFSMSdss Storage Administration Reference*, SC26-7402
- ▶ *z/OS MVS Initialization and Tuning Guide*, SA22-7591
- ▶ *z/OS V1R9.0 MVS Programming Authorized Assembler Services Guide*, SA22-7608
- ▶ *z/OS MVS Setting Up a Sysplex*, SA22-7625
- ▶ *z/OS V1R9.0 DFSMSdss Storage Administration Guide*, SC35-0423

Online resources

These Web sites and URLs are also relevant as further information sources:

- ▶ IMS Home Page:
<http://www.ibm.com/ims/>
- ▶ IBM Redbooks publications
<http://www.ibm.com/redbooks>

How to get IBM Redbooks publications

You can search for, view, or download IBM Redbooks publications, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy IBM Redbooks publications 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

Archived

Index

A

ACEE 7, 80–81, 207, 212, 219–220, 234
ACK 208–210, 219, 221–223, 237–239, 242–243
ACK/NAK 209
AGN 14–15, 17
ALOT 119, 311
ALTPCB 78, 204, 206, 238–239, 257–258
APF 228, 301
APIs 36
APPC 6–7, 23–24, 75–80, 109, 118–120, 159, 167, 237, 288, 310
APPC/IMS 76–77, 81
APPC/MVS 76–77
APPC/IOT 76
APPCPMxx 77
APPLCTN 95, 97, 134, 151
Application Recovery Tool 292, 302
APPLID 92
ARLN 163, 179, 200
ARMRST 33–34
asynchronous output 79, 206–207, 268
automatic RECON loss notification 163
autonomic computing xi, 1, 3

B

BACKUP.RECON 174
Base Primitive Environment (BPE) 9, 25
batch bailout 299
BLDL 11
BPE 9, 13, 25–27, 30, 35, 153, 180, 227, 230, 286
BPECFGxx 25
buffer invalidation 164
buffer pools 9, 61–62, 159, 161, 166, 176, 301

C

CA splits 4, 67
cache structures 160–161, 164, 170–171, 175, 179
CBPDO 289
CFRM 106, 282
CFRM policy 106, 167
CFSIZER 107
CHANGE.DB 170, 191
CHANGE.DBDS 69, 101
CHANGE.PART command 188
CHANGE.RECON 162–163, 171–172, 180, 192, 198–199, 201, 274, 290
CHKP call 247
CI splits 67, 308
clientID 237–238
COBOL 5–6, 226–230, 236–237, 248–250, 252, 254, 257–258, 269
coexistence 179, 198–199, 201, 263, 266, 268–270, 274, 289, 291–292

COMM macro 83
commit mode 145, 212, 222, 241
commit-then-send 7, 208, 211–212, 223, 230
Concurrent Copy 304
connection pooling 5, 234
contention 158–159, 172, 175–176
conversational 5, 21, 76, 208, 230–234, 241
conversational status 234
conversations 11, 18, 21, 77–78, 80
Coupling Facility 106–107, 159, 170
CPI-C 99
CPLOG 66, 288
CQS 25–26, 29–31, 33–34, 96, 112, 180, 302
CQSIPxxx 29
CQSSGxxx 30
CQSSLxxx 30
CSA 12, 62
CSL 25, 51–52, 95, 106, 111–113, 136, 139, 141–142, 162, 269, 282, 286
CSLOIxxx 29, 106–107, 120
CSLOMCMD 110
CSLRIxxx 29
CSLSIxxx 29
cursor 10, 44, 121, 126

D

Data Encryption 292, 295
Data Refresher 292, 312
data sharing 92, 159–160, 175, 187, 264, 266–267, 298, 302
DATABASE 77, 134, 151
database reorganization 49, 122, 307–308
DATASTORE statement 213, 220–221
DB/DC 18, 22–23, 281, 285, 292, 310, 312
DB2 xii, 2, 8, 96, 98, 161, 243–245, 266, 268, 295, 299, 302, 305–306, 312
DB2 stored procedures 244
DBBF 101, 285
DBCTL 18, 62, 96–98, 119, 138–139, 268, 285, 310
DBFX 285
DBRC 1, 4–5, 22, 68–69, 71, 73, 99, 101, 134, 157–164, 166, 168–174, 177–184, 186–187, 190–194, 197–201, 267, 274, 276, 289–290, 297, 301, 307, 310
 commands 170
 RECON data sets 5, 162, 179, 184
DBRC API 4–5, 179, 183–184, 186–187, 190, 192–193, 197–199
DBRC command authorization 191
DCCTL 22–23, 119, 310
deadlock 166, 168–169, 177
DEALLOC 69
decomposed storage 259
DEDB 5, 9, 62, 99–101, 113, 177, 187, 293, 296, 298, 303, 306

DEDB statement 101
 DEDBMAS statement 101
 DEFINE LOGSTREAM statement 106, 167
 DELETE 15, 62, 114, 138, 142, 145–147, 152, 184, 187, 194, 197, 306
 dependent region 81, 95, 209–210, 246, 280
 DESC 80, 140, 145–146
 DFS058I 118
 DFS066 16
 DFS067 16
 DFS3689W 16
 DFS62DTx 78
 DFSAOE00 11, 118
 DFSCCMD0 278
 DFSCGxxx 18, 29, 51, 55, 112–113, 136, 141–142, 269, 278, 286
 DFSCtSE0 20
 DFSDCxxx 20, 29, 76, 79, 207, 278
 DFSERA10 12, 108–110
 DFSERA30 12
 DFSFDRxx 10, 62, 101
 DFSILTA0 12, 22–23
 DFSINSX0 9
 DFSISIS0 279
 DFSISTS0 12, 23–24
 DFSJVMEV 280
 DFSMRC20 290
 DFSMRCL0 290
 DFSMS 4, 48, 74, 158–159, 171, 175, 265, 267
 DFSMSCE0 8, 82–83, 94
 DFSMSdss 4, 68, 70–71, 74
 DFSPBxxx 29, 119
 DFSPREC0 4, 67
 DFSPZPxx 61
 DFSRAS00 279
 DFSSCSRT 28
 DFSSQxxx 29–31, 286
 DFSUARC0 274
 DFSUDMT0 265
 DFSUOCU0 53
 DFSUOLC0 268
 DFSURGU0 307
 DFSUSVC0 290
 DFSVSMxx 101
 DFSYDRU0 204–205
 DFSYPRX0 204
 diagnosing problems 288
 DL/I 66, 173, 299
 DLIModel 6, 235, 244, 246–254, 259, 261
 DLIModel Utility 247–248, 250, 252, 254, 259
 DRA 61–62, 97
 DRA startup table 61–62
 DRU 205
 DSN 48–49, 51, 69–70, 109, 122, 137–138, 152, 173, 182, 194–195, 200, 271–274
 DSPAPI FUNC=QUERY 184, 189
 DSPAPI FUNC=STARTDBRC 184, 186
 DSPAPI FUNC=STOPDBRC 187
 DSPAPQHD 192
 DSPAPQLI 189

DSPCEXT0 180, 289
 DSPDCAX0 191
 DSPSCIX0 162, 200
 DSPURX00 5, 197–198, 274
 dynamic allocation 9, 49, 51–52, 71, 183–184, 296, 304, 309

E

EBCDIC 225
 ECSA 5, 9, 61–62, 100–101
 EJB 238–239
 EMH 7, 32–33, 99, 102–103, 264
 ESAF 98
 ETO 21, 266–267, 293, 310–311
 exit 8, 10–11, 14–15, 17–18, 20, 52, 67, 78, 82–83, 94, 107–108, 111, 114, 118, 125–126, 162–163, 180, 190, 199, 205–207, 218, 220, 225–227, 230, 255, 272, 278–279, 286, 289, 295
 EXITMBR 227

F

Fast Path 1, 5, 7, 9, 61–63, 66, 75, 98–99, 101–103, 119, 140, 144, 180, 187, 267, 285, 288, 293, 296, 301, 303, 306
 Fast Path response mode 102–103
 FDBR 9, 13, 61–63, 101, 119
 FMID 267
 FRCNRML 54

G

GENJCL 68–70, 73–74, 170, 184, 196–197, 201
 GENJCL.CA 196–197
 GENJCL.IC 73
 GENJCL.RECOV 68–70, 73, 197
 global online change 9, 51–52, 59–60, 269
 GSAM 4, 6, 8, 49–50, 243, 245–247, 250

H

HALDB 1, 4, 66–67, 113, 188, 198, 274, 293, 296–297, 302–304, 307–309
 HDAM 297–298
 HIDAM 69–70, 297
 High Availability Large Database 1, 296
 High Performance Pointer Checker 304, 308
 HISAM 297, 309
 HSRE 101
 HSSP 66, 101
 HTML 46
 HTTP 255
 HWS 26, 213, 221, 227
 HWSCFGxx 213, 220–221, 234
 HWSIMSO0 216, 225
 HWSIMSO1 216, 225
 HWSJAVA0 216, 220, 225
 HWSMPL0 220, 225
 HWSMPL1 216, 220, 225

I
 ICH408I message 19–20
 IDCAMS 138, 164, 197
 ILDS 4, 67
 ILOGREC 270
 Image Copy 2 4, 68, 70–71, 74, 265, 267, 304
 IMS xi–xiii, 1–36, 38–51, 54, 56–62, 65–67, 69, 73–88, 91–99, 101, 103, 105–106, 108–111, 113–121, 123–125, 127–128, 130, 133–150, 152–162, 166–168, 170, 172, 175, 177–178, 180, 182–183, 186–190, 192–201, 203–229, 231–252, 254–261, 263–281, 284–292, 295, 297–304, 306–308, 310, 312
 IMS ADF II 292
 IMS Advanced ACB Generator 297
 IMS Application Menu 28, 42, 48, 153
 IMS Batch Backout Manager 292, 299
 IMS Batch Terminal Simulator 292, 299
 IMS Buffer Pool Analyzer 292, 301
 IMS Checkpoint Wrapper 292
 IMS command 19, 76, 116, 210, 219
 IMS Command Control Facility 293, 310
 IMS Connect 2, 7, 25–26, 30–31, 35, 39, 77, 203–205, 207, 212–216, 218–234, 236–239, 241–243, 255–258, 268–269, 276, 281, 289, 293, 300–302
 PORTID 223
 IMS Connector for Java 38–39, 218, 225, 235, 242
 IMS Control Center 143
 IMS conversation 241
 IMS Database Control Suite 293, 309
 IMS Database Repair Facility 293, 296
 IMS DataPropagator 293, 305
 IMS DEDB Fast Recovery 303
 IMS Dump Formatter 181
 IMS Fast Path Basic Tools 306
 IMS Fast Path Online Tools 293, 306
 IMS HD Compression Extended 293
 IMS HP Change Accumulation 293
 IMS HP Image Copy 293, 297, 308
 IMS HP Load 293, 297, 308
 IMS HP Pointer Checker 293, 297, 308
 IMS HP Prefix Resolution 294, 308
 IMS HP Sysgen Tools 294
 IMS HP Unload 294, 297, 307–308
 IMS Index Builder 294, 308
 IMS Java 6, 8, 243, 245–247, 250–252, 267, 276, 280
 IMS Library Integrity Utilities 294, 297, 309
 IMS Library Management Utilities 294, 297
 IMS MFS Reversal Utilities 294
 IMS Monitor 96, 301
 IMS Network Compression Facility 294, 301
 IMS Online Reorganization Facility 294, 297
 IMS OTMA 222, 256
 IMS Parallel Reorganization 294, 304, 307–309
 IMS Parameter Manager 294, 298
 IMS Performance Analyzer 294, 300–301
 IMS Problem Investigator 294, 300, 302
 IMS Program Restart Facility 294, 300
 IMS Queue Control Facility 292, 294, 311
 IMS security 21, 80, 278
 IMS Sequential Randomizer Generator 294, 298

IMS SOAP Gateway 255, 257–258, 269
 IMS Workload Router 295, 312
 IMSInteractionSpec 218, 242–243
 imsjava.jar 260
 IMSPLEX 31, 33–34, 106–107, 121–123, 162–163, 173, 182, 199–201
 IMSplex 4, 6, 10–11, 18, 33–34, 51, 54–56, 77, 80, 92–93, 95–96, 106, 111, 113–115, 117–118, 120–123, 125, 127, 129, 141, 162–163, 199, 270, 276, 278, 290, 298–299, 302, 310–311
 INIT OLC 16
 INIT.DBDS 101
 INIT.RECON 171, 192, 199
 INITIATE 52–56, 58–60, 142
 intact storage 259
 integrated IMS Connect function 268
 intent conflict 98
 Internet xi, 3, 41, 237
 IOPCB 78–79, 216–217, 219, 269
 IPCS 25, 180
 IRLM 161, 168–169, 175, 199, 264, 266–267
 IRM header 220, 225–227, 232
 ISC 92, 267
 ISIS 15, 278
 ISPF 9–10, 28, 41–42, 44, 48, 125–127, 137–138, 153, 270–272, 274, 282, 296, 298, 308–309
 IVP 10, 137, 263, 277–278, 280–281, 284, 289

J

J2C 235–236
 J2EE 235–236, 239–240
 Java 1, 6, 8, 38–39, 231–232, 235–238, 243–247, 249–252, 259, 267, 276, 280, 300
 Java applications 236, 251–252
 Java classes 251
 Java Virtual Machine 246
 JBP 95, 243, 246, 280
 JCA 239
 JCL 9–10, 14, 22–24, 41, 43, 48–49, 51, 53, 69, 71, 73, 108, 122, 137–138, 151–152, 162, 170, 184, 196–197, 220, 228, 271–272, 274, 284, 303–304, 306–309
 JDBC 2–3, 235, 243, 245, 252, 259, 280, 306
 JMP 95, 243, 280
 JVM 246, 280

K

KBLA 9, 12–13, 22, 47–48, 263, 270–274
 KSDS 67, 69

L

LIST.RECON 69, 173–174, 187
 load balancing 82, 212–214, 219, 223
 local online change 60
 lock structures 166
 log records 12, 24, 48, 59, 62, 66, 83, 96–97, 99, 101, 107, 119–120, 138, 161, 166–167, 193, 270–271, 304
 LOGALERT 173, 201
 logical relationships 249, 307, 309

LOGR policy 106–107, 167, 171, 282
loss of connectivity 168

M

management class 71
MAXPST 101
message queue data sets 50
metadata 6, 245–252, 259, 305
MFS 6, 19, 235, 284, 294, 300
MFS Web Enablement 6
MFS Web Services 6, 235
middleware 235
migration 17, 29, 83, 94, 136, 139, 142–143, 198, 201, 218, 246, 263, 270–274, 288, 290–292, 298
MINVERS 162, 179, 181, 192–194, 197, 199, 201
MOD 165, 300
mode 3–4, 7, 19, 31–32, 67, 83–88, 91, 93–94, 99, 102–103, 140, 145, 171, 174, 198, 212, 217, 222, 241, 253, 259, 264, 296, 304
MODE=MULT 96, 145
MODSTAT 51
MPP 40, 116
MQSeries 305
MSC 1, 6, 8, 14, 17, 21–23, 63, 75, 82–83, 85–87, 91–94, 147, 265, 268, 278, 312
MSC logical links 87
MSDB 101, 134, 249, 303
MVS 25, 38, 76–77, 107, 152, 165, 167, 177, 180, 182, 192, 221, 267, 301

N

NACK 208, 210, 237
NAK 208–210, 222, 225, 231
NORSCC 18, 269, 278
NOTIFY.UIC 68

O

ODBA 11–12, 61, 95, 97, 118, 159, 167
OLCSTAT data set 9, 51, 53–54, 56, 268
OLDS 49–50, 150, 158, 193, 195
OLR 274, 304
OM 10, 26, 29, 51, 84, 96, 99, 106–112, 114, 120–121, 125, 127–130, 136, 143–144, 269, 282
OM API 10, 121, 143
on demand 39–42, 44, 235, 259
online change xi, 8–9, 14, 50, 52–54, 56–57, 59–60, 115, 134–135, 141–142, 144, 269, 310–311
Operations 10, 106, 111, 143, 269, 275, 289
Operations Manager 10, 51, 84, 106, 111, 120, 128, 136, 282, 310
OSAM 4, 18, 49–50, 96, 98, 175, 265, 296, 301
OTMA 6–7, 26, 38–39, 76–77, 86, 109, 119, 167, 203–222, 224–225, 232, 236–240, 242–243, 256–258, 268, 281, 288–289, 297
OTMA client 6–7, 207, 209, 211, 219, 268
OTMA header 216–217
OTMASE 288
OUTBND 78–80

P

Parallel 4, 157–162, 164, 166, 171–172, 174, 177–179, 182, 199, 266–267, 294, 304, 307–309
parallel scheduling 96
Parallel Sysplex 2, 94, 159, 161, 179, 266, 302
PCB 7, 61, 78, 204, 242–243, 245–246, 248–249, 251–254, 259, 268
performance xi, 1–5, 7–8, 12, 22, 27, 36–38, 67, 82, 85, 87, 91, 94, 96, 174, 176, 212, 219, 238, 240, 295, 300, 302, 304–305, 307–308
 DBRC 4, 174
 objectives 37
persistent socket 218, 230–231, 242
PHDAM 297
PHIDAM 297
PL/I 6, 236–237
port 223
portid 223
PQ80264 101
PREFLIST 106
PRILOG 193, 195
PROCLIB 9–11, 18, 25, 27–35, 44, 51, 55, 62, 76–77, 79, 97, 101, 106–107, 109, 112–113, 120, 136, 139–143, 149, 167, 204–208, 210, 227, 278, 280–282, 284, 286, 301
PROCOPT 98
ProductPac 289
PROGxx 289
project xii, 248
PSB pool 98, 285
PSBGEN 52
PST 193

Q

QRY IMSPLEX 122
QRY MEMBER 118
QRY TRAN 97, 109–110, 121, 130
QSAM 4, 265
QUERY 8, 10, 54–56, 58–59, 87, 90, 97, 99–100, 115–118, 126, 145, 148, 157, 183–184, 186, 188–189, 191, 220–221, 234
QUERY DB 100, 115, 148
QUERY MEMBER 55–56, 59, 117
QUERY OLC 58
QUERY TRAN 97, 115–116

R

RACF 8, 11, 14–19, 42, 80, 114, 162, 164, 171, 180, 190–191, 197, 206–207, 212, 219–221, 238, 240, 267, 278–279
Rational Application Developer 236–237, 248
RECON 4, 22, 69, 137, 157–164, 166, 168, 170–172, 174, 176, 178–179, 182–183, 187, 192–193, 195, 197–200, 263, 266–267, 274, 289–290, 309
RECON data sets 5, 164, 198
RECONs 5, 69, 71, 74, 157–164, 166, 168, 170–172, 174, 177–179, 183–184, 187–188, 192–194, 197–201, 274, 290
Redbooks Web site 319

Contact us xiv
 RESLIB 166, 272
 Resource xi, 2, 5–6, 8, 11, 14, 29–30, 33–34, 38–39, 55, 57, 59, 61–62, 95–97, 114, 126–127, 133, 135, 140, 142, 144–145, 149, 156, 159, 190, 218, 220, 225, 230–232, 235–247, 259–261, 269–270, 282, 290
 resource xi, 1, 6, 10, 14–18, 20, 26, 37, 48, 50, 53, 55, 59–61, 83, 95, 114–116, 125, 133–151, 153–156, 159, 167, 171, 190–191, 207, 236, 238, 241, 269–270, 273, 278–279, 282–283, 286, 290, 301–302, 311
 Resource Manager 11, 51, 95, 167
 resource structure 51, 95, 113
 response time
 I/O 36
 RESUME TPIPE 206, 214, 216
 REXX API 128
 REXX SPOC API 111, 128, 130, 143
 RM 11, 26, 29, 33–34, 51, 95–96, 109, 113, 269
 RMF 175
 RRS 109, 159–160, 167, 171, 177, 179–180, 210, 218, 221
 RSR 62, 171, 196, 266–267, 270, 290
 RTCODE 134, 151

S

SAF 5–6, 14–17, 19, 22, 190–191, 197, 206, 220, 278
 scheduling 9, 11, 80, 94–96, 98, 146, 239, 257–258, 285, 304
 SCHEDxx 289
 SCI 26, 29, 51, 96, 111, 136, 162, 172, 179, 199–200
 SDFSEEXEC 28, 42, 48
 SDFSRESL 23–24, 48, 83, 109, 122, 201, 207, 265, 271–274, 279
 SDFSSMPL 83
 security 1, 6–7, 11, 14–19, 21, 80, 114, 157, 164, 190, 204, 206–207, 211–212, 218–219, 222, 234, 263, 267, 278–279, 295, 300
 SECURITY macro 17, 207, 278
 send-then-commit 7, 208–211, 221, 230
 ServerPac 289
 shared queue structures 96
 Shared Queues 21, 29–31, 85, 92–96, 219, 224, 232, 268
 shared queues 9, 11–12, 22–23, 60, 95–96, 98, 119, 211–214, 217, 268, 286, 301, 310
 Shared Virtual Storage Option 62
 shared VSO 99–100
 SIGNON=ALL 278
 SIZALERT 173, 201
 SMB 134
 SMF 19–20, 166
 SMP/E 265, 268, 289
 SMS storage class 71
 SMS-managed 71
 SMU 8, 11, 14–18, 263, 278, 288
 SOAP 5–6, 218, 225, 235, 254–258, 269, 300
 SOAP Gateway 225, 254–258, 269
 sockets 5, 230–231, 237
 SPA 241
 SPE 17, 198–199, 201, 212, 274, 278, 289

SPOC 9–10, 55, 58–59, 99, 106, 108, 111–112, 120–130, 143, 153, 282, 298
 SQL 251, 305–306
 state data 209–210, 213
 Statistical Analysis Utility 24
 storage 2, 5, 9, 12–13, 25, 49, 61–62, 66, 70–71, 73–74, 101, 151–152, 164, 170, 176, 185, 204, 208, 211, 219, 253, 259, 261, 265, 269, 303, 311
 storage class 71, 164, 170, 176
 stored procedure 8, 244
 Structured Call Interface 51, 136
 SUBSYS record 186
 super member 213, 215
 SVC 99 49, 183
 SVSO 5, 9, 62
 sync point 243, 246
 synchronization level 241
 syncpoint 209–210, 218
 Syntax Checker 10, 13, 28–30, 32, 34–35, 44
 SYS1.PARMLIB 162, 164–165, 177, 301
 sysplex 5, 11, 16, 167, 174, 213, 223, 234, 299, 302, 310–311
 Sysplex Distributor 212–213, 224, 232
 system definition 8, 13–14, 17, 81–84, 86, 97, 134, 278, 284–285, 290
 system generation 119, 133–134, 145, 287
 SystemPac 289

T

takeover 297
 TCP/IP 41, 243, 255, 300
 TERMINATE 55–56
 TIMEOUT 76, 209–210, 218
 timeout 6, 76, 130, 170, 207, 210, 218, 221, 223
 TMEMBER 7, 204–206, 208–211, 213, 216–219, 221–222
 TPIPE 7, 204–207, 209, 211, 213–218, 222, 224–225, 238–239, 243, 256–258
 TRANSACT 9, 60, 95, 97, 134, 151, 278, 285
 transaction code 220, 226
 TRCLEV 25–26
 TSO 9–10, 28, 41, 44, 49, 55, 58–59, 99, 108, 111, 120, 122–123, 125–127, 143, 153, 282, 299, 310
 TSO SPOC 10, 108, 111, 120, 122–123, 125–127, 153
 tuning 128, 308
 type-1 command 113
 type-2 command 52, 63, 113, 121

U

UOR 169
 UPD TRAN 97
 UPDATE 5, 25–27, 62–63, 84, 86–87, 93–94, 97, 99–100, 113–114, 142, 145, 147–148, 197, 199, 221, 306
 UPDATE DB 99–100
 UPDATE TRAN 97
 user data 38, 226
 user exit 10, 20, 111, 126, 191, 206–207, 230, 279
 user exits 127, 281
 user message exit 225, 227

V

VAUT 119
VIEWDS 220–221
VIEWHWS 220–221, 234
virtualization xi, 3, 37
VLF 305
VSAM 4, 18, 67, 69, 96, 98, 158–161, 164, 166,
168–169, 174–175, 177, 179, 192, 197, 265, 278, 296,
301, 305–306
VSO 5, 99–100, 264
VSPEC 119
VTAM 6, 82, 84–86, 92, 94, 265

W

WADS 119–120
Web Services 5–6, 235, 256
WebSphere xii, 2, 5–6, 38, 204, 212, 226, 228, 230,
232–233, 236, 239, 241, 243, 248, 255, 268, 300, 305
WebSphere Application Server 6, 38, 230, 236, 240
WebSphere MQ 305
WebSphere Studio 236
WLM 36, 38–39, 240
workload balancing 1, 302
Workload Manager 9, 36–38
WSDL file 255
WTOR 134, 169

X

XCF 31, 109, 208, 214
XMI descriptions 249, 252
XML 1, 3, 5–6, 128, 130–131, 205, 219, 225–230,
234–235, 247, 249, 251–261, 269, 305
XML data 249, 258
XML schema 228, 249, 252, 254, 259
XML storage 261
XQuery 1, 3, 6, 247, 260–261
XRF 5, 13, 62, 99, 138, 199, 281

Z

z/OS xii, 2, 4, 6, 10, 14, 18, 26, 29, 36, 38, 41, 44, 48, 61,
67, 74, 76–77, 80, 106–107, 127, 159–161, 165, 167, 171,
177–178, 192, 207, 232, 236, 240, 243, 246, 248, 261,
264, 266–267, 269, 277, 289–290, 295–303, 305,
307–308, 310–312
z/OS 1.7 267



IBM IMS Version 10 Implementation Guide: A Technical Overview

(0.5" spine)
0.475" <-> 0.873"
250 <-> 459 pages



IBM IMS Version 10 Implementation Guide

A Technical Overview



**Get familiar with the
Dynamic Resource
Definition feature**

**See the possibilities
of XQuery and other
integration
enhancements**

**Explore all the new
features of IMS
Version 10**

This IBM Redbooks publication provides an overview of the new functions and enhancements in IBM Information Management System (IMS) Version 10. IBM continues to enhance IMS integration, manageability, and scalability.

IMS helps you with On Demand Business enablement, growth, availability, and systems management that current and newer environments and cost measures require. IMS Version 10 helps in addressing your On Demand Business needs through integration/openness, manageability, and scalability, providing:

- ▶ Integration with other products and platforms across the Internet, supporting open standards that benefit you, and taking best advantage of the latest industry tooling for application development and connectivity
- ▶ Manageability in staging users to autonomic computing, easing installation and use, eliminating and reducing outages, and minimizing the education curve for users of IMS
- ▶ Scalability with virtualization in assuring flexibility for growth and expansion in a heterogeneous environment while utilizing the latest hardware and software facilities to optimize performance, capacity, availability, and recovery

This book is intended for individuals who are migrating their IMS systems to IMS Version 10 and provides the essential necessary information.

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