Database Recovery Control (DBRC) Guide and Reference

Version 9
Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 565.

Third Edition (December 2006)

This edition replaces or makes obsolete the previous edition, SC18-7818-01. The technical changes for this version are summarized under "Summary of Changes" on page xiii.

© Copyright International Business Machines Corporation 1974, 2006. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Figures ........................................ v
Tables ........................................ vii

About This Book. .............................. ix
Prerequisite Knowledge ....................... ix
How to Use This Book ....................... ix
Related Reading ............................... x
IBM Product Names Used in This Information x
How to Send Your Comments .............. xi

Summary of Changes ......................... xiii
Changes to the Current Edition of This Book for Version 9 .................. xiii
Changes to This Book for IMS V9 .......... xiii
Library Changes for IMS V9 ................ xiv

Part 1. How to Use DBRC ................. 1

Chapter 1. Overview of DBRC ............ 3
What Is DBRC? ............................... 3
DBRC Tasks ................................ 4
DBRC Components .......................... 4
When Should You Use DBRC? .............. 5
Communicating with DBRC ................ 6
DBRC Functions ............................. 7
Defining DBRC to IMS ...................... 19
Registering Databases and Database Data Sets . 20
Planning for Recovery ..................... 21

Chapter 2. Database Recovery Control Utility (DSPURX00) .................. 27
What Is the Database Recovery Control Utility (DSPURX00)? ............ 27
Input and Output for the Database Recovery Control Utility .......... 27
Invoking the Database Recovery Control Utility from Your Program . 28
Example of Database Recovery Control Utility JCL .................. 30

Chapter 3. Considerations for a DBRC System ................................. 33
General Considerations for Using DBRC .................. 33
Data Set Naming Conventions ................ 33
Database Backup Copies .................... 35
Log Record Change Accumulation .......... 43
DBDS Group Considerations ............... 48
DB Groups .................................. 49
DBRC Considerations for HALDB Online Reorganization ............... 49

Chapter 4. Initializing and Maintaining the RECON Data Set .......... 55
Planning Considerations for the RECON Data Set .................. 55
Initially Accessing the RECON Data Set ................ 62
Records in the RECON Data Set ............... 64
Maintaining the RECON Data Set ............ 73
Tracking Changes Made to the RECON Data Set ................ 81

Chapter 5. Skeletal JCL ......................... 83
Generating JCL and User-Defined Output ................ 83
Using IBM-Supplied Skeletal JCL ............ 84
Writing Your Own Skeletal JCL ............... 85
Contents of the Skeletal JCL Data Set ........ 85
Skeletal JCL Syntax .......................... 86

Chapter 6. Hints and Tips for DBRC ....... 105
Changing the RECON Data Set to Output Time Stamps in Local Time of Origin .... 105
Locating the Last SLDs Stop Time in RECON ................ 106
Adjusting GENLX when It Is Reached or It Is Too High ................ 107
Getting PRILOG Compression to Work ........ 108
Getting Notified When the PRILOG Record Exceeds the Maximum Record Size ..... 109
Closing an Open Online PRILOG ................ 109
Deleting Log Records ........................ 110
Working with Subsystem Records (SYS) .... 111
Removing Authorization Inconsistency Between the SYS from DB/AREA Records . 112
Getting Change Accumulation to Start Processing Logs Again .......... 112
Getting Change Accumulation Working When It States Nothing to Process ... 112
Moving Log Data Sets ........................ 113
Cataloging Data Sets ......................... 113
Performing Multiple Cold Starts in a Test Environment ................ 114
Avoiding Some Causes of RECON Data Set Enqueue Problems .......... 115

Part 2. DBRC Command Reference .... 117

Chapter 7. DBRC Commands ................. 119
DBRC Batch Commands ...................... 119
DBRC Online Commands ..................... 120
DBRC Command Syntax ..................... 120

Chapter 8. BACKUP Command ............. 131
BACKUPRECON ............................ 131

Chapter 9. CHANGE Commands ........... 133
CHANGE Commands In This Chapter .......... 133

Chapter 10. DELETE Commands ............. 201
DELETE Commands In This Chapter .......... 201

© Copyright IBM Corp. 1974, 2006
### Figures

1. Logs Produced for Recovery and Restart ........................................ 7
2. IMS Database Recovery process ...................................................... 10
3. How DBRC Works With the Database 
   Recovery Utility ........................................................................... 12
4. DBRC’s Role in Utility Execution .................................................... 15
5. Database Recovery Control Utility Input and Output ....................... 28
6. Registers and Operand List Contents 
   To Invoke the Database Recovery Control Utility .......................... 30
7. Inputs and Outputs of the Database Recovery Control Utility .......... 31
8. What Change Accumulation Does with Redo 
   Data from Divergent Data Streams ............................................... 45
9. DBRC Dual RECON Set Up with a Spare ................................. 55
10. Major RECON Record Types and the 
    Relationships Between Them ..................................................... 64
11. DBRC DL/I Records ................................................................... 68
12. Record Structure for a HALDB in the RECON 
    Data Set ................................................................................. 69
13. DBRC Fast Path Database Records ................................................ 70
14. Skeletal JCL Data Set Contents ...................................................... 86
15. General Format of Query Output .................................................... 358
16. DSECT of DSPAPQHD - QUERY Output 
    Block Header ............................................................................ 358
17. Format for QUERY TYPE=BACKOUT Output ................................. 361
18. DSECT of DSPAPQBO ................................................................. 362
19. Format of QUERY TYPE=DB (Full Function) 
    Output...................................................................................... 368
20. DSECT of DSPAPQDB ................................................................. 369
21. DSECT of DSPAPQSL ................................................................. 370
22. Format of QUERY TYPE=DB (Fast Path 
    DEDB) Output ........................................................................... 371
23. DSECT of DSPAPQFD ................................................................. 372
24. DSECT of DSPAPQAR ................................................................. 373
25. DSECT of DSPAPQEL ................................................................. 375
26. Format of QUERY TYPE=DB (HALDB Master 
    and Partitions) Output ................................................................ 375
27. DSECT of DSPAPQHB ................................................................. 376
28. DSECT of DSPAPQHP ................................................................. 377
29. Format of QUERY TYPE=DB (DBDS) Output ................................. 379
30. DSECT of DSPAPQDS ................................................................. 380
31. Format of QUERY TYPE=DB (RCVINFO) 
    Output...................................................................................... 381
32. DSECT of DSPAPQRI ................................................................. 382
33. DSECT of DSPAPQAL ................................................................. 382
34. DSECT of DSPAPQIC ................................................................. 383
35. DSECT of DSPAPQRV ................................................................. 384
36. DSECT of DSPAPQRR ................................................................. 385
37. Format of QUERY TYPE=DB (Database Not 
    Found) Output ........................................................................... 385
38. DSECT of DSPAPQNF ................................................................. 386
39. Format for QUERY TYPE=DBDSGROUP, 
    DBGROUP, RECOVGROUP Output .............................................. 389
40. DSECT of DSPAPQDG ................................................................. 390
41. Format for QUERY TYPE=CAGROUP Output ................................. 391
42. DSECT of DSPAPQCG ................................................................. 392
43. DSECT of DSPAPQCA ................................................................. 393
44. Format for QUERY TYPE=GSGROUP Output ................................. 394
45. DSECT of DSPAPQG ................................................................. 395
46. Format for QUERY TYPE=LOG Log Information Output .................. 400
47. DSECT of DSPAPQL ................................................................. 400
48. Format for QUERY TYPE=LOG Output for 
    PRILOG, SECLOG, PRISLDS, SECSLDS, 
    PRTSLDS, and SECTSLDS .......................................................... 401
49. DSECT of DSPAPQLG ................................................................. 401
50. Format for QUERY TYPE=LOG Output for LOGALL ....................... 403
51. DSECT of DSPAPQLA ................................................................. 403
52. Format for QUERY TYPE=OLDS Output ......................................... 407
53. DSECT of DSPAPQOL ................................................................. 408
54. Format for QUERY TYPE=RECON Output ....................................... 411
55. DSECT of DSPAPQRC ................................................................. 412
56. Format for QUERY TYPE=SUBSYS Output ...................................... 417
57. DSECT of DSPAPQSS ................................................................. 417
58. IBM-Supplied Skeletal JCL for the Log 
    Archive Utility ........................................................................... 441
59. IBM-Supplied Skeletal JCL for the Database 
    Change Accumulation Utility ...................................................... 447
60. IBM-Supplied Skeletal JCL for the Log 
    Recovery Utility ........................................................................... 450
61. IBM-Supplied Skeletal JCL for the Database 
    Image Copy Utility and the Database Image 
    Copy 2 Utility ........................................................................... 453
62. IBM-Supplied Skeletal JCL for the Online 
    Database Image Copy Utility ...................................................... 457
63. IBM-Supplied Skeletal JCL for the Database 
    Recovery Utility - Tracking Site ................................................... 460
64. IBM-Supplied Skeletal JCL for the Database 
    Recovery Utility ........................................................................... 462
65. IBM-Supplied Skeletal JCL for the HALDB 
    Index/ILDS Rebuild Utility ........................................................... 467
66. Sample LIST.HISTORY Output ..................................................... 471
67. Sample Listing of a RECON at the Active Site 
    - RECON Status ........................................................................ 476
68. Sample Listing of a RECON at the Active Site 
    - Log Records ........................................................................... 477
69. Sample Listing of a RECON at the Active Site 
    - GSG Record ........................................................................... 486
70. Sample Listing of a RECON at the Active Site 
    - SSYS Record ........................................................................... 487
71. Sample Listing of a RECON at the Active Site 
    - BACKOUT Record .................................................................. 487
72. Sample Listing of a RECON at the Active Site 
    - CAGRP and CA Records .......................................................... 488
73. Sample Listing of a RECON at the Active Site 
    - DBGRP, DBDSGRP, and RECOVGRP 
    Records ...................................................................................... 489
74. Sample Listing of a RECON at the Active Site 
    - DB (IMS) and Related Records .................................................. 490

© Copyright IBM Corp. 1974, 2006
| 75. Sample Listing of a RECON at the Active Site | 81. Sample Listing of a RECON at the Tracking Site - BACKOUT Record  | 512 |
| 76. Sample Listing of a RECON at the Active Site | 82. Sample Listing of a RECON at the Tracking Site - CAGR and CA Records  | 513 |
| 77. Sample Listing of a RECON at the Tracking Site | 83. Sample Listing of a RECON at the Tracking Site - DBDGRP Records  | 514 |
| 78. Sample Listing of a RECON at the Tracking Site | 84. Sample Listing of a RECON at the Tracking Site - DB (IMS) and Related Records  | 515 |
| 79. Sample Listing of a RECON at the Tracking Site | 85. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records  | 517 |
| 80. Sample Listing of a RECON at the Tracking Site | 86. Sample Listing of a RECON at the Tracking Site - DB (FP) and Related Records  | 528 |
### Tables

1. Licensed Program Full Names and Short Names | x
2. Results of GENJCLIC Processing When GENMAX and RECOVPFD are Specified with REUSE | 40
3. Results of GENJCLIC Processing When GENMAX and RECOVPFD are Specified with NOREUSE | 41
4. DSID DCB Numbers for the A-J, L, and X and the M-V and Y Data Sets | 51
5. Data Sharing Between the Releases | 52
6. Determining Which RECON Data Sets Are Accessed | 63
7. GENJCL Commands and What They Generate | 83
8. Records That Can Be Selected Using the %SELECT Keyword | 90
9. Parameter Definitions for HALDB Partition and Master | 142
10. CHANGE.DBDS for HALDBs | 150
11. Supported dbname specifications for HALDBs on CHANGE.DBDSGRP command parameters | 158
12. Service Level IDs and Associated Required Maintenance | 179
13. Supported dbname specifications for HALDBs on INIT.DBDSGRP command parameters | 273
14. Parameters of NOTIFY.PRILOG (for OLDS) Command for Open, Switch, and Close | 314
15. Parameters of NOTIFY.PRILOG (for RLDS) Command for Open, EOV, and Close | 318
16. Parameters of NOTIFY.PRILOG (SLDS or TSLDS) Command for Open, EOV, and Close | 323
17. Parameters of NOTIFY.SECLOG (for OLDS) Command for Open, Switch, and Close | 332
18. Parameters of NOTIFY.SECLOG (for RLDS) Command for Open, EOV, and Close | 335
19. Parameters of NOTIFY.SECLOG (for SLDS or TSLDS) Command for Open, EOV, and Close | 339
20. Return and Reason Codes for TYPE=BACKOUT Query Requests | 360
21. Return and Reason Codes for TYPE=DB Query Requests | 366
22. Return and Reason Codes for TYPE=xxxxGROUP Query Requests | 388
23. Return and Reason Codes for TYPE=LOG Query Requests | 398
24. Return and Reason Codes for TYPE=OLDS Query Requests | 405
25. Return and Reason Codes for TYPE=RECON Queries | 410
26. Return and Reason Codes for TYPE=SUBSYS Query Requests | 415
27. Return and Reason Codes for RELBUF | 420
28. Return and Reason Codes for the STARTDBRC Request | 424
29. Return and Reason Codes for the STOPDBRC Request | 428
30. Symbolic Keywords for All Supported Utilities | 431
31. Symbolic Keywords for Log Archive Utility | 432
32. Symbolic Keywords for Database Change Accumulation Utility | 433
33. Symbolic Keywords for Log Recovery Utility | 434
34. Symbolic Keywords for Database Image Copy Utilities | 434
35. Symbolic Keywords for Database Recovery (Receive) Utility | 436
36. Symbolic Keywords for Database Recovery Utility - Recovery (RECOVJCL) | 438
37. Fields Present in the RECON Record | 529
38. Fields Present in a Log Record | 533
39. Fields Present in the LOGALL Record | 535
40. Fields Present in an Online Log Record | 536
41. Fields Present in a CSG Record | 537
42. Fields Present in a SSYS Record | 537
43. Fields Present in a BACKOUT Record | 539
44. Fields Present in a CAGRP Record | 539
45. Fields Present in a CA Record | 540
46. Fields Present in the DBDSGRP Record | 542
47. Fields Present in the DB (IMS) Record | 542
48. Fields Present in the DB (HALDB) Record | 544
49. Fields Present in the DB (PART) Record | 545
50. Fields Present in the DB (Fast Path) Record | 549
51. Fields Present in the DBDS (non-Fast Path) Record | 550
52. Fields Present in the DBDS (Fast Path) Record | 552
53. Fields Present in the ALLOC Record | 555
54. Fields Present in the IMAGE Record | 556
55. Fields Present in the REORG Record | 558
56. Fields Present in the RECOV Record | 559
57. Resource Names for Command Authorization | 561
About This Book

This information is available as part of the Information Management Software for z/OS® Solutions Information Center. To view the information within the Information Management Software for z/OS Solutions Information Center, go to publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp. This information is also available in PDF and BookManager® formats. To get the most current versions of the PDF and BookManager formats, go to the IMS™ Library page at www.ibm.com/software/data/ims/library.html.

This book describes the administrative and operational tasks associated with the IMS Database Recovery Control facility (DBRC). DBRC provides tools for tracking information used during database recovery. This book is for system and database administrators who are responsible for design, operation, and recovery procedures for installation.

With IMS Version 9, you can reorganize HALDB partitions online, either by using the integrated HALDB Online Reorganization function or by using an external product. In this information, the term HALDB Online Reorganization refers to the integrated HALDB Online Reorganization function that is part of IMS Version 9, unless otherwise indicated.

Prerequisite Knowledge

IBM® offers a wide variety of classroom and self-study courses to help you learn IMS. For a complete list of courses available, go to the Education link off of the IMS home page on the Web: http://www.ibm.com/ims.

Before using this book, you should understand:

• Basic IMS concepts
• The IMS environment
• Your installation’s IMS system
• Administration of the IMS system and databases
• z/OS control programs
• VSAM file structure and VSAM Access Method Services (AMS)

How to Use This Book

This book is structured as both a guide and reference.

• Read [Part 1, “How to Use DBRC,” on page 1] to familiarize yourself with how to use DBRC and the DBRC utilities.
• Use [Part 2, “DBRC Command Reference,” on page 117] to find information regarding the syntax and explanations associated with the DBRC commands.
• Read [Part 3, “DBRC Application Programming Interface (API) Reference,” on page 347] to find information regarding the usage, syntax, and explanations associated with the DBRC API requests.
• Use the Appendixes for additional reference information.
Related Reading

The following books in the IMS library contain information related to DBRC.

- For definitions of terminology used in this manual and references to related information in other manuals, see the [IMS V9: Master Index and Glossary]
- For more information on upgrading the RECON data set from previous releases of IMS, see the [IMS V9: Release Planning Guide]
- For installation and initialization topics, see the:
  - [IMS V9: Installation Volume 1: Installation Verification]
  - [IMS V9: Installation Volume 2: System Definition and Tailoring]
- For information on database and system administration, see the:
  - [IMS V9: Administration Guide: Database Manager]
  - [IMS V9: Administration Guide: System]
- For information on operating procedures, see the [IMS V9: Operations Guide]
- For information on system utilities, see the [IMS V9: Utilities Reference: System]
- For information on the utilities used in database recovery, see the [IMS V9: Utilities Reference: Database and Transaction Manager]
- For information on the RECON I/O exit routine, see the [IMS V9: Customization Guide]
- For diagnostic information and messages, see the:
  - [IMS V9: Diagnosis Guide and Reference]
  - [IMS V9: Messages and Codes, Volume 1] and [IMS V9: Messages and Codes, Volume 2]

IBM Product Names Used in This Information

In this information, the licensed programs shown in Table 1 are referred to by their short names.

<table>
<thead>
<tr>
<th>Licensed program full name</th>
<th>Licensed program short name</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Application Recovery Tool for IMS and DB2®</td>
<td>Application Recovery Tool</td>
</tr>
<tr>
<td>IBM CICS® Transaction Server for OS/390®</td>
<td>CICS</td>
</tr>
<tr>
<td>IBM CICS Transaction Server for z/OS</td>
<td>CICS</td>
</tr>
<tr>
<td>IBM DB2 Universal Database™</td>
<td>DB2 Universal Database</td>
</tr>
<tr>
<td>IBM DB2 Universal Database for z/OS</td>
<td>DB2 UDB for z/OS</td>
</tr>
<tr>
<td>IBM Enterprise COBOL for z/OS and OS/390</td>
<td>Enterprise COBOL</td>
</tr>
<tr>
<td>IBM Enterprise PL/I for z/OS and OS/390</td>
<td>Enterprise PL/I</td>
</tr>
<tr>
<td>IBM High Level Assembler for MVS™ &amp; VM &amp; VSE</td>
<td>High Level Assembler</td>
</tr>
<tr>
<td>IBM IMS Advanced ACB Generator</td>
<td>IMS Advanced ACB Generator</td>
</tr>
<tr>
<td>IBM IMS Batch Backout Manager</td>
<td>IMS Batch Backout Manager</td>
</tr>
<tr>
<td>IBM IMS Batch Terminal Simulator</td>
<td>IMS Batch Terminal Simulator</td>
</tr>
<tr>
<td>IBM IMS Buffer Pool Analyzer</td>
<td>IMS Buffer Pool Analyzer</td>
</tr>
<tr>
<td>IBM IMS Command Control Facility for z/OS</td>
<td>IMS Command Control Facility</td>
</tr>
</tbody>
</table>
Table 1. Licensed Program Full Names and Short Names (continued)

<table>
<thead>
<tr>
<th>Licensed program full name</th>
<th>Licensed program short name</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM IMS Connect for z/OS</td>
<td>IMS Connect</td>
</tr>
<tr>
<td>IBM IMS Connector for Java™</td>
<td>IMS Connector for Java</td>
</tr>
<tr>
<td>IBM IMS Database Control Suite</td>
<td>IMS Database Control Suite</td>
</tr>
<tr>
<td>IBM IMS Database Recovery Facility for z/OS</td>
<td>IMS Database Recovery Facility</td>
</tr>
<tr>
<td>IBM IMS Database Repair Facility</td>
<td>IMS Database Repair Facility</td>
</tr>
<tr>
<td>IBM IMS DataPropagator™ for z/OS</td>
<td>IMS DataPropagator</td>
</tr>
<tr>
<td>IBM IMS DEDB Fast Recovery</td>
<td>IMS DEDB Fast Recovery</td>
</tr>
<tr>
<td>IBM IMS Extended Terminal Option Support</td>
<td>IMS ETO Support</td>
</tr>
<tr>
<td>IBM IMS Fast Path Basic Tools</td>
<td>IMS Fast Path Basic Tools</td>
</tr>
<tr>
<td>IBM IMS Fast Path Online Tools</td>
<td>IMS Fast Path Online Tools</td>
</tr>
<tr>
<td>IBM IMS Hardware Data Compression-Extended</td>
<td>IMS Hardware Data Compression-Extended</td>
</tr>
<tr>
<td>IBM IMS High Availability Large Database (HALDB) Conversion Aid for z/OS</td>
<td>IBM IMS HALDB Conversion Aid</td>
</tr>
<tr>
<td>IBM IMS High Performance Change Accumulation Utility for z/OS</td>
<td>IMS High Performance Change Accumulation Utility</td>
</tr>
<tr>
<td>IBM IMS High Performance Load for z/OS</td>
<td>IMS HP Load</td>
</tr>
<tr>
<td>IBM IMS High Performance Pointer Checker for OS/390</td>
<td>IMS HP Pointer Checker</td>
</tr>
<tr>
<td>IBM IMS High Performance Prefix Resolution for z/OS</td>
<td>IMS HP Prefix Resolution</td>
</tr>
<tr>
<td>IBM z/OS Language Environment</td>
<td>Language Environment</td>
</tr>
<tr>
<td>IBM Tivoli® NetView® for z/OS</td>
<td>Tivoli NetView for z/OS</td>
</tr>
<tr>
<td>IBM WebSphere® Application Server for z/OS and OS/390</td>
<td>WebSphere Application Server for z/OS</td>
</tr>
<tr>
<td>IBM WebSphere MQ for z/OS</td>
<td>WebSphere MQ</td>
</tr>
<tr>
<td>IBM WebSphere Studio Application Developer Integration Edition</td>
<td>WebSphere Studio</td>
</tr>
<tr>
<td>IBM z/OS</td>
<td>z/OS</td>
</tr>
<tr>
<td>IBM z/OS C/C++</td>
<td>C/C++</td>
</tr>
</tbody>
</table>

How to Send Your Comments

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

- Click the Feedback link located at the bottom of every page in the Information Management Software for z/OS Solutions Information Center. The information center can be found at http://publib.boulder.ibm.com/infocenter/imzic.
- Go to the IMS Library page at www.ibm.com/software/data/ims/library.html and click the Library Feedback link, where you can enter and submit comments.
- Send your comments by e-mail to imspubs@us.ibm.com. Be sure to include the title, the part number of the title, the version of IMS, and, if applicable, the
specific location of the text on which you are commenting (for example, a page number in the PDF or a heading in the Information Center).
Summary of Changes

Changes to the Current Edition of This Book for Version 9

This edition includes technical and editorial changes.

Changes to This Book for IMS V9

New information on the following enhancements is included:

- **Command Authorization for /RM Commands** (see "DBRC Online Commands" on page 120)
- Eliminate the DSPLDRC Part from DBRC (PRISLD / SECSLD / PRITSLDS / SECTSLDS now mapped by DSPLOGRC, see the RECON data set record DSECTs listed in Appendix B, "Sample Listings from the RECON Data Set," on page 469)
- Fast Path Multiple Areas Shared VSO Structures (see "CHANGE.DBDS" on page 149, "INIT.DBDS" on page 266, and "DB (FP) and Related Records" on page 503)
- Sysgen Conditional Bind Removal (see "Defining DBRC to IMS" on page 19)
- Recommendations about "RECON Data Set Serialization Strategies" on page 58
- Chapter 1 (DBRC in the IMS Database Recovery Process) was completely rewritten and renamed to Chapter 1, "Overview of DBRC," on page 3.
- Some reorganization has occurred:
  - Some sections from Chapter 3, "Considerations for a DBRC System," on page 33 were moved and merged into Chapter 1, "Overview of DBRC," on page 3.
  - Chapter 2, "Database Recovery Control Utility (DSPURX00)," on page 27 was moved to from Chapter 4 to being Chapter 2.
  - Appendix D (Invoking the Database Recovery Utility from your Program) was merged into Chapter 2, "Database Recovery Control Utility (DSPURX00)," on page 27.
- A terminology change has occurred. All instances of “simple keyword” have been changed to “symbolic keyword”.
- HALDB Online Reorganization. See:
  - "DBRC Considerations for HALDB Online Reorganization" on page 49
  - "CHANGE.DB" on page 141
  - "CHANGE.RECON" on page 177
  - "GENJCL.IC" on page 229
  - "GENJCL.OIC" on page 238
  - "INIT.DB" on page 263
  - "NOTIFY.IC" on page 310
  - "NOTIFY.REORG" on page 328
  - "NOTIFY.UIC" on page 342
  - "Sample Listing of a RECON at the Active Site" on page 475
- DBRC Application Programming Interface (API). See:
  - "DBRC Tasks" on page 4
  - "Communicating with DBRC" on page 6
  - Chapter 16, "Using the DBRC API," on page 349
Library Changes for IMS V9

Changes to the IMS Library for IMS V9 include the addition of one title, a change of one title, organizational changes, and a major terminology change. Changes are indicated by a vertical bar (|) to the left of the changed text.

The IMS V9 information is now available in the Information Management Software for z/OS Solutions Information Center, which is available at http://publib.boulder.ibm.com/infocenter/imzic. The Information Management Software for z/OS Solutions Information Center provides a graphical user interface for centralized access to the product information for IMS, IMS Tools, DB2 Universal Database (UDB) for z/OS, DB2 Tools, and DB2 Query Management Facility (QMF™).

New and Revised Titles

The following list details the major changes to the IMS V9 library:

- **IMS V9: IMS Connect Guide and Reference**

  The library includes new information: *IMS V9: IMS Connect Guide and Reference*. This information is available in softcopy format only, as part of the Information Management Software for z/OS Solutions Information Center, and in PDF and BookManager formats.

  IMS Version 9 provides an integrated IMS Connect function, which offers a functional replacement for the IMS Connect tool (program number 5655-K52). In this information, the term IMS Connect refers to the integrated IMS Connect function that is part of IMS Version 9, unless otherwise indicated.

- **The information formerly titled IMS Version 8: IMS Java User’s Guide is now titled IMS V9: IMS Java Guide and Reference.** This information is available in softcopy format only, as part of the Information Management Software for z/OS Solutions Information Center, and in PDF and BookManager formats.


Organizational Changes

Organization changes to the IMS V9 library include changes to:

- **IMS V9: Customization Guide**
- **IMS V9: IMS Java Guide and Reference**
- **IMS V9: Messages and Codes, Volume 1**
- **IMS V9: Utilities Reference: System**

A new appendix has been added to the **IMS V9: Customization Guide** that describes the contents of the ADFSSMPL (also known as SDFSSMPL) data set.

The chapter titled "DLIModel Utility" has moved from **IMS V9: IMS Java Guide and Reference** to **IMS V9: Utilities Reference: System**.
The DLIModel utility messages that were in IMS V9: IMS Java Guide and Reference have moved to IMS V9: Messages and Codes, Volume 1.

Terminology Changes
IMS V9 introduces new terminology for IMS commands:

**type-1 command**
A command, generally preceded by a leading slash character, that can be entered from any valid IMS command source. In IMS Version 8, these commands were called classic commands.

**type-2 command**
A command that is entered only through the OM API. Type-2 commands are more flexible than type-2 commands and can have a broader scope. In IMS Version 8, these commands were called IMSplex commands or enhanced commands.

Accessibility features for IMS
Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

**Accessibility features**
The following list includes the major accessibility features in IMS. These features support:
- Keyboard-only operation.
- Interfaces that are commonly used by screen readers.

**Note:** The Information Management Software for z/OS Solutions Information Center, which is available at http://publib.boulder.ibm.com/infocenter/imzic, and its related publications are accessibility-enabled. You can operate all features using the keyboard instead of the mouse.

**Keyboard navigation**
You can access the information center and IMS ISPF panel functions by using a keyboard or keyboard shortcut keys.

You can find information about navigating the information center using a keyboard in the information center home at publib.boulder.ibm.com/infocenter/imzic.

For information about navigating the IMS ISPF panels using TSO/E or ISPF, refer to the z/OS V1R1.0 TSO/E Primer, the z/OS V1R5.0 TSO/E User’s Guide, and the z/OS V1R5.0 ISPF User’s Guide, Volume 1. These guides describe how to navigate each interface, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

**IBM and accessibility**
See the IBM Human Ability and Accessibility Center at www.ibm.com/able for more information about the commitment that IBM has to accessibility.
Part 1. How to Use DBRC

Chapter 1. Overview of DBRC
What Is DBRC? ........................................... 3
DBRC Tasks ............................................. 4
DBRC Components .................................... 4
RECON Data Set ....................................... 4
Database Recovery Control Utility (DSPURX00) ........... 4
Skeletal JCL ............................................ 5
When Should You Use DBRC? ........................ 5
Communicating with DBRC ............................ 6
DBRC Commands ....................................... 6
DBRC API Requests .................................... 6
DBRC Functions ....................................... 7
  Recording and Controlling Log Information ............. 7
  Recovering Databases ................................ 10
  Recording Information About Opening and Updating Databases ........................................... 16
  Supporting Data Sharing ............................... 16
  Supporting Remote Site Recovery ...................... 18
  Supporting IMSplexes ................................ 18
Defining DBRC to IMS ................................ 19
  The IMSCTRL Macro .................................. 19
  IMS.PROCLIB Execution-Parameter Members ........ 19
  IMS Procedures and DBRC .............................. 19
  DBRC Procedure ..................................... 20
Registrationing the RECON Data Set ..................... 20
Planning for Recovery ................................ 21
  Setting Up Recovery Mechanisms ..................... 21
  Recovery Facilities .................................. 22
  Recovery without DBRC ............................... 24
  Restart After an IMS Failure .......................... 24
  Restart After a DBRC Failure ......................... 25
  Recovery Involving IRLM Configurations ............. 25
  Batch Backout ....................................... 25

Chapter 2. Database Recovery Control Utility
(DSPURX00). ............................................. 27
What Is the Database Recovery Control Utility
(DSPURX00)? ......................................... 27
Input and Output for the Database Recovery
Control Utility ........................................ 27
Invoking the Database Recovery Control Utility
from Your Program .................................... 28
Example of Database Recovery Control Utility JCL .... 30

Chapter 3. Considerations for a DBRC System ......... 33
General Considerations for Using DBRC ................. 33
Data Set Naming Conventions ........................ 33
  Naming Convention for Image Copy Data Sets ....... 34
  Naming Convention for Duplicate Image Copy
  Data Sets ........................................... 34
  Naming Convention for Change Accumulation
  Data Sets ........................................... 34
Database Backup Copies ................................ 35
  The Image Copy Utilities (DFSUDMP0,
  DFSUDMT0, DFSUICP0) ............................ 35
  Concurrent Image Copy ................................ 38
  Creating Image Copy Data Sets for Future Use
  and Reuse ........................................... 39
  Controlling the Number of Image Copies .............. 39
  Managed .............................................. 39
  Recovery Period of Image Copy Data Sets and
  GENMAX .............................................. 40
  Reusing Image Copy Data Sets ....................... 41
  HSSP Image Copy ..................................... 42
  HISAM Copies (DFSURUL0 and DFSURRL0) .......... 42
  Nonstandard Image Copy Data Sets .................. 43
  Frequency and Retention of Backup Copies .......... 43
Log Record Change Accumulation ........................ 43
  Condensing the Accumulated SLDS or RLDS
  (Change Accumulation) .............................. 44
  When Is Change Accumulation Required? .............. 45
DBDS Group Considerations ........................... 48
DB Groups ............................................. 49
DBRC Considerations for HALDB Online
  Reorganization ....................................... 49
  GENJCL.IC and GENJCL.OIC Command
  Considerations ....................................... 50
  NOTIFY.IC and NOTIFY.UIC Command
  Considerations ....................................... 50
  Records Used in HALDB Online Reorganization .... 50
  RSR Tracking System ................................ 52
  Maintaining Recovery-Related Records ............... 52
  The Online Reorganization Process ................. 53

Chapter 4. Initializing and Maintaining the
RECON Data Set ....................................... 55
Planning Considerations for the RECON Data Set .... 55
  Initializing the RECON Data Set for the First
  Time .................................................. 56
  Avoiding RECON Data Set Contention Problems .... 56
Initially Accessing the RECON Data Set ............... 62
  Records in the RECON Data Set ..................... 64
  RECON Header Records ............................. 64
  Log Data Set Records ............................... 65
  Database Recovery Records ........................ 66
Maintaining the RECON Data Set ........................ 73
  Backing Up the RECON Data Set .................... 73
  Deleting Unnecessary RECON Log Records .......... 73
  Reorganizing the RECON Data Set .................. 75
  Replacing Damaged RECON Data Sets ............... 76
  Recovering the RECON Data Set .................... 76
  Replacing a Discarded RECON Data Set ............. 78
  RECON Loss Notification ........................... 78
  Preserving the Integrity of the RECON Data Set
  with DBRC Command Authorization Support ......... 79
  Tracking Changes Made to the RECON Data Set ... 81

Chapter 5. Skeletal JCL ................................ 83
  Generating JCL and User-Defined Output .......... 83
  Using IBM-Supplied Skeletal JCL ................... 84
  Writing Your Own Skeletal JCL ..................... 85
Contents of the Skeletal JCL Data Set .
Skeletal JCL Syntax . . . . . . .
Symbolic Keywords . . . . .
Using Control Keywords . . . .
Writing Control Keywords . . .
Understanding Skeletal JCL Default

. . .
. . .
. . .
. . .
. . .
Members

.
.
.
.
.

. 85
. 86
. 86
. 88
. 94
103

Chapter 6. Hints and Tips for DBRC . . . . . 105
Changing the RECON Data Set to Output Time
Stamps in Local Time of Origin . . . . . . . 105
Locating the Last SLDS Stop Time in RECON . . 106
Adjusting GENMAX When It Is Reached or It Is
Too High . . . . . . . . . . . . . . . 107
Solutions When GENMAX is Too Low . . . . 107
Solution When GENMAX is Too High . . . . 108
Getting PRILOG Compression to Work . . . . . 108
Getting Notified When the PRILOG Record
Exceeds the Maximum Record Size . . . . . . 109
Closing an Open Online PRILOG . . . . . . . 109
Deleting Log Records . . . . . . . . . . . 110
Working with Subsystem Records (SSYS) . . . . 111
Naming Conventions for SSIDs in RECON
Subsystem Records . . . . . . . . . . 111
Batch Backout . . . . . . . . . . . . 111
Deleting a Subsystem Record . . . . . . . 111
Subsystem Record Size . . . . . . . . . 112
Removing Authorization Inconsistency Between the
SSYS from DB/AREA Records . . . . . . . . 112
Getting Change Accumulation to Start Processing
Logs Again . . . . . . . . . . . . . . 112
Getting Change Accumulation Working When It
States Nothing to Process . . . . . . . . . 112
Moving Log Data Sets . . . . . . . . . . 113
Cataloging Data Sets . . . . . . . . . . . 113
Performing Multiple Cold Starts in a Test
Environment . . . . . . . . . . . . . 114
Avoiding Some Causes of RECON Data Set
Enqueue Problems . . . . . . . . . . . . 115
In a Shared DASD Environment . . . . . . 115
In a Non-Shared DASD Environment . . . . 116

2

DBRC Guide and Reference


Chapter 1. Overview of DBRC

This chapter provides an overview of the IMS Database Recovery Control facility (DBRC) and a discussion of the interactions between IMS and DBRC.

In This Chapter:
- “What Is DBRC?”
- “DBRC Tasks” on page 4
- “DBRC Components” on page 4
- “When Should You Use DBRC?” on page 5
- “Communicating with DBRC” on page 6
- “DBRC Functions” on page 7
- “Defining DBRC to IMS” on page 19
- “Registering Databases and Database Data Sets” on page 20
- “Planning for Recovery” on page 21

What Is DBRC?

DBRC is an integral part of IMS. IMS relies on DBRC to:
- Record and manage information about many items. DBRC keeps this information in a set of VSAM data sets that are collectively called the REcovery CONtrol (RECON) data set.
- Advise IMS (based on the information in the RECON data set) about how to proceed for certain IMS actions.

Specifically, DBRC:
- Helps you ensure IMS system and database integrity by recording and managing information associated with the logging process.
- Assists IMS in the restart process by notifying IMS which logs to use for restart.
- Assists IMS to allow or prevent access to databases in data-sharing environments by recording and managing database authorization information.
- Facilitates database and log recovery by:
  - Controlling the use and integrity of the information in the logs.
  - Recording and maintaining information about the databases and logs in the RECON data set.
  - Generating and verifying the Job Control Language (JCL) for various IMS utility programs.
- Supports Extended Recovery Facility (XRF) by identifying (in the RECON data set) if the subsystem is XRF capable.
- Supports Remote Site Recovery (RSR) by containing the RSR complex definition in the RECON data set and providing other services associated with controlling RSR.
- Supports IMSplexes by notifying all DBRCs in the same IMSplex when one of the DBRCs performs a RECON data set reconfiguration.
DBRC Tasks

DBRC automatically performs the following tasks:
- Records and manages information about logs for IMS
- Records recovery information in the RECON data set
- Verifies that database utilities have the correct input
- Controls the recovery of databases that are registered with DBRC
- Controls the access authorization information for the control and serialization of shared databases

You can perform the following tasks and more when you initiate them by passing commands or requests to DBRC:
- Start or stop the DBRC application programming interface (API)
- Record recovery information in the RECON data set
- Generate JCL for various IMS utilities and generate user-defined output
- List general information in the RECON data set
- Gather specific information from the RECON data set

Related Reading:
- See “Recording and Controlling Log Information” on page 7 for additional information about DBRC’s logging support.
- See “Supporting Data Sharing” on page 16 for additional information about DBRC’s data sharing support.

DBRC Components

DBRC includes the following components, which are introduced in this section:
- “RECON Data Set”
- “Database Recovery Control Utility (DSPURX00)”
- “Skeletal JCL” on page 5

RECON Data Set

DBRC stores recovery-related information in the RECON data sets.

Recommendation: Define three VSAM KSDSs for the RECON data sets when you install DBRC. The first two data sets are active data sets; the third one is a spare. The second active data set is a copy of the first. For most purposes, you can think of the two active RECON data sets as if they were a single data set, the RECON data set.

Related Reading: These data sets are described in detail in Chapter 4, “Initializing and Maintaining the RECON Data Set,” on page 55.

Database Recovery Control Utility (DSPURX00)

Use the Database Recovery Control utility to issue DBRC commands.

The DBRC commands allow you to perform all of the following tasks:
DBRC overview

- List the information in the RECON data set
- Update the information in the RECON data set
- Use the information in the RECON data set to generate jobs for the IMS utilities

**Related Reading:** The details for using the Database Recovery Control utility can be found in Chapter 2, “Database Recovery Control Utility (DSPURX00),” on page 27.

**Skeletal JCL**

DBRC uses partitioned data set (PDS) members as input models (or templates) for generating input for some of the recovery utilities. These PDS members are distributed with IMS and are called skeletal JCL.

DBRC uses the skeletal JCL, information from the RECON data set, and instructions from a *GENJCL* command to generate the JCL and control statements that are needed to correctly run some of the recovery utilities. Modify the skeletal JCL to reflect your installation’s system configuration.

**Related Reading:** For more information about skeletal JCL, see Chapter 5, “Skeletal JCL,” on page 83.

IMS provides PDS members that contain skeletal JCL statements. These PDS members are called skeletal JCL execution members and can be found in the IMS.SDFSISRC target library.

The IMS Installation Verification Program (IVP), customizes the skeletal JCL execution members and places the customized members into the IMS.PROCLIB procedure library. DBRC uses the members (from IMS.PROCLIB) to generate jobs (JCL and control statements) for the IMS utilities listed in Table 7 on page 83. There is also a skeletal JCL execution member, JOBJCL, that produces a JOB statement.

**When Should You Use DBRC?**

Most IMS configurations require DBRC, including:
- Online configurations: DB/DC, DCCTL, or DBCTL
- Data-sharing environments, including IMSplexes
- Configurations that use Extended Recovery Facility (XRF)
- Remote Site Recovery (RSR)

DBRC also plays a key role in managing the log data needed to restart and recover IMS online subsystems.

**Attention:** DBRC is not required for IMS batch jobs and for some offline utilities. However, if batch jobs and utilities that access registered databases are allowed to run without DBRC, the recoverability and integrity of the databases could be lost. Even if your configuration does not require the use of DBRC (such as in a non-data sharing, non-RSR batch environment), you can simplify your recovery process by using DBRC to supervise recovery and protect your databases.

**Related Reading:**
- [IMS V9: Operations Guide](#) provides detailed descriptions of recovery procedures with and without DBRC.
DBRC overview

- "How DBRC Helps in Recovery" on page 12 further describes how DBRC helps with database recovery.

### Communicating with DBRC

Use **DBRC commands** or **DBRC API requests** to obtain services from DBRC. "**DBRC Commands**" and "**DBRC API Requests**" discuss these two topics in more detail.

#### DBRC Commands

Use DBRC batch and online commands to:

- Add to, change, and delete information in the RECON data sets.
- Generate the JCL and the control statements necessary to run the various IMS utilities used in database recovery.

The following is a list of the DBRC batch commands:

- BACKUP.RECON
- CHANGE
- DELETE
- GENJCL
- INIT
- LIST
- NOTIFY
- RESET.GSG

Use the Database Recovery Control utility (DSPURX00) to issue DBRC batch commands. This utility can process these commands while running either in a batch environment or as a TSO foreground program.

A variation of some of the DBRC batch commands can be issued online using the /RMxxxxxx command (for example, /RMCHANGE DBRC='RECON'). Although most of the command examples in this book feature the DBRC batch commands, the online version (where appropriate) of the example command can be used instead.

**Related Reading:** For more information about the DBRC commands, see Chapter 7, "DBRC Commands," on page 119.

#### DBRC API Requests

Use DBRC API requests to:

- Start and stop DBRC.
- Query specific information from the RECON data set.

The following is a list of the DBRC API requests:

- STARTDBRC
- QUERY
- RELBUF
- STOPDBRC

**Related Reading:** For more information about the DBRC application programming interface and the API requests, see Chapter 16, "Using the DBRC API," on page 349.
DBRC Functions

The following sections describe DBRC’s major functions:
- “Recording and Controlling Log Information”
- “Recovering Databases” on page 10
- “Recording Information About Opening and Updating Databases” on page 16
- “Supporting Data Sharing” on page 16
- “Supporting Remote Site Recovery” on page 18
- “Supporting IMSplexes” on page 18

Recording and Controlling Log Information

As IMS operates, it constantly records its activities in the IMS logs. The IMS logs contain information about:
- IMS startups and shutdowns
- Changes made to databases
- Transaction requests received and responses sent
- Application program initializations and terminations
- Application program checkpoints
- IMS system checkpoints

The IMS logs consist of the:
- Write-ahead data set (WADS)
- Online log data set (OLDS)
- System log data set (SLDS)
- Recovery log data set (RLDS)

Figure 1 shows the log data sets and restart data set (RDS) that IMS produces and the RECON data set that DBRC creates and maintains.

Figure 1. Logs Produced for Recovery and Restart
DBRC overview

Related Reading: See the **IMS V9: Operations Guide** for detailed discussions of the IMS logging process.

After you have initialized DBRC, it participates in the IMS logging process by recording and controlling information about IMS’s logging activities. This information is recorded in the RECON data set. If you want DBRC to control the recovery of your database data sets (DBDSs), you must register them with DBRC.

DBRC automatically records many items in the RECON data set, including:

- Information about log data sets
- Information about database data sets
- Information about events, such as:
  - Database updates
  - Database authorizations
  - Creation of database image copies
  - Reorganizations of databases
  - Recoveries of databases
  - Archiving of an OLDS and creation of the corresponding SLDS and RLDS
  - Execution of the Log Recovery utility
  - Subsystem sign-on
- Definitions and information about events for Remote Site Recovery

IMS uses this information for restarting itself and for database recovery jobs (if the databases are registered with DBRC). DBRC also tracks the archiving requirements of the OLDS and, if requested, generates and submits the JCL for archiving jobs.

DBRC also provides unit-of-recovery management for all attached subsystems. DBRC provides information about these units of recovery for batch backout, dynamic backout, partial backout, and restart.

For logs produced by batch systems, you are not required to use DBRC. The advantage of using DBRC for batch jobs is that DBRC records information about all the log data sets that are produced by batch jobs and prevents batch update jobs from executing if you specify a dummy or null log data set.

Attention: If registered databases are updated without DBRC control, DBRC cannot correctly control recovery for the databases and database integrity can be jeopardized.

Changing Log Records in the RECON Data Set

Use the `CHANGE.PRILOG` or `CHANGE.SECLOG` commands to modify the information about OLDSs, RLDSs, or SLDSs in the log record of the RECON data set. Update the log record to indicate when errors have occurred on the data sets or that volume serial numbers have changed. You do not normally need to use these commands.

Archiving an OLDS

Run the Log Archive utility to archive an OLDS to an SLDS so that IMS can reuse the OLDS. How frequently you should archive depends on the load on the subsystem and the number of log records written to the OLDSs.

The Log Archive utility always produces an SLDS. The SLDS contains all log records that are required for both database recovery and for online IMS restart.
You can ask the Log Archive utility to produce an RLDS in addition to an SLDS. The RLDS contains only those log records that are required for database recovery.

If you request an RLDS, information about the output RLDS data sets is recorded in the PRILOG record in the RECON data set and information about the SLDS data sets is recorded in the PRISLD record. If you do not request an RLDS, the same information about the SLDS data sets is recorded in both the PRILOG and PRISLD records.

If there is a secondary OLDS, or if you request that dual logs be produced from a single OLDS, the information about the secondary-log output is recorded in corresponding SECLOG and SECSLD records.

**Important:** Log data sets that are output from IMS batch jobs are technically SLDSs, but the information about them is recorded in the PRILOG and SECLOG records.

Run the Log Archive utility by issuing the `GENJCL.ARCHIVE` command. DBRC then determines which OLDSs are full, and generates the appropriate JCL.

**Related Reading:** See Member DFSVSMxx in the [IMS V9: Installation Volume 2: System Definition and Tailoring](#) for more information on the ARCHDEF statement and automatic archiving.

**Recommendation:** Whether you use automatic archiving or invoke archiving yourself, make sure the archive jobs run as quickly as possible. The online subsystem only reuses an OLDS after it has been archived. If the archive job is not run and all the OLDS become full, the online subsystem waits. One way to ensure that archive jobs run quickly is to use an initiator that runs with a fairly high priority and is not used by many other users. This ensures that the archive jobs do not remain on the internal reader queue for too long.

If DBRC has marked an OLDS in the RECON data set as having errors, the `GENJCL` function does not submit it for archiving. If one of a pair of OLDSs has been destroyed or is unavailable, you can choose to mark it in the RECON data set as having errors.

The following references point to where you can find more information about archiving log records.

**Related Reading:**
- See [IMS V9: Utilities Reference: System](#) for more information about specifying entry points and running the Log Archive utility.
- Refer to [IMS V9: Customization Guide](#) for more information about the Log Archive and the Logger exit routines.

**DBRC Log-Related Commands**

Use the following commands to perform DBRC log-related functions in your operational procedures:

- `CHANGE.PRILOG`
- `CHANGE.RECON`
- `CHANGE.SECLOG`
In addition to the LIST.LOG command, you can use a Log or OLDS query API request to retrieve log-related information from the RECON data set.

**Related Reading:** See the command chapters in Part 2, “DBRC Command Reference,” on page 117 for descriptions of the commands and “Log Query” on page 395 for the details of the Log and OLDS query requests.

**Recovering Databases**

This topic provides an overview of how databases are recovered. Included in this topic are:

- “The Basic Steps of Database Recovery”
- “How DBRC Helps in Recovery” on page 12

**The Basic Steps of Database Recovery**

The recovery process for IMS databases can include these three basic steps, although the details of the process can vary with the type of database to be recovered:

1. Restore the database to the most current image copy.
2. Use the log data sets (or change accumulation data sets) to restore changes made to the database since the image copy was made.

   **Definition:** Change accumulation is the process of creating a compacted version of one or more IMS log data sets by eliminating records not related to recovery, and by merging multiple changes to a single segment into a single change. For more information about change accumulation, see “Log Record Change Accumulation” on page 43.

3. Back out any incomplete changes.

   Figure 2 illustrates a simple database recovery.

   ![Figure 2. IMS Database Recovery process](image)

   Information for a database recovery can come from any or all of the following sources:

   - Image copies of the database
• Database reorganization data sets
• Log data sets (SLDSs and RLDSs)
• Change accumulation data sets

You can use DBRC to track all of these information sources, greatly simplifying the task of database recovery.

**Related Reading:** Refer to [IMS V9: Operations Guide](#) for more information about the recovery process.

If you register recoverable databases in the RECON data set, DBRC records the association of the databases to the log data sets containing database change records.

DBRC also records information about:
• Database image copies
• Reorganizations (except DEDB online reorganizations)
• Recoveries
• Change accumulations
• Backout

DBRC can generate JCL for executing a database recovery, because DBRC records this information in the RECON data set. Whether you use the GENJCL commands to generate JCL or provide the JCL yourself, DBRC uses information in the RECON data set to determine exactly which data sets are required for input. The Database Recovery utility runs only if DBRC verifies that the JCL is correct.

You can omit all logged changes after a certain time from the input by performing a *time-stamp recovery*. A time-stamp recovery is equivalent to backing out the omitted changes from the database.

Most time-stamp recoveries require DBRC in order to be successful. When you involve DBRC in your request for a time-stamp recovery, DBRC selects the correct logs and, at execution time, communicates to the Database Recovery utility where to stop processing the input to correctly perform your request.

[Figure 3 on page 12](#) shows how DBRC works with the Database Recovery utility.
**Recommendation:** Implement DBRC in phases, defining at first only a few recoverable databases in the RECON data set. This allows you to gain experience in the use of DBRC, and gives you an opportunity to assess, make, and test any changes needed in your backup, recovery, and operational procedures.

**How DBRC Helps in Recovery**

DBRC is invoked by recovery tools and utilities to provide information about the resources required for database recovery. These resources can include information about image copies, logs, change accumulations, and database data sets.

DBRC can:

- Generate JCL that can be used to run various utilities (see “Generating Recovery JCL” on page 13)
- Validate the input to those utilities (see “Validating Utility JCL” on page 14)
- Record the result (in the RECON data set) of running the utilities (see “Recording the Result” on page 15)
Generating Recovery JCL: You can use the GENJCL.RECOV command to generate the JCL that is necessary to recover a registered database data set. Using information recorded in the RECON data set, DBRC:

1. Selects the image copy data set to use for loading the most recent image copy
2. Selects the change accumulation and log data sets that are to be input to applying all the changes that were logged since the image copy was created

If change accumulation input is required (because of data sharing), but it is not present or usable, DBRC informs you of that fact and the GENJCL.RECOV command fails.

The GENJCL.USER command can generate user-defined output, which can include JCL. No skeletal JCL execution members are supplied to support the GENJCL.USER command. If you want to enter GENJCL.USER commands, you must supply the members to support them.

Issue the GENJCL command to request that DBRC generate JCL in batch or issue the /RMGENJCL command online. When you enter either command, DBRC reads skeletal JCL and replaces symbolic parameters with actual values based on the information recorded in the RECON data set to build the appropriate JCL. For example, if you request that DBRC generate JCL to recover a database, DBRC retrieves the skeletal JCL member from the library and completes the JCL with information about the latest image copy, change accumulation, and log data sets, if necessary. Your databases must be registered in order for DBRC to generate JCL to process them.

The amount of time and effort required to recover a database can be significantly reduced by using the GENJCL to generate the JCL and control statements necessary for the recovery. Using the GENJCL command also eliminate the causes of many recovery errors. You could spend a large amount of time during database recoveries determining which input data sets should be provided in what order to the Database Recovery utility.

When change accumulation data sets or PRILOG records (in the RECON data set) are available, DBRC selects them rather than the SLDS for recovery. This results in quicker database recoveries if you run the Database Change Accumulation regularly. DBRC knows which log data sets are required and ensures that IMS processes all volumes in the correct order. DBRC also selects the most recent image copy for database recovery.

DBRC always selects the optimum input for the Database Recovery utility by using change accumulation data sets whenever possible. If you have not used the Database Change Accumulation utility, or if that utility did not process some log data sets, DBRC selects the required log data sets from the PRILOG (or SECLOG) records, which can contain RLDS, SLDS, or both RLDS and SLDS entries.

Related Reading:
- See IMS V9: Installation Volume 2: System Definition and Tailoring for more information about the tailoring actions for IMS.PROCLIB members, the DBRC procedure, and the JCOLUT and JCLPDS DD statements.
- See Chapter 5, “Skeletal JCL” on page 83 for details about customizing your own skeletal JCL and about the contents of IMS-supplied skeletal JCL.

Recommendation: For increased availability of data entry databases (DEDBs), use the DEDB Area Data Set Create utility to provide additional usable copies of an
online area. It does not provide backup copies for recovery. The DEDB Area Data
Set Create utility uses the RECON data set as part of its input.

Validating Utility JCL: When the a recovery utility processes a registered
database data set, the utility presents its input to DBRC for validation. Whether the
recovery JCL was created by you or by DBRC, DBRC verifies that the input JCL to
the utility is correct (according to the current information in the RECON data set).
It is possible, even if you created the JCL with the \$ENJCL command, that
intervening events could invalidate the input JCL before the utility is run.

DBRC is invoked by the following IMS utilities and services to validate input and
record the results:
• Index/ILDS Rebuild utility (DFSPREC0)
• Database Image Copy utility (DFSUDMP0)
• Database Image Copy 2 utility (DFSUDMT0)
• Online Database Image Copy utility (DFSUICP0)
• Database Change Accumulation utility (DFSUCUM0)
• Batch Backout utility (DFSBBO00)
• Database Recovery utility (DFSURDB0)
• Log Recovery utility (DFSULTR0)
• Log Archive utility (DFSUARCI0)
• HALDB online reorganization
• HD Reorganization Unload utility (DFSURGU0)
• HD Reorganization Reload utility (DFSURGL0)
• HISAM Reorganization Unload utility (DFSURUL0)
• HISAM Reorganization Reload utility (DFSURRL0)
• Database Prefix Update utility (DFSURGP0)
• DEDB Area Data Set Create utility (DBFUMRI0)
• /RECOVER commands

Figure 4 on page 15 shows DBRC’s role in running the previously mentioned
utilities.
When you run the Batch Backout utility (DFSBB00), DBRC determines the complete set of logs that are needed for a particular backout job. In addition, DBRC manages information about the logs so that backout and restart jobs can be easily coordinated.

**Exception:** DBRC does not verify the JCL input for the HD and the HISAM Reorganization utilities, but does record information about their execution in the RECON data set.

**Recording the Result:** When the recovery completes successfully, DBRC records information about the recovery in the RECON data set. If you performed a time-stamp recovery, DBRC records information about the range of omitted changes.

**Requirement:** If a time-stamp recovery is performed within a database allocation interval, you must immediately create an image copy to ensure a valid starting point for subsequent recovery attempts. DBRC then prevents the changes from being reapplied on any subsequent recovery.
Recording Information About Opening and Updating Databases

After IMS opens a database, DBRC passes back the RECON data set initialization token (RIT) and any extended error queue elements (EEQEs) associated with each DBDS. The RIT allows IMS to determine whether the database has been used without DBRC or whether the database has been controlled by a different RECON data set.

Related Reading: See IMS V9: Operations Guide for information on EEQEs.

When changes to DBDSs and areas occur, DBRC records information about these changes in the RECON data set. DBRC subsequently uses this information to determine which log data sets might contain change records for a given DBDS or area.

When a DBDS that is registered in the RECON data set is first opened for updates (or allocated), IMS tells DBRC to create an ALLOC record. In the case of a DEDB area, the ALLOC record is created when the area is first opened for update. The ALLOC record identifies the DBDS or area and contains the time stamp of the first update and the open time stamp of the corresponding PRILOG.

When DBRC creates the ALLOC record, DBRC also enters the name of the DBDS or area being changed in the LOGALL record for the PRILOG that is active at the time of the change.

When you de-allocate (close) a DBDS or area using a /DBRECOVERY command from the operator console of the online IMS subsystem, DBRC writes a de-allocation time stamp in the ALLOC record. If no de-allocation time is recorded, DBRC uses the closing time of the associated log as the de-allocation time. Thus the RECON data set contains a list of the names of DBDSs or areas for which change records might exist on a given log data set (LOGALL record) and a list of the time ranges where changes could exist for a specific DBDS or area (ALLOC records) and a list of the logs containing the changes.

Supporting Data Sharing

Data sharing requires that the databases be registered with DBRC. DBRC checks that subsystems have authority to perform the requested task and that other subsystems are not currently reserving the database.


Levels of Data Sharing

DBRC supports the two levels of IMS data sharing:

Database level

The entire database or DEDB area is a resource that can be accessed for update by a single IMS system at a time. For area resources this can also be called Area-level sharing.

Block level

A database or DEDB area can be accessed by multiple IMS subsystems concurrently. Data integrity is preserved for the IMS subsystems that access the shared data. Within a database or area, resources are reserved at the block level.

Definition:
• For OSAM databases, the block is a physical data block stored on DASD. For VSAM databases and DEDBs, the block is a control interval (CI).

**Recording Data-Sharing Information in the RECON Data Set**
DBRC records the following data-sharing information in the RECON data set for all registered databases:
- Sharing level allowed for each database
- Names of databases or areas currently authorized for processing
- Names of IMS subsystems that are involved
- Statuses of the IMS subsystems
- Database statuses from a recovery viewpoint

**Assigning a Sharing Level with DBRC**
The sharing level of a database or DEDB area determines whether a request for access is granted. DBRC allows you to establish one of four sharing levels.

The following sharing levels are defined using the INIT.DB command and modified with the CHANGE.DB command.

**SHARELVL 0**
The database is not to be shared. The database can be authorized for use by one IMS system at a time. SHARELVL 0 is equivalent to specifying ACCESS=EX on the /START command.

**SHARELVL 1**
Sharing is at the database level. One IMS system can be authorized for update at one time; any sharing systems can only be authorized for read-only processing. Otherwise, the data sharing is for multiple readers.

**SHARELVL 2**
Sharing is at the block level but only within the scope of a single IRLM and a single z/OS. Sharing requires that IMS subsystems sharing a database use the same RECON data set. Multiple IMS systems can be authorized for update or read processing.

**SHARELVL 3**
Sharing is at the block level by multiple IMS subsystems on multiple Realms. Multiple IMS systems can be authorized for non-exclusive access. The IMSs can be on multiple z/OS images using different IRLMs.

**Note:**

To ensure the integrity of databases in data sharing environments when batch jobs running without IRLM support access SHARELVL 3 databases, DBRC authorizes batch jobs that have update access only if all other IMS systems and batch jobs that are currently authorized by DBRC to the database have read-only access.

If IRLM=N is specified in the IMSCTRL macro or DBBBATCH procedure, DBRC authorizes a batch job for update access to a database only if all other IMS systems and batch jobs that are currently authorized by DBRC to the database have read-only access.

When a batch IRLM=N job has authorization for update, authorization for an online system or other batch job fails unless it is for read-only access.
Supporting Remote Site Recovery

DBRC assists you with the definition and management of IMS components in the RSR complex. In support of RSR, DBRC provides:

- Commands to define, update, and display the status of the RSR complex.
- Services that are used by an active subsystem to identify the tracking subsystem and the databases covered by RSR.
- Services used by a tracking subsystem to record information about log data that is received from an active subsystem.
- Tracking subsystem database support:
  - Two types of tracking (called shadowing): DB level tracking (DBTRACK) or Recovery level tracking (RCVTRACK).
  - Maintains log data set information for online forward recovery.
  - Records which database change records have actually been applied to the covered databases.
- Services to assist in the takeover process
  - During a remote takeover, DBRC changes the state information of the registered databases at the new active site to indicate that they are now the active databases.

Related Reading: See the IMS V9: Operations Guide and the IMS V9: Administration Guide: System for more information about RSR.

Supporting IMSplexes

When an I/O error occurs on a RECON data set in an IMSplex and a spare data set is available, the instance of DBRC that noticed the error copies the good RECON data set to the spare, activates the spare, and deallocates the original RECON data set.

At this point in the processing, the DBRC that noticed the I/O error can automatically notify the other DBRCs in the IMSplex about the reconfiguration. Then, after the original RECON data set is deallocated, it can be deleted and redefined as the spare RECON data set.

Related Reading: See the IMS V9: Common Service Layer Guide and Reference for more information about IMSplexes.
Defining DBRC to IMS

You initialize DBRC as part of the IMS system definition process. This section provides a brief overview of the requirements for creating and executing a DBRC region.

Related Reading:
- See Chapter 4, “Initializing and Maintaining the RECON Data Set,” on page 55 for information on creating and allocating the RECON data set.

IMS online systems always use DBRC; you cannot override this.

You can choose whether IMS batch jobs use DBRC. But you must understand that certain functions, such as data sharing, cannot be used without DBRC.

Batch and utility regions can use the DFSIDEF0 module during initialization. In DFSIDEF0, you can set the DBRC= keyword to YES, NO, or FORCE. The DFSIDEF0 module that is shipped in the IMS.ADFSSMPL library contains a batch and utility region default of DBRC=YES. This value is coded on the DFSIDEF macro.

You are not required to use the DFSIDEF0 module. If you do not use the module or if the module cannot be loaded at initialization time, IMS defaults to DBRC=YES. Therefore, if DBRC is to be used for your batch and utility regions, then creating the DFSIDEF0 module is not required.

Related Reading: For more information about the DFSIDEF0 module, see the [IMS V9: Installation Volume 2: System Definition and Tailoring]

The IMSCTRL Macro

As of IMS V9, the DBRC= parameter on the IMSCTRL macro is ignored by IMS.

Use the DBRCNM= parameter to request that IMS create a cataloged DBRC procedure.

Place DBRC’s load modules into a load library that is in the normal load library search sequence for your IMS load modules, for example, IMS.SDFSRESL.

IMS.PROCLIB Execution-Parameter Members

You can override the DBRC procedure name specified in your system definition in the DFSPBIMS, DFSPBDBC, and DFSPBDCC members.

IMS Procedures and DBRC

The EXEC parameter, DBRC=, determines whether DBRC is used in a batch procedure. It is ignored by an online IMS.

The DBRCNM= parameter can be used to override the DBRC procedure name for an online IMS execution.
DBRC overview

DBRC Procedure

IMS automatically starts the DBRC procedure by issuing a z/OS START command during control region initialization. This procedure specifies parameters for the DBRC region. DBRC runs in its own address space for an online IMS subsystem.

To include the DBRC procedure during system definition, copy the skeletal DBRC procedure from IMS.PROCLIB to SYS1.PROCLIB. The member name must match the name specified on the DBRCNM parameter in the IMSCTRL macro or the applicable EXEC procedure.

If DBRCNM is specified in more than one place, or if DBRCNM is not explicitly specified, the following order of precedence applies:

- DBRCNM=DBRC is the default.
- DBRCNM=name in the IMSCTRL macro overrides the default.
- DBRCNM=name defined in a DFSPBxxx member in IMS.PROCLIB overrides the IMSCTRL macro setting.
- DBRCNM=name defined in a JCL EXEC parameter overrides the IMS.PROCLIB member setting.

Related Reading: See the IMS V9: Installation Volume 2: System Definition and Tailoring for a complete description of the DBRC procedure and its parameters.

Initializing the RECON Data Set

Use the Access Method Services (IDCAMS) DEFINE CLUSTER command to create the RECON data sets and then use the INIT.RECON command to initialize the RECON data sets as usable by DBRC.

If you do not intend to register databases, the INIT.RECON command is the only command you need to issue in order to initialize the data set.

Related Reading:
- See Chapter 4, “Initializing and Maintaining the RECON Data Set,” on page 55 for information about creating the RECON data set.
- See “Registering Databases and Database Data Sets” for information about registering databases.
- See “INIT.RECON” on page 280 for detailed information about the INIT.RECON command.

Registering Databases and Database Data Sets

The RECON data set must have a DB record for each database whose recovery DBRC is to control.

If you want DBRC to control database recovery, you must register the databases in the RECON data set. DBRC then records information about database updates and about the corresponding log data sets that contain updated log records. DBRC also records the creation of image copy and change accumulation data sets, and records database recoveries and reorganizations that affect registered databases.

For non-HALDBs (High Availability Large Databases), use the INIT.DB and INIT.DBS commands to register databases in the RECON data set and to define them as recoverable or non-recoverable. For HALDBs, you can use the INIT.DB and INIT.PART commands, or the HALDB Partition Definition utility.
For each non-HALDB database that you have registered, issue the \texttt{INIT.DBDS} command to register all its data sets or DEDB areas. For DEDBs, use the \texttt{INIT.ADS} command to identify the data sets within each area. An area can have up to seven area data sets (ADSs).

\textbf{Requirement:} Utility jobs that issue either an \texttt{INIT.DB} or \texttt{INIT.PART} command (to register a HALDB) or any \texttt{INIT.DBDS} command, require a ddname for the IMS.DBDLIB data set that contains an entry for the HALDB DBD (or non-HALDB DBDS) for which you are issuing the command. A ddname for the IMS.DBDLIB is not needed for non-HALDB \texttt{INIT.DB} command.

\textbf{Related Reading:} See Chapter 12, “INIT Commands,” on page 259 for specifics about the INIT commands.

From a DBRC perspective, a HALDB consists of a HALDB master (TYPE=HALDB) and HALDB partitions (TYPE=PART).

To update or delete information about HALDBs in the RECON data set, you can use the HALDB Partition Definition utility or the following DBRC commands; CHANGE.DB, CHANGE.DBDS, CHANGE.PART, DELETE.PART, or DELETE.DB.

\textbf{Related Reading:}
- See the \textbf{IMS V9: Administration Guide: Database Manager} for an overview of HALDBs and detailed information about how to create them.
- See the \textbf{IMS V9: Utilities Reference: Database and Transaction Manager} for information about the HALDB Partition Definition utility.
- See Chapter 9, “CHANGE Commands,” on page 133 for information about the CHANGE commands.

\section*{Planning for Recovery}

Planning for recovery (in non-data sharing and data sharing environments) involves these primary tasks:

- Setting up the mechanisms of recovery (logging, taking checkpoints, keeping records)
- Setting up your operational procedures for situations that require recovery

The following sections describe some additional aspects of planning for recovery:

- “Setting Up Recovery Mechanisms”
- “Recovery Facilities” on page 22
- “Recovery without DBRC” on page 24
- “Restart After an IMS Failure” on page 24
- “Restart After a DBRC Failure” on page 25
- “Recovery Involving IRLM Configurations” on page 25
- “Batch Backout” on page 25

\section*{Setting Up Recovery Mechanisms}

The logging and checkpoint mechanisms of online IMS subsystems in a non-sharing environment are also active in a data sharing environment. These include:

- System log data sets and the use of the WADS and restart data sets
- Program, system, and message-queue checkpoints
Making image copies of databases

The primary difference between non-sharing and data sharing environments is in their degree of reliance on DBRC. DBRC helps control the data-sharing environment; it does not merely keep records.

Specifically, the following items are different in a non-data sharing environment:
- Database reconstruction (forward recovery)
- Program-level (dynamic) backout
- Database backout
- IMS restart (with embedded recovery)

Related Reading: See the [IMS V9: Operations Guide](#) for detailed information on recovery procedures in non-data sharing and data-sharing environments.

Recovery Facilities

The following recovery facilities behave differently in non-data sharing and data-sharing environments:
- "Dynamic Backout"
- "Batch Backout Support"
- "Forward Recovery" on page 24

Also included in this section is "Limitations of DBRC Backout Support" on page 23.

Dynamic Backout

In non-data sharing and data-sharing environments, an online IMS subsystem dynamically backs out the uncommitted database changes of an application program and discards its uncommitted output messages under either of these conditions:
- The program fails.
- The program requests backout with a rollback call.

In a data-sharing environment, however, IRLM locks and status indicators in the RECON data set ensure integrity by protecting uncommitted changes from sharing subsystems. Operation of the system and other programs continues uninterrupted.

Related Reading: See the [IMS V9: Operations Guide](#) for detailed information about dynamic backout.

Batch Backout Support

DBRC improves database integrity by interfacing with IMS restart dynamic backout and the Batch Backout utility to control access to databases in need of backout. DBRC creates a BACKOUT record to track each unit-of-recovery (UOR) for each database in need of backout. DBRC verifies logs that are input to the Batch Backout utility.

BACKOUT records are created for online subsystems. BACKOUT records are not created for DL/I batch subsystems unless dynamic backout was being used and it failed.

Prior to performing an emergency restart with the COLDSYS parameter (/ERESTART COLDSYS), run batch backout with either the COLDSTART or ACTIVE parameter. After the batch backout completes, DBRC creates a BACKOUT record for all
in-flight and indoubt UORs as candidates for backout. The next IMS restart promotes candidate UORs to backout-needed status and DBRC sets BACKOUT NEEDED=ON in the database records in the RECON data set.

When dynamic backout fails, a BACKOUT record is created with a UOR indicating dynamic backout failure and DBRC sets BACKOUT NEEDED=ON in the database record in the RECON data set.

When the databases are backed out successfully, DBRC resets the backout flags in the database records (in the RECON data set) appropriately and updates the BACKOUT records. When backout processing for all of the UORs have been completed, DBRC deletes the BACKOUT record from the RECON data set.

For DBCTL, if you choose not to back out an unresolved indoubt UOR, use either the DELETE.BKOUT or CHANGE.BKOUT command to remove the unresolved indoubt UOR from the BACKOUT record in the RECON data set.

DBRC commands are available to update BACKOUT records if needed. The need to make manual changes to the backout record should be minimal.

**Attention:** Use the DELETE.BKOUT or CHANGE.BKOUT commands with extreme caution because you could compromise database integrity.

DBRC verifies the validity of the logs used as input to the Batch Backout utility. DBRC checks the input log volumes to ensure that they are in the correct sequence, that all the needed logs are provided, and that they are properly closed. When you include ACTIVE or COLDSTART statements in the Batch Backout utility SYSIN statement, DBRC performs an additional check to ensure all volumes related to restart are included. For DL/I batch logs, a check is also performed to ensure that the correct volumes are from the last batch execution.

**Attention:** If BYPASS LOGVER is included in the SYSIN statement of the Batch Backout utility, then DBRC does not verify the input log and the Batch Backout utility does not notify DBRC about the UORs. Existing UORs are updated when backout processing has completed successfully.

**Related Reading:** See [IMS V9: Operations Guide](#) and [IMS V9: Utilities Reference](#) for more information about UORs, the backout process, and the Batch Backout utility.

**Limitations of DBRC Backout Support**

Although DBRC’s backout support protects databases from damage caused by the most common errors, the following list describes some other possible sources of damage to databases:

- If an IMS subsystem abends and an /ERE COLDSYS command is issued before the in-flight and indoubt UORs have been identified to DBRC (by a Batch Backout run), the databases associated with those in-flights and indoubts are not protected from erroneous updates until the first Batch Backout run using the COLDSTART or ACTIVE statements.
- Including logs from multiple runs of a batch job in the same log data set (by specifying DISP=MOD) makes log verification unreliable.
- You can modify the RECON data set with DBRC commands. Careless use of these commands could lead to errors, such as:
  - Allowing backouts that should not be performed.
  - Allowing other subsystems access to databases that need to be backed out.
DBRC overview

- The Batch Backout BYPASS LOGVER control statement allows backouts to be performed when information in the RECON data set indicates that the input log is invalid or that the backouts are not needed. Careless use of this control statement can cause backouts to be performed that should not be performed.
- Unregistered databases are not protected from being used while backout is needed.
- Backing out a normally terminated job (that did not use IRLM) after a time-stamp recovery was performed (to recover the database to a point in time prior to this log) makes log verification unreliable. DBRC will erroneously validate that the log can be used as input because the log being backed out is the last log for this subsystem name (SSID).

Forward Recovery

The process of recovering a database in a data-sharing environment has certain similarities to recovering a database in a non-data sharing environment. In both environments, use DBRC for database recovery and use the following as input to the recovery process:

- The most recent image copy of the lost database
- The recovery log data set (RLDS) or all pertinent system log data sets used since that copy was made

**Recommendation:** Use the RLDS as input to the recovery process. The RLDS is a more efficient input to the recovery process than the pertinent system log data sets because the RLDS contains only database change log records.

If you are using the Database Recovery utility for forward recovery and block-level data sharing is involved, you might require the additional step of change accumulation.

**Related Reading:** See the [IMS V9: Operations Guide](#) for further discussion of forward recovery.

Recovery without DBRC

If you perform any recovery-related actions offline when DBRC is not running, such as making database image copies, problems could arise because DBRC would not be notified of the changes in status. Therefore, DBRC must be specifically informed of such changes. DBRC offers you several commands for this purpose.

**Related Reading:** See the [IMS V9: Operations Guide](#) for detailed information about recovery without DBRC and its related commands.

Because restart is required, the Utility Control Facility (UCF) might have to be used without DBRC being active. If you use UCF for any recovery-related work, DBRC must be informed of status changes afterward.

Restart After an IMS Failure

Restart an IMS subsystem in a data sharing environment in the same way as in a non-data sharing environment:

- If you can shut IMS down normally (a checkpoint shutdown using the /CHECKPOINT PURGE, /CHECKPOINT DUMPQ, or /CHECKPOINT FREEZE commands), use the /NRESTART command to restart IMS normally.
- If the IMS failed, you must use the /ERESTART command to perform an emergency restart.
In a data-sharing environment, if the associated IRLM also stops or fails, you must first start IRLM before you start IMS in a block-level environment.

**Related Reading:** See the [IMS V9: Operations Guide](#) for more information about restarting IMS after it fails.

### Restart After a DBRC Failure
IMS knows whether its DBRC fails because DBRC runs under the control of IMS. If a DBRC failure occurs, IMS terminates abnormally, attempts to flush its buffers, and closes the system log. After correcting the DBRC problem, you can restart IMS by issuing the `/ERESTART` command.

### Recovery Involving IRLM Configurations
DBRC records IRLM status information in the subsystem record in the RECON data set.

**Related Reading:** See Appendix B, “Sample Listings from the RECON Data Set,” on page 469 and [IMS V9: Operations Guide](#) for information on the IRLM status information that DBRC records in the subsystem record.

### Batch Backout
Prior to executing an emergency restart with the COLDSYS parameter (`/ERESTART COLDSYS`), run the Batch Backout utility with either the COLDSTART or ACTIVE parameter. Provide DBRC with the names of the logs to protect registered databases (that need to be backed-out) from being accessed until their backouts are complete.

For DB level control, if you choose not to back out an unresolved in-doubt UOR, use the `DELETE.BKOUT` command to remove it from the BACKOUT record.

**Attention:** Use the `DELETE.BKOUT` command with extreme caution. It deletes all backout information for a subsystem from the RECON data set.
Chapter 2. Database Recovery Control Utility (DSPURX00)

In This Chapter:

- “What Is the Database Recovery Control Utility (DSPURX00)?”
- “Input and Output for the Database Recovery Control Utility”
- “Invoking the Database Recovery Control Utility from Your Program” on page 28
- “Example of Database Recovery Control Utility JCL” on page 30

What Is the Database Recovery Control Utility (DSPURX00)?

Use the Database Recovery Control utility to issue commands that build and maintain the RECON data set, add information to the RECON data set, and generate jobs for utilities.

Commands submitted to the Database Recovery Control utility have the same general format. Each command is composed of a verb and a modifier (separated by a period) and then followed by parameters.

Additionally, the Database Recovery Control utility is used to upgrade a RECON data set from an earlier, supported IMS version using the CHANGE.RECON UPGRADE command.

The Database Recovery Control utility runs as a standard z/OS job.

Related Reading:

- See Chapter 7, “DBRC Commands,” on page 119 for information about command syntax.

Input and Output for the Database Recovery Control Utility

[Figure 5 on page 28](#) shows the input and output requirements for the Database Recovery Control utility. Notes that describe the input and output in more detail follow the figure.
Notes® to Figure 5
1. The DBRC command (input to the Database Recovery Control utility)
2. The RECON data set (input to the Database Recovery Control utility)
3. The PDS, which contains the JCL and control statements for the utility that DBRC uses to generate a job (input to the Database Recovery Control utility)
4. The data set that contains the database descriptions for the databases that are under the control of DBRC (occasional input to the Database Recovery Control utility)
5. Jobs created by GENJCL commands (output from the Database Recovery Control utility)
6. The RECON data set, which might have been updated by the utility (output from the Database Recovery Control utility)
7. One or more of the following (output from the Database Recovery Control utility):
   - A listing of the input commands
   - Informational messages associated with their execution or diagnostic messages explaining any failures and return codes
   - A listing of each job that was created in the case of GENJCL commands

Invoking the Database Recovery Control Utility from Your Program

This section is intended to help you understand how to invoke the Database Recovery Control utility from your program, using entry point DSPURXRT to specify alternate ddnames for SYSPRINT, SYSIN, and IMS. You might want to use different ddnames if your calling program already uses these ddnames and you might want the Database Recovery Control utility to use different data sets.

DSPURXRT is invoked by your program through the ATTACH, LINK, or LOAD and CALL macro instructions. The dynamic invocation of DSPURXRT enables respecification of some of the ddnames normally used by DBRC. You may specify an alternate ddname for SYSPRINT, SYSIN and IMS. See “Example of Database Recovery Control Utility JCL” on page 30 for a description of these data sets. The
RECON1, RECON2, and RECON3 DD statements cannot be substituted and must be supplied (if required) by your program. If there is no RECON1 DD statement, the RECON data set will be dynamically allocated. Also, substitution is not allowed for the JCLOUT and JCLPDS DD statements.

**Related Reading:** Refer to z/OS V1R3.0 MVS Programming: Assembler Services reference Volume 1 and z/OS V1R3.0 MVS Programming: Assembler Services Reference Volume 2 for more information on ATTACH, LINK, or LOAD and CALL macro instructions.

Unless stated otherwise, programs that invoke DBRC do not have to pass DBRC any parameters specified on the EXEC statement. DBRC will locate the parameter list passed by z/OS. The invoking program should not alter this parameter list.

DSPURXRT must be called in 31-bit mode. It executes in 31-bit mode and returns to the caller in 31-bit mode with the return code set in register 15. Up to two parameters may be passed on the CALL. The first parameter specifies the address of an option that could be specified in the PARM parameter of the EXEC statement. DSPURXRT currently does not support any options. This address must point to a halfword of binary zeros. The second parameter specifies the address of a list of alternate ddnames for standard data sets used during DSPURXRT processing. If standard ddnames are to be used, the second parameter should either point to a halfword of binary zeros or be omitted.

A processing program invokes DSPURXRT with the ATTACH, LINK, or LOAD, and CALL macros. Before issuing the invoking macro, however, the program must initialize the appropriate registers and operand list contents. The register contents follow standard linkage conventions:

- Register 1 contains the address of the argument list.
- Register 13 contains the address of a save area.
- Register 14 contains the address of the return point.
- Register 15 contains the address of the entry point DSPURXRT.

The argument list pointed to by register 1 consists of up to two pointers to parameters. The last word in the list must have the high order bit on, indicating it is the last word. The first word is the address of options passed to DSPURXRT. Currently, no options are recognized and this word must point to a halfword of binary zeroes. See [Figure 6 on page 30](#) for an illustration of the necessary registers and operand list contents.
The second word is the address of the ddname list (see Figure 6 on page 30). The ddname list consists of the following:

- A halfword containing the total length of the ddname list, including this halfword.
- A reserved halfword.
- An eight-byte field containing the ddname to be used in place of SYSIN, or eight bytes of blanks (X'40') if SYSIN is not to be substituted. The ddname must be left justified and padded on the right with blanks if necessary.
- An eight-byte field containing the ddname to be used in place of SYSPRINT, or eight bytes of blanks (X'40') if SYSPRINT is not to be substituted. The ddname must be left justified and padded on the right with blanks if necessary.
- An eight-byte field containing the ddname to be used in place of IMS, or eight bytes of blanks (X'40') if IMS is not to be substituted. The ddname must be left justified and padded on the right with blanks if necessary.

**Example of Database Recovery Control Utility JCL**

Figure 7 on page 31 is a sample job that initializes the RECON data set and registers one database with two DBDs. Descriptions of the individual numbered JCL statements are listed under Figure 7 on page 31.
The numbers in the following list refer to the JCL statements illustrated in Figure 7.

1. **EXEC** Indicates the program to be executed.

2. **STEPLIB** Points to IMS.SDFSRESL, which contains the IMS nucleus and the required action modules.

3. **SYSPRINT** Defines the destination of DBRC diagnostic messages and the listing output. The destination can be a tape or DASD data set, a printer, or it can be routed through the output stream (SYSOUT). The destination can be a tape or DASD data set, a printer, or it can be routed through the output stream (**SYSOUT**).

4. **RECON** DD statements for RECON1, RECON2, and RECON3 are omitted so that the RECON data set is allocated dynamically.

5. **IMS** Defines the IMS DBDLIB data set. It is required only for the INIT.PART commands; the INIT.DBDS commands; the NOTIFY.REORG commands; the INIT.DB command, if you are initializing a HALDB; and the CHANGE.DBDS commands, if you change a DBDS ddname or area name; and the CHANGE.PART command if you change the KEYSTRNG value.

6. **JCLPDS**, or the DD name you supply with the JCLPDS parameter

   Defines the PDS containing skeletal JCL members. It is required only for the GENJCL commands.

7. **JCLOUT**, or the DD name you supply with the JCLOUT parameter

   Defines the data set which is to receive generated JCL. It is required only for the GENJCL commands.

8. **SYSIN** Defines the source of input commands. SYSIN can be a tape or DASD data set, a card reader, or it can be routed through the input stream (**DD * or DD DATA**).
The following are the input command statements:

**9. INIT.RECON**
Initializes the RECON data set with control information and a default subsystem name of IMS3.

**10. INIT.DB**
Registers a database with sharing level 2.

**11. INIT.DBDS**
Identifies a DBDS for the database. It is to have a maximum of three image copy data sets, which are to be predefined and are to be reused when the GENMAX value is exceeded. GENJCL.IC and GENJCL.RECOV commands issued for this DBDS uses the user-specified skeletal JCL members of the JCLPDS instead of the default members ICJCL and RECOVJCL.

**12. INIT.IC**
Commands identify data sets to receive image copies of the DBDS that was just defined. The data set names are specified using the naming convention described in "Data Set Naming Conventions" on page 33. The ICDSN2 parameter on the second INIT.IC command identifies a duplicate image copy data set.

**13. INIT.DBDS**
Identifies a second DBDS for the database, for which a maximum of four image copies is to be maintained. Image copy data sets are not reused, but are deleted when the GENMAX value is exceeded.

**14. INIT.CAGRP**
Defines a change accumulation group that includes the two DBDSs just defined. A maximum of two change accumulation data sets is to be maintained for this group, which are to be predefined and reused when the GRPMAX value is exceeded.

**15. INIT.CA**
Commands identify change accumulation data sets that are available for future use by the CA group just defined. They are named according to the naming convention described in "Data Set Naming Conventions" on page 33.
Chapter 3. Considerations for a DBRC System

This chapter discusses concepts under the following headings that might be helpful for you to consider for your DBRC system.

In This Chapter:
- “General Considerations for Using DBRC”
- “Data Set Naming Conventions”
- “Database Backup Copies” on page 35
- “DBDS Group Considerations” on page 48
- “DB Groups” on page 49
- “Log Record Change Accumulation” on page 43
- “DBRC Considerations for HALDB Online Reorganization” on page 49

This chapter also describes how to use DBRC to control these database-related and log-related processes:
- Creating backup copies of databases
- Creating DB change accumulation data sets
- Recovering databases
- Protecting databases that need backout
- Archiving OLDSs

General Considerations for Using DBRC

Be aware of the following considerations when using DBRC:
- DBRC does not support main storage databases (MSDBs).
- DBRC plays no role in the processing of GSAM databases, so there is no reason to register them.
- Logging is required for batch jobs that use DBRC and have update access.
- Never update the RECON data set information (such as for DBDSs or log data sets) about a data set that is currently in use.
- Be sure to set the time-of-day (TOD) clock value in the host processors accurately, or unpredictable results might occur in the RECON data set.

Data Set Naming Conventions

The use of z/OS catalog facilities for image copy and change accumulation data sets is optional because DBRC always records volume serial information pertaining to these data sets in the RECON data set. If you catalog image copy and change accumulation data sets, they must have unique data set names.

DBRC provides a data set naming convention to help you generate unique data set names for those image copy data sets (for HSSP image copies and concurrent image copies, as well as standard image copies) and change accumulation data sets that you define for future use.

If you use this convention all the time, uniqueness of your data set names is assured. If you use the convention only occasionally, you are sent a message at the
end of your job step that indicates that you did not follow the naming convention and that duplicate data set names could exist in the RECON data set. DBRC assumes that data set names specified in quotation marks do not follow the naming convention. Therefore, DBRC does not check data set names surrounded by quotation marks.

When you add records to the RECON data set that create data sets for one of the recovery utilities to use in the future and you are using this data set naming convention, you can specify either the fully-qualified data set name or simply the abbreviation high-level-qualifier.*.low-level-qualifier (for example, ALPHA1.*.OMEGA). You can use these abbreviated names on any INIT, CHANGE, DELETE, or NOTIFY.REORG command of DBRC when you are specifying the name of a data set that follows the naming convention. DBRC expands the abbreviated name to its fully-qualified form before it accesses the RECON data set.

The following sections describe additional data set naming conventions:

- “Naming Convention for Image Copy Data Sets”
- “Naming Convention for Duplicate Image Copy Data Sets”
- “Naming Convention for Change Accumulation Data Sets”

### Naming Convention for Image Copy Data Sets

The format for image copy data sets is:

```
high-level-qualifier.dbdname.ddname.IC.low-level-qualifier
```

- **high-level-qualifier**
  - A character string that can be from 1 to 8 alphanumeric characters long. The first character must be alphabetic.

- **dbdname**
  - The database name of the DBDS for which the image copy data set is being recorded in the RECON data set.

- **ddname**
  - The data set ddname of the DBDS for which the image copy data set is being recorded in the RECON data set.

- **IC**
  - Indicates that this is the image copy data set.

- **low-level-qualifier**
  - A character string that can be from 1 to 8 alphanumeric characters long. It must be unique for each DBDS and the first character must be alphabetic.

### Naming Convention for Duplicate Image Copy Data Sets

The format for duplicate image copy data sets is:

```
high-level-qualifier.dbdname.ddname.IC2.low-level-qualifier
```

This format is identical to the convention for image copy data sets, except that the IC2 field indicates that this is a duplicate image copy data set.

### Naming Convention for Change Accumulation Data Sets

The format for change accumulation data sets is:

```
high-level-qualifier.cagrpname.CA.low-level-qualifier
```

- **high-level-qualifier**
  - A character string that can be from 1 to 8 alphanumeric characters long. The first character must be alphabetic.
DBRC Considerations

cagrpname
The name of the CA group for which you are creating the change accumulation data set.

CA Indicates that this is a change accumulation data set.

low-level-qualifier
A character string that can be from 1 to 8 alphanumeric characters long and must be unique for each CA group. The first character must be alphabetic.

Database Backup Copies

When IMS takes a regular system checkpoint, it records internal control information for DL/I (for Fast Path, IMS also records buffers and MSDBs), but it does not record the external contents of the database. If the database is lost, examining the last system checkpoint does not help. The log can tell you what changes have occurred, but without a backup copy of the database, recovery is impossible.

Recommendation: Make a backup copy of the database after you initially load it, and make new backup copies at regular intervals. The more recent the backup copy is, the fewer log change records need to be processed during recovery, thus reducing the time that is needed for recovery.

IMS enables you to make backup copies in these ways:

- The Database Image Copy utility (DFSUDMP0) runs offline and uses access method services to make the copy.
- The Database Image Copy 2 utility (DFSUDMT0) runs offline and invokes DFSMS Concurrent Copy to make the copy.
- The Online Database Image Copy utility (DFSUICP0) runs online and uses IMS services to make the copy.
- The unloaded data that is output from the HISAM Reorganization Unload utility (DFSURUL0) can be used as a backup copy.
- HSSP processing can also create image copies of DEDB areas.

When these utilities run, they can (depending on installation parameters) call DBRC to update essential information in the RECON data set. If you use a supported image copy utility, DBRC records the image copies for registered databases. DBRC also generates the JCL for the utility if you enter the GENJCL.1C or GENJCL.01C command.

You can also use various utilities supplied by the operating system to make your backup copies; however, these do not interact with DBRC, and so you need to take certain actions to notify DBRC of your non-standard image copies. See “Nonstandard Image Copy Data Sets” on page 43 for a discussion of how to notify DBRC about these data sets.

Related Reading: See [IMS V9: Administration Guide: Database Manager] for more information on HSSP processing of DEDB areas.

The Image Copy Utilities (DFSUDMP0, DFSUDMT0, DFSUICP0)

IMS enables you to “take a picture” of your database before and after changes have been made to the database. The “pictures” are called image copies. The term refers to the fact that the copy is an as-is image; the image copy utilities do not
Backup Considerations

alter the physical format of the database as they copy it. Image copies are backup copies of your data that help speed up the process of database recovery.

The Database Image Copy utility (DFSUDMP0), Database Image Copy 2 utility (DFSUDMT0), and Online Database Image Copy utility (DFSUICP0) create image copies of databases. All of the image copy utilities operate on data sets or DEDB areas, so if a database is composed of multiple data sets or areas, be sure to supply the utility with multiple specifications. You can request that one of the supported image copy utilities produce both an image copy data set and a duplicate image copy data set in one run of the utility.

**Recommendation:** Copy all data sets or areas belonging to a database at one time. If you perform multiple recoveries in order to reset a database to a prior state, recover all data sets belonging to the database and to all logically related databases (including those related by application processing) to the same point to avoid integrity problems.

Each of the image copy utilities provide the option to create backup copies without taking databases and areas offline. You can use this capability to provide increased database availability. Image copies taken while the database is available for concurrent update processing by IMS applications are called concurrent image copies or *fuzzy* image copies. Changes already made to the database by active applications might be missing from the copy because the changes might not have been physically written to the data set. These changes, however, have been written to the log. In this case, it is necessary to go back to some earlier point in the logs to ensure that all changes are applied. How far to go back depends on the type of database and which image copy utility was used.

When the concurrent image copy option is not used, the database must be either taken offline or made available only for ‘read’ access and a consistent or ‘clean’ image copy is taken. See "Concurrent Image Copy" on page 38 for more information.

If the image copy was made while the database was not being accessed for update, only changes that were logged after the run time of the copy are required.

When using these utilities, you have the option of creating one to four output image copies. Only the Database Image Copy 2 utility allows three or four output copies and only the first two output copies are recorded in the RECON data set. The advantage of making multiple copies is that if an I/O error occurs on one copy, the utility continues to completion on the other copies. Also, if one copy cannot be read, you can perform recovery using another. The trade-off in deciding whether to make multiple copies, is that performance can be degraded because of the time required to write the additional copies.

DBRC works similarly with the three image copy utilities. The rules for pre-definition and reuse of image copy data sets apply to all three. Each utility calls DBRC:

- To verify the input to the utility (DBRC allows it to run only if the input is valid.
- Calls DBRC to record information in the RECON data set about the image copy data sets that it creates.
Backup Considerations

An image copy record in the RECON data set has the same format whether its corresponding image copy data set was created by the Database Image Copy utility, the Database Image Copy 2 utility, or by the Online Database Image Copy utility.

Two different commands create image copy jobs:
- `GENJCL.IC` for the offline utilities Database Image Copy and Database Image Copy 2
- `GENJCL.01C` for the online utility Online Database Image Copy

When you run batch jobs without logging, take an image copy immediately afterwards; do not count on rerunning the batch jobs, as part of a subsequent recovery, in combination with the Database Recovery utility. The database could be damaged by the combination because the batch processing is not guaranteed to be physically repeatable.

Database Image Copy (DFSUDMP0)

Use the Database Image Copy (DFSUDMP0) utility to create data set copies of:
- HISAM databases
- HIDAM databases
- HDAM databases
- PHDAM (partitioned HDAM databases)
- PHIDAM (partitioned HIDAM) databases
- PSINDEX (partitioned secondary index) databases
- DEDB areas

The Database Image Copy runs offline. It supports a concurrent image copy (CIC) option that enables you to create an image copy while the database remains online. The CIC option can create image copies of OSAM data sets and VSAM Entry Sequenced Data Sets (ESDSs). The Database Image Copy does not support creating image copies of VSAM Key Sequenced Data Set (KSDSs).

To request a concurrent image copy, use the CIC keyword on the `GENJCL.IC` command. Alternatively, you can specify the CIC parameter on the EXEC statement for image copy job. DBRC must be used by the utility and you can only take a concurrent image copy of a database that has been registered with DBRC.

When you run the Database Image Copy utility to take a consistent image copy (CIC option not specified), DBRC is not required but is recommended. DBRC ensures that there is no update activity against the database or area while the utility is executing. If you run the utility without using DBRC you must make certain that no updates occur to the database or area. You can issue a `/DBDUMP` command or a `/ST0P AREA` command, for example, to prevent updating of the database or area by transactions in the system previously doing updates.

Database Image Copy 2 (DFSUDMTO)

The Database Image Copy 2 utility (DFSUDMTO) uses DFSMS Concurrent Copy to take an image copy. It invokes DFSMS to dump the input data set using the Concurrent Copy option. The output is recorded in the RECON data set and can be used for database recovery in the same way as the output from either of the other image copy utilities. You can use this utility for HISAM, HIDAM, HDAM, PHDAM, PHIDAM, and PSINDEX databases and for DEDB areas. Database data sets that are to be copied by this utility must reside on hardware that supports the Concurrent Copy option.
Backup Considerations

An optional COMPRESS parameter invokes the “compress” option in DFSMS Concurrent Copy. The compress option enables you to reduce the storage space required to hold the image copy; however, using the compress option increases the CPU time that is required to perform the copy operation.

The Database Image Copy 2 utility provides greater database availability when taking consistent image copies. The database needs to unavailable for update processing for only a very short period of time while DFSMS establishes a concurrent copy session. Update processing can then be resumed while the image copy data sets are actually being written. The updates are not included in the image copy.

This utility can also copy the database while it is being updated by IMS applications. The image copy created in this case is a concurrent image copy. The Database Image Copy 2 utility can copy all supported data set types, including VSAM KSDSs, while the databases remain online.

Requirement: The Database Image Copy 2 utility must use DBRC and the databases and areas being copied must be registered with DBRC. The utility can create up to 4 output copies, but only the information about the primary and secondary copies are recorded in the RECON data set.

The image copy output from the Database Image Copy 2 utility is in DFSMS dump format, which is different than the format of the output of the other image copy utilities. It is, however, directly usable as input to database recovery. Information about the image copies are recorded in the RECON data set with image copy type SMSCIC or SMSNOCIC.

Requirement: The Database Recovery utility must use DBRC when using the SMSCIC or SMSNOCIC image copy data sets for recovery.

Online Database Image Copy (DFSUICP0)

Online Database Image Copy (DFSUICP0) runs as a BMP program in the online IMS and DBCTL environments. You can use it for HISAM, HIDAM, HDAM, PHDAM, PHIDAM, and PSINDEX databases only; it does not support areas. A database being copied by this utility is available for updating by the IMS subsystem in which the utility is executing. Other IMS subsystems cannot have concurrent update access to the database.

If the database being copied is updated while the utility is running, a fuzzy image copy is produced. Recovery with this image copy requires all logs starting with the log that was in use when the Online Database Image Copy utility was started.

Concurrent Image Copy

IMS provides the capability to take an image copy of a database without taking that database offline. This means the database can be updated while the image copy is being taken and some, all, or none of the updates might appear in the image copy. This image copy is called “fuzzy” because the copy represents the state of the database over a certain period of time rather than at one specific instant in time. It is also called a ‘concurrent image copy’ because the copy was taken while update processing is happening.

The ability to take image copies while the databases are being updated by IMS applications allows increased database availability. The offline image copy utilities, Database Image Copy and Database Image Copy 2, provide an option to take a concurrent image copy. A database being copied by the Online Database Image
Backup Considerations

Copy utility can be concurrently updated by the IMS subsystem in which the utility is running (but not by other IMS subsystems). Image copies created by HSSP processing are also ‘fuzzy’ copies because the areas are available for update processing while HSSP is running.

When a consistent image copy is input to database recovery, the recovery only requires logs from after the image copy job completed. A concurrent image copy, might not include updates that were made before the image copy process started or while it was executing. Therefore, when a concurrent image copy is input to recovery, logs from before the image copy process was started might have to be supplied to database recovery.

**Restrictions:** The following restrictions apply to creating concurrent image copies:
- Only databases that are registered with DBRC are eligible for concurrent image copying.
- The Database Image Copy utility (which can take consistent image copies without using DBRC) must use DBRC when creating a concurrent image copy.
- Non-recoverable databases are not eligible for concurrent image copying because there is no log data to complete the image copy.

Creating Image Copy Data Sets for Future Use and Reuse

Use the REUSE parameter to inform DBRC that you want to be able to define image copy data sets and record them in the RECON data set for future use. You define image copy data sets with the INIT.IC command. When processing the GENJCL.IC command, DBRC selects one of the image copy data sets for use by the image copy utility.

When the Database Image Copy utility uses an available image copy data set, DBRC updates its record in the RECON data set with the time stamp of the run of the Database Image Copy utility during which the image copy data set was used.

If you specify the NOREUSE parameter, you cannot predefine image copy data sets (This is the default). You need to supply the output data set name for the utility in either the skeletal JCL member used in processing the GENJCL.IC command or in the JCL that you produce yourself. When you specify NOREUSE, DBRC dynamically sets the unit type of the output image copy data set. DBRC sets it to the default unit type for the device as specified in the INIT.RECON and CHANGE.RECON commands. Specify NOREUSE when you want more than two DFSMS concurrent copies (you can have up to four).

If you do not specify REUSE, every time the image copy utility is run DBRC deletes the oldest image record that exceeds both the GENMAX and RECOVPD values. The image copy data set itself is not scratched-only its record in the RECON data set is scratched. You must either scratch the data set or keep track of it yourself, because DBRC is no longer aware of its existence.

If you are using the image copy option of HSSP for a DEDB area, the area must be defined with the REUSE parameter and the data sets you predefine must be cataloged.

Controlling the Number of Image Copies Managed

You can use the INIT.DBDS and the CHANGE.DBDS commands to specify how many image copies of the DBDS or area that DBRC is to maintain records of.
Backup Considerations

The GENMAX parameter specifies the maximum number of recovery generations (images) that DBRC maintains for the identified DBDS or DEDB area. Duplicate image copy data sets are not included in this number.

Use the RECOVPD parameter to maintain data for a certain period.

Related Reading:
- See “Recovery Period of Image Copy Data Sets and GENMAX” for more information about the RECOVPD parameter.
- See “INIT.DBDS” on page 266 for more information on the REUSE, NOREUSE, GENMAX, and RECOVPD.

Recovery Period of Image Copy Data Sets and GENMAX

When you register a DBDS in the RECON data set, you specify the maximum number of image copy generations for DBRC to record with the GENMAX parameter of the INIT.DBDS command. When this number is exceeded, DBRC discards the information relating to the oldest image copy. For example, if you take image copies daily and want to keep four days of back level image copies, specify a GENMAX of 4. However, in some emergency situations, you may take backup copies more frequently, possibly three in one day.

To prevent DBRC from discarding information relating to the earlier copies, specify the optional recovery period parameter (RECOVPD) to indicate the number of days you want information retained. If the GENMAX limit is reached, but the RECOVPD for the oldest image copy record has not expired, DBRC issues a warning message (DSP0065I), and does not discard the record. If the DSP0065I warning message appears frequently, you might need to tune the GENMAX or RECOVPD values with the CHANGE.DBDS command.

A recovery period is the minimum amount of time that information is maintained in the RECON data set. For example, if the recovery period of a DBDS or DEDB area is 14 days, DBRC maintains sufficient recovery-generation information for at least 14 days.

If both GENMAX and RECOVPD have been specified for a DBDS or DEDB area, DBRC considers both when deciding whether to reuse or delete an image copy data set.

Table 2 shows the results of GENJCL.IC processing when both GENMAX and RECOVPD have been specified for a DBDS or area defined with the REUSE parameter.

Table 2. Results of GENJCL.IC Processing When GENMAX and RECOVPD are Specified with REUSE

<table>
<thead>
<tr>
<th>Number of Image Data Sets</th>
<th>Number of In-Use Image Copies</th>
<th>Age of Oldest Image Copy</th>
<th>GENJCL Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>=GENMAX</td>
<td>=GENMAX</td>
<td>&lt;RECOVPD</td>
<td>Fail (DSP0063I)</td>
</tr>
<tr>
<td>&gt;GENMAX</td>
<td>=GENMAX</td>
<td>&lt;RECOVPD</td>
<td>Avail IC DS used (DSP0065I)</td>
</tr>
<tr>
<td>=GENMAX</td>
<td>=GENMAX</td>
<td>&gt;RECOVPD</td>
<td>Oldest IC DS reused</td>
</tr>
<tr>
<td>&gt;GENMAX</td>
<td>=GENMAX</td>
<td>&gt;RECOVPD</td>
<td>AVAIL IC DS used</td>
</tr>
</tbody>
</table>
### Backup Considerations

Table 3 shows the results of running an image copy utility when both GENMAX and RECOVPD have been specified for a DBDS or area defined with the NOREUSE parameter.

<table>
<thead>
<tr>
<th>Number of Image Copies</th>
<th>Age of Oldest Image Copy</th>
<th>Utility EOJ Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>=GENMAX</td>
<td>&gt;RECOVPD</td>
<td>Delete oldest image copy</td>
</tr>
<tr>
<td>=GENMAX</td>
<td>&lt;RECOVPD</td>
<td>No delete (DSP0064I)</td>
</tr>
<tr>
<td>&lt;GENMAX</td>
<td>N/A</td>
<td>No delete</td>
</tr>
</tbody>
</table>

If you issue a `CHANGE.0BDS` command and specify new GENMAX and RECOVPD values that are less than the existing values, any used image copy data sets that are beyond the recovery period are deleted until the number of remaining image copy data sets equals the specified GENMAX value.

If you issue the `DELETE.1C` command, any specified image copy data set record is deleted regardless of RECOVPD or GENMAX.

### Reusing Image Copy Data Sets

DBRC enables you to reuse old image copy data sets. The REUSE parameter of the `INIT.0BDS` command, in addition to enabling you to define image copy data sets for future use, enables DBRC to reuse image copy data sets. To reuse the image copy data set means that the same name, volumes, and physical space are used for the new image copy.

A run of one of the IMS image copy utilities automatically reuses the oldest image copy data set for a DBDS or area with the REUSE attribute when all of the following conditions are met:

- A number of image copy data sets equal to the current GENMAX value are recorded in the RECON data set. To see the current GENMAX value, use the `LIST.0BDS` command.
- The Database Image Copy utility or Online Database Image Copy utility used all image copy data sets for this DBDS. None of the image copy data sets is available.
- The oldest image copy is beyond the recovery period.
- A run of one of the IMS image copy utilities automatically uses an available image copy if one exists. If GENMAX has been reached and the oldest image copy is beyond the recovery period or there is no recovery period, the oldest image copy data set is made available.

When you use a `GENJCL.1C` command to generate the job for the Database Image Copy utility, the image copy data set that is to be reused is automatically selected. If the number of image copy data sets is less than the GENMAX value and all image copy data sets have been used, more image copy data sets must be defined for the DBDS or area before running the Database Image Copy utility. The number of image copy data sets should be greater than the GENMAX value if you want to use a recovery period.

The Database Image Copy 2 utility can create up to four output image copy data sets. However, if you issue a `GENJCL.1C` command for a DBDS defined as REUSE, DBRC generates JCL for only one or two output copies (because you can define
image copy data sets for only one or two copies). If you use the GENJCL.IC command for the Database Image Copy 2 utility to process a DBDS defined as REUSE and you want more than two output copies, modify the generated JCL before you run the job.

DBRC generates JCL so that the oldest DASD data set is always used for output for the image copy. DBRC has the same capability with tape volumes. However, you need to analyze your existing tape library techniques to make sure there is no conflict.

**HSSP Image Copy**

The high speed sequential processing (HSSP) image copy (IC) process requires that a database be registered with DBRC and requires that an area be registered with the REUSE attribute for recycling the predefined IC data sets. Image copies that are created by HSSP are concurrent image copies.

**HISAM Copies (DFSURUL0 and DFSURRL0)**

Using the HISAM Reorganization Unload utility (DFSURUL0) to make backup copies of a database lets you process an entire HISAM database in one pass (the image copy utilities deal with single data sets or areas). The unload utility (DFSURUL0) also reorganizes the database as it copies it.

Because the unload utility (DFSURUL0) reorganizes the database, before resuming normal online operations, the data set must be reloaded using the HISAM Reorganization Reload utility (DFSURRL0). The logging, which is done between the unload and reload, reflects the old data set organization.

When using the HISAM utility to make a backup copy, reload immediately, or the actual database and the backup database are mismatched.

The reload utility (DFSURRL0) notifies DBRC. The unload utility creates a reorganized copy of each data set. Then the reload utility reloads each data set from the reorganized copy, and through DBRC, creates a REORG and an IC record in the RECON data set for each data set. The IC record points to the data set that was output from the Unload utility and input to the Reload utility. After a database has been reorganized, a DBDS can be authorized only if an image copy of that data set has been created.

Updates of the database between unload and reload operations must be prevented. Updates of the database made after an unload operation but before a reload operation are wiped-out by the reload operation. In addition, the change records that are logged reflect the old organization, so that a subsequent recovery using those log records damages the database.

You can prevent access to a shared database during reorganization by using one of the following methods:

- From an online IMS subsystem, issue a global /DBRECOVERY command for the database that is to be reorganized. This prevents any subsequent authorizations except for reorganization and recovery. Ensure that recovery utilities do not run during the reorganization.
- Manually update the RECON data set by specifying the NOAUTH parameter of the CHANGE.DB command. This prevents any subsequent authorizations except for the reorganization and recovery utilities. Ensure that recovery utilities do not run during the reorganization. After the reorganization process is complete,
manually update the RECON data set by specifying the AUTH parameter of the CHANGE.DB command for the database that was just reorganized.

**Nonstandard Image Copy Data Sets**

You can create backup copies of DBDSs by using some means other than the Database Image Copy utility. For example, you could make a copy of the volume on which a DBDS resides. DBRC does not record the existence of these nonstandard image copy data sets in the RECON data set automatically; use a NOTIFY.UIC command to inform the RECON data set of these data sets. If this information is not recorded in the RECON data set, DBRC might misinterpret subsequent information about changes to the DBDS. You cannot issue the NOTIFY.UIC command for a DBDS that is defined with the REUSE option.

Before recovering a DBDS or DEDB area with a nonstandard image copy data set, perform the following steps:

1. Close the database using /DBR (without NOFEOV). Load the data set from the nonstandard image copy (UIC) and record the event in the RECON data set (by issuing NOTIFY.RECOV with RCVTIME specified).
2. Apply the change records from the logs that were produced since the UIC (by running DBRC with USEDDBDS or USEAREA for the GENJCL.RECOV command or DFSDFDUMP DD DUMMY statement in the DBRC JCL).

DBRC does not allow DBRC to process any log that contains changes outside the recovery range because an image copy is not used for Step 2. The recovery range is defined by the time-stamp recovery record RECOV TO (image copy time) and RUNTIME values.

**Frequency and Retention of Backup Copies**

Consider the following questions when making backup copies of databases:

- How frequently should I make new copies?
- How long should I keep old (back-level) copies?

There are no precise answers to these questions. Generally, the more frequently you copy, the less time recovery takes. The further back in time your old copies go, the further back in time you can recover. (Remember that program logic errors are sometimes not discovered for weeks.) Conversely, making each new copy requires work, and each old copy that you save uses additional resources.

The only firm guidelines are these:

- If a database is composed of several data sets, be sure to copy all of the data sets at the same time.
- If you reorganize a database, immediately make a new backup copy of it. (This is not necessary for online DEDBs or HISAM reorganizations.)
- If you create a new database, immediately make a backup copy of it.
- If you perform a time-stamp recovery, make a backup copy for use in subsequent recoveries.

**Log Record Change Accumulation**

As IMS runs, the number of stored SLDSs or RLDSs grows. You can use these stored volumes to recover a lost or damaged database, but to use them in their raw form would be inefficient because:
Change Accumulation Considerations

- Each SLDS or RLDS contains a record of activities of the entire system and all the data sets within all the databases. Yet when you are recovering a database, you are doing so for a single data set only. Thus, much of what is on the SLDS or RLDS does not apply.
- The SLDS or RLDS chronologically notes each change to any single record. If a record were changed 100 times since the last backup copy of the data set, the SLDS or RLDS would include 100 such notations. Yet, in recovery, you are only interested in the value the data had at the moment the data set was lost. The previous 99 changes are irrelevant.

The Database Recovery utility requires that change records be input in chronological order, but if a database is shared, the change records might be distributed among different log data sets in a way that makes their input to the utility impossible. A DBRC GENJCL.RECOV command or DB Recovery utility execution fails if this log data has not been properly merged. Such a failure is accompanied by a message informing you that a Change Accumulation should be run.

Condensing the Accumulated SLDS or RLDS (Change Accumulation)

The Database Change Accumulation utility offers you a way of sorting through your accumulated log data sets in advance, merging and condensing them. This utility:
- Merges updates from different subsystems
- Picks out only those log records relating to recovery of databases
- Sorts these records by data set within a database
- Saves only the most recent state of each changed part of each data set.

[Figure 8 on page 45](#) illustrates how the Database Change Accumulation utility merges the relevant log data (a, b, and c) from 3 different SLDSs (A, B, and C), from one IMS system, into one change accumulation data set record.
When Is Change Accumulation Required?

Running the Database Change Accumulation utility is not required if, during the time period in question, only one system updated the database; using it periodically speeds any database recovery that becomes necessary. Alternately, you can run the Database Change Accumulation utility only when the need for recovery arises (just before running DB Recovery utility).

Change accumulation would not be required when non-concurrent data set update information exists in various logs. The database changes are received in the correct order if the logs are input serially to DB Recovery utility.

Change accumulation can be required when concurrent data set update information exists in various logs. The logs cannot be input to DB Recovery utility in a way that the change records are seen in the correct order.

Related Reading: See the [IMS V9: Utilities Reference: Database and Transaction Manager](#) for detailed instructions on running the Database Change Accumulation utility.

Input to the Database Change Accumulation Utility

In addition to stored SLDSs and RLD5s, you can use a previous change accumulation data set and other IMS log volumes as input. The utility writes the accumulated changes in a new change accumulation data set.

You can specify all log volumes or a subset of log volumes as input to the Database Change Accumulation utility. When you specify a subset of log volumes,
Change Accumulation Considerations

DBRC determines whether the subset is complete for each DBDS or area. A subset of log volumes is complete for a DBDS or area when all of the following conditions are true:

- The first volume in the subset is the earliest volume, that could possibly have changes to the DBDS, that were not included in the last change accumulation or in the last image copy.
- The remaining volumes are in sequence.
- In a data sharing environment, logs from all updating subsystems containing changes and any open data streams for a DBDS are included.

The DBRC LIST.CAGRP command indicates whether the log subset for each DBDS of the change accumulation group is complete.

You can use the change accumulation data set as input to a later run of the Database Change Accumulation utility whether your subset of log volumes is complete or incomplete; however, you can use a change accumulation data set as input to DB Recovery utility only if it represents a complete log subset.

**Recommendation:** If DBRC is being used, it is recommended that the change accumulation process is performed using the GENJCL.CA command. This command creates the correct JCL and includes all unprocessed log data sets. If you use your own JCL instead, verify the specifications for change accumulation before execution.

An image copy of the specified database data set is needed and must be identified to the RECON data set in order to create a valid starting point for change accumulation records.

All changes since the last valid image copy are collected by the utility. If a time-stamp recovery has occurred since the last image copy, the change accumulation that is created is invalid for use in future recoveries. No error messages are generated by GENJCL.CA or by the execution of the utility.

You can run the Change Accumulation utility with a valid log subset at any time to reduce data to a minimum.

When the most recent image copy is used as input to the Change Accumulation utility and that image copy is a concurrent image copy, changes already made to the database by active applications might be missing from the copy because the changes might not have been physically written to the data set. These changes, however, have been written to the log. In this case, it is necessary to go back to some earlier point in the logs to ensure that all changes are applied. How far to go back depends on the type of database and which image copy utility was used.

The point-in-time selected to start the Change Accumulation utility is called the **purge time**.

**Related Reading:**

- See the [IMS V9: Utilities Reference: System](#) for more information on the Database Change Accumulation utility.
- See the [IMS V9: Operations Guide](#) for more information on the change accumulation process in general.
Change Accumulation Considerations

Change Accumulation Groups
You can use the Database Change Accumulation utility to process some or all of your recoverable, registered databases as change accumulation (CA) groups. You can, for example, organize the DBDSs into the following groups:

- Application-associated databases
- Physical database clusters
- Logical database clusters
- Volatile (critical data) databases

You can add or delete members of a CA group after you have created it. A database can be a member in only one CA group. To move a member from one CA group to another CA group, you must first delete it from its current CA group and then add it to the new CA group.

Recommendation: Create an image copy of a CA-group member before you move that member to a new CA group. The change accumulation utility processes all logs for the member that are needed for recovery, based on the last good image copy.

A Change Accumulation group can contain from one to 32767 members, which must be registered DBDSs or areas.

Use INIT.CAGRP to create your Change Accumulation group. To modify your Change Accumulation group, use CHANGE.CAGRP. A DBDS or area can belong to only one Change Accumulation group.

DBRC supports the Change Accumulation utility only by Change Accumulation group. Likewise, GENJCL.CA generates JCL only for a Change Accumulation group. The DBRC verification exit routine verifies for a whole Change Accumulation group.

You can run the Change Accumulation utility for DBDSs that do not belong to a Change Accumulation group even with DBRC=YES in effect. However, DBRC does not verify the input to the utility nor record its output.

Defining Change Accumulation Data Sets for Future Use
You can use the REUSE keyword on the INIT.CAGRP command to inform DBRC that you want to define a “pool” of data sets to receive the output from the Change Accumulation utility. Define the data sets using the INIT.CA command. The number of change accumulation data sets you can define, can equal up to the value of the INIT.CAGRP command GRPMAX parameter that defined the group.

If you define a Change Accumulation group with the REUSE parameter and also use a GENJCL.CA command to generate the job for the Database Change Accumulation utility, data set reuse can occur. When all available change accumulation data sets for this Change Accumulation group have been used and the maximum number of change accumulation data sets has been reached, the next run of the Database Change Accumulation utility for this group reuses the change accumulation data set containing the oldest change records. To reuse a change accumulation data set means that its data set name, volumes, and physical space are used as if they were for an empty change accumulation data set.

If you define the Change Accumulation group with NOREUSE:
- You must specify the output data set name for the utility either in the skeletal JCL member used for the GENJCL.CA command or in the JCL that you produce.
Change Accumulation Considerations

- When you run the Change Accumulation utility and the number of data sets specified by GRPMAX has been reached, DBRC deletes the record of the change accumulation data set (from the RECON data set) that contains the oldest change records. The data set itself is not scratched. The data set must be manually scratched or monitored if you want to keep it, because DBRC no longer recognizes its existence.

Related Reading: See “Database Change Accumulation Utility (CAJCL)” on page 433 and “Database Change Accumulation Utility JCL (CAJCL)” on page 446 for details on the Change Accumulation JCL specifications.

DBDS Group Considerations

A DBDS group is a named collection of DBDSs or DEDB areas. DBRC can perform various operations by DBDS group so that you do not need to repeat the command for every member of the group.

You can specify DBDS groups on the following commands:
- GENJCL.IC
- GENJCL.OIC
- GENJCL.RECEIVE
- GENJCL.RECOV
- GENJCL.USER
- LIST.DBDS
- LIST.HISTORY

When you specify a DBDS group on a command, DBRC invokes that command for each member of the DBDS group. For example, you might have a DBDS group for a particular application, like payroll. When performing a time-stamp recovery, for example, all DBDSs of a particular application of a database must be recovered to the same point. If you specify a DBDS group on the GENJCL.RECOV command, you need only invoke the command once to recover all DBDSs.

You can also specify a CA group as a DBDS group. DBRC then executes the command for each member of the CA group.

You can define as many DBDS groups as you want. Up to 2000 DBDSs can be in a group. All DBDSs in a group must be registered in the RECON data set. A DBDS can belong to more than one DBDS group.

A database is an implied DBDS group for the GENJCL and LIST commands. It is unnecessary to define a DBDS group consisting of the DBDSs or areas of a single database. Specify the database name and omit the DD name to operate on the whole database.

The following commands affect the definition of a DBDS group:
- INIT.DBDSGRP
- CHANGE.DBDSGRP
- DELETE.DBDS
- DELETE.DBDSGRP
- LIST.DBDSGRP

DBDS groups can include ILDS (Indirect List Data Set) and index data sets.
**DB Groups**

A DB group is a named collection of databases or DEDB areas. A DB group name can be specified in the /START, /STOP, /DBRECOVERY, and /RECOVER commands, instead of issuing these commands separately for each database or area. Specifying a DB group name with these commands greatly reduces the number of times these commands must be issued. Use the DATAGROUP keyword to specify the DB group name.

DB groups can also include HALDB master and HALDB partition names. Be aware of the effects of a command issued using a DB group that has a HALDB master name and one or more of its partitions. See IMS V9: Command Reference for more information.

You can define as many DB groups as you want. Up to 32767 databases or areas can be in a group. A database or area can belong to more than one DB group and need not be registered in the RECON data set.

**Recommendation:** Use a database group whenever possible, even though a DBDS group can be used as a DB group. Processing a DBDS group as a DB group entails increased overhead.

DB groups are a form of a DBDS group, so they are stored in the RECON data set using the DBDS group record. The following commands affect the definition of a DB group:
- INIT.DBDSGRP
- CHANGE.DBDSGRP
- DELETE.DBDSGRP
- LIST.DBDSGRP

**DBRC Considerations for HALDB Online Reorganization**

The following topics describe the DBRC considerations with respect to HALDB online reorganization.

The following DBRC commands support HALDB online reorganization:
- CHANGE.DB
- GENJCL.IC
- GENJCL.OIC
- INIT.DB
- NOTIFY.IC
- NOTIFY.REORG
- NOTIFY.UIC

The records in the RECON data set that contain HALDB online reorganization information are the:
- HALDB master record
- Partition DB record
- DBDS record

Related Reading: See Chapter 11, “GENJCL Commands,” on page 217 for the impact of these data sets on the GENJCL commands.
DBRC Considerations for HALDB Online Reorganization

- REORG record

**GENJCL.IC and GENJCL.OIC Command Considerations**

You can generate the JCL for the GENJCL.IC and GENJCL.OIC commands in advance and not execute the JCL immediately. Therefore, the active set of DBDSs might change in the interim time period.

GENJCL no longer generates DD statements for the HALDB DBDS that is being copied because dynamic allocation is recommended for the Database Image Copy utility and the Database Image Copy 2 utility. The SYSIN control statement identifies which DBDS to copy: either the A-through-J or the M-through-V set of data sets. The image copy utility then copies the corresponding active set of data sets: A-through-J or M-through-V.

**NOTIFY.IC and NOTIFY.UIC Command Considerations**

You cannot take an image copy if a HALDB online reorganization cursor is active (OLREORG CURSOR ACTIVE =YES) or has been terminated. The NOTIFY.IC and NOTIFY.UIC commands fail if they attempt to record an image copy while the HALDB online reorganization cursor is active.

**Records Used in HALDB Online Reorganization**

HALDB online reorganization relies on a number of records to track information and activities related to online reorganization. This section includes descriptions of some of the records impacted by HALDB online reorganization:

- "DB Record Fields Used in HALDB Online Reorganization"
- "REORG Record Fields Used in HALDB Online Reorganization" on page 52

**DB Record Fields Used in HALDB Online Reorganization**

The DB record for a HALDB partition contains information for HALDB online reorganization. Complete descriptions of the fields in the DB record for a HALDB partition are in Table 49 on page 545. The DB record includes:

- A field that indicates whether the database can be reorganized online (ONLINE REORG CAPABLE=). This field is also in the HALDB master DB record. This field cannot be reset if:
  - Online reorganization is active
  - Any of the M-through-V and Y data sets are active
  - Any partition for the HALDB is authorized

If these conditions do not exist and the field can be reset, all of the M-through-V and Y data set information in the RECON data set is deleted.

- A field that indicates whether the HALDB online reorganization cursor is active (OLREORG CURSOR ACTIVE=).

- The SSID of the IMS that is currently performing a HALDB online reorganization (OLRIMSID=).

- A field that indicates which set or sets of DBDSs is or are active for the partition (ACTIVE DBDS=).

- Counters showing the number of cursor-active status changes (OLR ACTIVE HARD COUNT=) and cursor-inactive status changes (OLR INACTIVE HARD COUNT=) that are not yet hardened at the RSR tracking site.

**DBDS Record Definitions:** The M-through-V and Y DBDS records are defined in the RECON data set with the same attributes as the corresponding A-through-J, L, and X DBDS records. The M-through-V and Y DBDS records are added to the same...
CAGROUP as the A-through-J and X DBDS. Similarly, the M-through-V and Y DBDS records are added to each DBDS group where the A-through-J DBDS records are found.

The exceptions to the similarities in attributes include DD names, data set names, and the DSID. The DSID of the M-through-V and Y data set is the same as the corresponding A-through-J, L, and X data set, except that its X’80’ bit is set on. For the RECON data set list purposes, the X’80’ bit is ignored.

Table 4 shows the data set ID (DSID) DCB number associated with the A-through-J, L, and X data sets and the M-through-V and Y data sets.

<table>
<thead>
<tr>
<th>DSID DCB Number</th>
<th>A-J, L, and X Data Sets</th>
<th>M-V and Y Data Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PHIDAM</td>
<td>PHDAM</td>
</tr>
<tr>
<td>1</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>B</td>
</tr>
<tr>
<td>5</td>
<td>X</td>
<td>C</td>
</tr>
<tr>
<td>6</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>7</td>
<td>C</td>
<td>E</td>
</tr>
<tr>
<td>8</td>
<td>D</td>
<td>F</td>
</tr>
<tr>
<td>9</td>
<td>E</td>
<td>G</td>
</tr>
<tr>
<td>10</td>
<td>F</td>
<td>H</td>
</tr>
<tr>
<td>11</td>
<td>G</td>
<td>I</td>
</tr>
<tr>
<td>12</td>
<td>H</td>
<td>J</td>
</tr>
<tr>
<td>13</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>J</td>
<td></td>
</tr>
</tbody>
</table>

**Support for OLR Coexistence:** The DB record header also includes parameters that indicate if online reorganization is allowed. These parameters are OLRCAP or OLRNOCAP.

If online reorganization is enabled, sharing of HALDBs with IMS Version 7, or IMS Version 8 without the OLR Coexistence SPE applied, is no longer allowed. Full data sharing is allowed for IMS Version 8 with the OLR Coexistence SPE applied. IMS Version 7 or IMS Version 8 without the OLR Coexistence SPE applied are allowed to process databases that are marked as not capable of being reorganized online. You can set or reset this field using the CHANGE.DB DBD(HALDB_master) command, and display it in the output from a LIST.DB command or the appropriate Query API request.

Table 5 on page 52 shows whether full data sharing of databases is allowed between these IMS Versions:

- Version 7
- Version 8 (without the OLR Coexistence SPE applied)
- Version 8 (with the OLR Coexistence SPE applied)
- Version 9
DBRC Considerations for HALDB Online Reorganization

The ability to share data also depends on whether OLRCAP or OLRNOCAP is specified for a database, and Table 5 also identifies those cases.

Table 5. Data Sharing Between the Releases

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>OLRCAP</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>OLRNOCAP</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

REORG Record Fields Used in HALDB Online Reorganization

The REORG record for a HALDB partition DBDS, which is shown in Table 5 on page 558 contains a stop time for HALDB online reorganization. To run an offline reorganization of a database, you must take it offline by issuing a /DBR command. Therefore, only the start time of the online reorganization is significant. With HALDB online reorganization, the database is online, so the duration of the process is significant. The stop time of the online reorganization is recorded in the REORG record (STOP=timestamp), with information that indicates that it is an online reorganization and indicates whether the online reorganization can be used as input to recovery.

A REORG record is recorded under both the A-through-J, L, and X DBDS and the M-through-V and Y DBDS. The REORG record for the DBDS that was the output data set indicates that it can be used as input for recovery. The REORG record for the DBDS that was the input data set indicates that it cannot be used for recovery after online reorganization completes.

RSR Tracking System

Ownership of the HALDB online reorganization is not recorded in the tracking RECON data set.

The RECON data set at the tracking site always reflects the current status of the covered databases. Counters keep track of outstanding HALDB online reorganizations that have been recorded in the RECON data set at the tracking site but not yet applied to the databases.

Maintaining Recovery-Related Records

The A-through-J and M-through-V data sets are treated as one logical set of DBDSs. The M-through-V data sets are defined in the RECON data set with the same attributes as their corresponding A-through-J data sets.

The recovery information in the two DBDS headers is maintained as if they are one logical DBDS. For example, the values for GENMAX, RECOVPD, and CAGRP are the same. Any changes that are made to the active DBDS are also made to the inactive DBDS.

There is a one-to-one relationship between each partition data set; for example, A correlates to M, B correlates to N, C correlates to O, and so on. You can perform online reorganization on any of these data sets (A-through-J, L and X or M-through-V and Y) at any time, depending on which set is active before the online reorganization starts. Therefore, data sets A and M are one logical data set during an online reorganization. After an image copy is taken, DBRC removes any extraneous recovery-related records (IMAGE, ALLOC, RECOV, and REORG) from the RECON data set. The extraneous records are determined by the maximum
DBRC Considerations for HALDB Online Reorganization

number of recovery generations (GENMAX) settings and the optional recovery period (RECOVPD) settings for the DBDS.

For example, assume that the GENMAX value for data set A (and M) is 2, and that data set A has two image copies. When data set M becomes the active data set and an image copy is taken, the image copy cleanup process deletes the oldest image copy from data set A, including any of its extraneous recovery records. Another image copy of data set M causes the rest of data set A’s image copy and recovery records to be deleted. Note that data set A becomes unrecoverable if its image copy and recovery records are deleted.

The Online Reorganization Process

When online reorganization is started, the ID of the IMS that is responsible for the reorganization is assigned to the partition. After the cursor-active status is recorded in the RECON data set and copying begins between the A-through-J and M-through-V data sets, these changes also occur:

- The DB record for the partition indicates that online reorganization is active (OLREORG CURSOR ACTIVE = YES).
- The status of the active DBDSs shows that both data sets are active (ACTIVE DBDS=A-J and M-V).
- REORG records that show the start time of the online reorganization are written. The stop time is zero.
- The REORG record associated with the output data set now indicates that this data set can be used as input for recovery.
- The REORG record associated with the input data set does not indicate that this data set can be used for recovery. DBRC uses this record to ensure that this data set cannot be recovered after online reorganization completes.
- The M-through-V and Y data sets are defined in the RECON data set if they do not already exist.
- The M-through-V and Y DBDSs are added to the same change accumulation group as the A-through-J, L, and X DBDSs, if a change accumulation group is assigned.
- The M-through-V and Y DBDSs are added to any DBDS group where the A-through-J, L, and X DBDSs are found.

After these changes take place, the online reorganization must complete; otherwise, you must run an offline reorganization to reset the HALDB online reorganization settings.

You cannot start online reorganization if it is already running for a partition. However, another IMS can take over the online reorganization if it is terminated; that is, when there is no IMS ID assigned.

When the online reorganization process completes successfully, these changes occur:

- The REORG records are updated to include the HALDB OLR stop time.
- The DB record for the partition is changed to indicate that online reorganization is not active (OLREORG CURSOR ACTIVE = NO).
- The status of the active DBDSs is switched to the newly reorganized data sets.
- The online reorganization IMS ID is cleared.
Recommendation: You are not required to create image copies of a partition’s DBDSs after a successful online reorganization. However, it is recommended that you take an image copy as soon as possible to reduce recovery time.

Restriction: An image copy of a DBDS cannot be taken while the online reorganization cursor is active (OLREORG CURSOR ACTIVE = YES). DBRC causes the image copy to fail in this situation.
Chapter 4. Initializing and Maintaining the RECON Data Set

DBRC records recovery-related information in a pair of key sequenced data sets (KSDSs) called the REcovery CONtrol (RECON) data set. DBRC uses two RECON data sets to increase availability and recoverability. They contain identical information. The data sets are identified by the DD names RECON1 and RECON2.

If you define only two RECON data sets and an error occurs on one of them during operation, the current jobs continue using the remaining one. New jobs cannot start unless your RECON setting allows a job to start with only one active RECON data set. If you want to continue operations in dual mode, you can define a third RECON data set (RECON3). DBRC does not use this spare data set unless an error occurs on one of the two active RECON data sets. Then, DBRC copies the good RECON to the spare data set (RECON3), which then becomes active (thus maintaining RECON dual-mode operation).

Figure 9 shows the recommended three RECON data set operating configuration.

![Figure 9. DBRC Dual RECON Set Up with a Spare](image)

The RECON data sets are critical resources for both DBRC and IMS. If both RECON data sets are lost, DBRC abnormally terminates rather than compromising database integrity. IMS cannot continue processing transactions without a viable RECON data set, so IMS also abnormally terminates.

In This Chapter:
- “Planning Considerations for the RECON Data Set”
- “Initially Accessing the RECON Data Set” on page 62
- “Records in the RECON Data Set” on page 64
- “Maintaining the RECON Data Set” on page 73
- “Tracking Changes Made to the RECON Data Set” on page 81

Planning Considerations for the RECON Data Set

Each RECON data set is a VSAM KSDS with a 32-byte key.
Maintaining the RECON data set

Define your RECON data sets specifying VSAM SHAREOPTION(3,3). Ideally, each data set should be on a different device, channel, control unit, and catalog. The RECON data sets should also be of different sizes (see “Avoiding RECON Data Set Space Problems” on page 60).

Recommendations:
- When defining the RECON data sets using Access Method Services (AMS):
  - Use the same index control interval (CI) size and data CI size for all the RECON data sets.
  - Allocate at least twice as much space for an IMS V9 RECON data set as was allocated for an IMS Version 7 RECON data set.
- Ensure that the data CI size specified exceeds the specified index CI size by at least 2048 bytes. Failure to do so can seriously degrade your DBRC performance.
- Use secondary allocation to safeguard against a RECON data set becoming full.

Initializing the RECON Data Set for the First Time

After allocating the RECON data sets, use the INIT.RECON command to initialize the RECON data set. This command is only valid for an empty, uninitialized RECON data set. If the initialization job fails, delete and redefine the data sets before rerunning the job.

Avoiding RECON Data Set Contention Problems

Maximum RECON data set availability depends on the following:
- Eliminating deadlock
- Minimizing situations in which more than one RECON data set becomes simultaneously unavailable

For maximum availability, each RECON data set must:
- Have different space allocations. The spare data set must be at least as large as the largest RECON data set.
- Be on different devices.
- Be on different channels.
- Be in different user catalogs.

For additional information on avoiding space problems, see “Avoiding RECON Data Set Space Problems” on page 60.

Recommendation: To eliminate contention and facilitate recoverability, the RECON data sets should be the only type of data set on their respective devices if using implementing hardware reserves for the RECON data sets.

To eliminate deadlocks (if you are using implementing hardware reserves for the RECON data sets), the RECON data sets must:
- Be the only objects cataloged in their respective catalogs.
- Be on the same device as their catalogs. When the RECON data sets are accessed, an enqueue on the RECON data sets can result, followed by an enqueue on the catalog. When the RECON data sets and catalog are on the same device, the possibility of conflicts with another job enqueuing the devices in reverse order is eliminated.
Avoiding RECON Data Set Deadlocks

To avoid deadlock situations, give special consideration to the placement of RECON data sets that are shared among multiple processors. During a physical open, DBRC reserves RECON1, RECON2, and RECON3. DBRC determines which are available and which are Copy1, Copy2, and spare. DBRC then closes and dequeues the spare (if it exists) and any unusable RECON data sets. So, during the use of DBRC, two RECON data sets are reserved most of the time. DBRC always reserves both RECON data sets in this order: RECON1 and RECON2. If RECON1 and RECON2 are specified consistently throughout jobs, DBRC does not encounter deadlock.

However, other jobs that reserve multiple volumes can cause deadlock if any of the volumes also contain a RECON data set.

Related Reading: The z/OS MVS System Programming Library: Application Development Guide explains deadlock situations and volume assignments further; refer to it when placing your RECON data sets.

RECON Data Set Serialization

Access to the RECON data sets must be controlled to avoid deadlock. The following z/OS macros are used to control access to the RECON data sets: RESERVE, GRS, OBTAIN, and DEQ. DFP Record Management Services are also discussed and RECON data set serialization strategies are presented in the following sections.

RESERVE: DBRC issues the z/OS RESERVE macro to serialize access to each RECON data set. DBRC keeps the RECON data sets reserved until it completes its processing. The more RECON records DBRC must examine or change, the longer it holds the RECON data sets. The RESERVE macro serializes access to a resource (a data set on a shared DASD volume) by obtaining control of the volume on which the resource resides to prevent jobs on other systems from using any data set on the entire volume. This reserve is done under the major name DSPURI01 and has a scope of SYSTEMS.

Batch jobs will serialize on another resource name first, before issuing a RESERVE for the RECON data sets. The resource name for batch is DSPURI02 and has a scope of SYSTEM. Each job gets control of the resource, based on the position of the task’s request and whether the request was exclusive or share control. The queue is not ordered by the priority of tasks. The effect of this serialization of batch jobs is that an IMS online region never has to wait for more than one batch job to complete before gaining access to the RECON data set.

Global Resource Serialization (GRS): The GRS macro provides a method of converting a RESERVE request into an ENQ request. The ENQ, DEQ, and RESERVE macros identify a resource by its symbolic name. The symbolic name has three parts:

- Major name (qname)
- Minor name (rname)
- Scope (which can be STEP, SYSTEM, or SYSTEMS)

For example, on an ENQ or DEQ macro, a resource might have a symbolic name of APPL01,MASTER,SYSTEM. The major name (qname) is APPL01, the minor name (rname) is MASTER, and the scope is SYSTEM.
Maintaining the RECON data set

When an application uses the ENQ, DEQ, and RESERVE macros to serialize resources, global resource serialization uses resource name lists (RNLs) and the scope on the ENQ, DEQ, or RESERVE macro to determine whether a resource is a local resource or a global resource. Global resource serialization identifies each resource by its entire symbolic name. For example, a resource that is specified as A,B,SYSTEMS is considered a different resource from A,B,SYSTEM or A,B,STEP because the scope of each resource is different. To ensure that resources are treated as you want them to be without changes to your applications, global resource serialization provides three resource name lists (RNLs):

- **SYSTEMS EXCLUSION RNL**
  The SYSTEMS exclusion RNL contains a list of resources that are requested with a scope of SYSTEMS that you want global resource serialization to treat as local resources.

- **RESERVE CONVERSION RNL**
  The RESERVE conversion RNL contains a list of resources that are requested on RESERVE macros for which you want global resource serialization to suppress the RESERVE.

- **SYSTEM INCLUSION RNL**
  The SYSTEM inclusion RNL contains a list of resources that are requested with a scope of SYSTEM that you want global resource serialization to treat as global resources.

**Related Reading:** See the following publications for more information about GRS and the z/OS RESERVE, DEQ, and ENQ macros:

- z/OS MVS Programming: Authorized Assembler Services Reference, GC28-1763
- z/OS MVS Planning: Global Resource Serialization, GC28-1759

**OBTAIN:** DBRC uses a VSAM DADSM (Direct Access Device Storage Management) OBTAIN request to the FORMAT-4 DSCB (the VTOC) to force an I/O that insures that DBRC really has the RECON data set reserved in a multi-system environment. For more information about the OBTAIN DADSM macro, see the DFSMS/MVS® V1R5 MDFSMdfp Advanced Services, SC26-4921, manual.

The OBTAIN macro is not issued if the Synchronous RESERVE (SYNCHRES) option is in effect. The Synchronous RESERVE feature allows specification upon installation of whether the hardware RESERVE should be obtained for a device prior to granting a global resource serialization ENQ.

The SYNCHRES option can be activated through either the GRSCNFnxx parmlib member or the SETGRS operator command. The GRSCDEF statement of GRSCNFnxx contains the SYNCHRES (Y | N) parameter. The default value for SYNCHRES is NO. During normal system operation, the operator can modify the setting of SYNCHRES by issuing the SETGRS command. Activate SYNCHRES by issuing SETGRS SYNCHRES=YES and deactivate it by issuing SETGRS SYNCHRES=NO.

**DEQ:** DBRC releases the RECON data sets by using the z/OS DEQ macro.

**DFP Record Management Services:** DBRC uses VSAM services to retrieve, manipulate, and store the RECON data set records. These records have a 32-byte record key.

**RECON Data Set Serialization Strategies:** This section describes several serialization strategies, and the effects of implementing those strategies.
Recommendations:

- In a GRS Star configuration, a RESERVE CONVERSION RNL should be implemented for DSPURI01 if all systems accessing the RECON data sets are within the sysplex (or GRSplex).

Note: If the RECON data sets are accessed by systems that are outside of the sysplex, the reserve must not be converted. A SYSTEMS EXCLUSION RNL must be implemented instead.

- In a GRS Ring configuration, a SYSTEMS EXCLUSION RNL should be implemented for DSPURI01 so that the RECON is serialized by hardware reserve.

**GRS RESERVE CONVERSION RNL**

If you implement GRS RNL CONVERSION by adding the QNAME for the RECON data set, DSPURI01, to the conversion list, the hardware reserve is eliminated and replaced by a GRS enqueue that is communicated to all other sharing z/OS systems.

Other data sets on the same DASD volume can be used while the RECON data set is reserved; this is the benefit of performing the RNL conversion. GRS RNL conversion uses CPU and storage and affects system performance positively. The performance (in terms of least CPU time used, least storage used, and least elapsed time) is best using this option in a GRS STAR configuration.

To implement this method, follow these steps:

1. Add the RECON data set QNAME to the RESERVE CONVERSION RNL. For example:
   
   ```
   RNLDEF RNL(CON) TYPE(GENERIC) QNAME(DSPURI01)
   ```

2. Consider carefully the placement of the following VSAM QNAMEs: SYSZVVDS and SYSIGGV2.

**Related Reading**: For more information on implementing GRS RNL CONVERSION, see z/OS MVS Planning: Global Resource Serialization.

**GRS SYSTEMS EXCLUSION RNL**

If you implement GRS SYSTEMS EXCLUSION RNL, then GRS does not perform global serialization and the RESERVE macro is issued. This might work well provided that the RECON data set is located on a DASD volume that does not contain other data sets that are needed by other z/OS systems.

To implement this method, follow these steps:

1. Add the RECON data set QNAME to the SYSTEMS EXCLUSION RNL. For example:

   ```
   RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(DSPURI01)
   ```

2. Carefully consider the placement of the following VSAM QNAMEs; SYSZVVDS and SYSIGGV2.

**Related Reading**: The z/OS MVS Planning: Global Resource Serialization provides more information about VSAM QNAMEs.

**Related Reading**: There are a number of possible solutions to contention or performance problems associated with the RECON data sets.
Maintaining the RECON data set

- See the Informational APAR II10915 for information on RECON contention and performance problems.
- See the Informational APAR II10735 for information on performance and tuning diagnostics.

Allocating the RECON Data Sets

For both online and batch DBRC jobs, you can allocate the RECON1, RECON2, and RECON3 data sets with JCL, or you can let DBRC dynamically allocate them.

When dynamically allocating the RECON data set, omit the DD statements for RECON1, RECON2, and RECON3.

Use the TYPE=RECON macro statement of the IMS DFSMDA dynamic allocation macro to establish three dynamic allocation parameter lists in IMS.SDFSRESL. When multiple processors access the same RECON data sets, keep the IMS.SDFSRESL information pertaining to dynamic allocation parameters in synchronization on all processors.


DBRC always allocates RECON data sets with DISP=SHR.

Although JCL allocation and dynamic allocation are both valid methods for allocating RECON data sets, JCL allocation should be used only in a controlled test.

Recommendation: Use dynamic allocation for your production system and all other test or development environments.

The principal advantages of dynamically allocating the RECON data set are:
- All DBRC jobs automatically use the correct and current RECON data sets, and no JCL statements are left to become outdated.
- You can reorganize and restore RECON data sets, in case of error, without having to shut down online IMS systems.

Avoiding RECON Data Set Space Problems

Allocate the RECON data sets with different amounts of space so that if one becomes full, the system can continue using the other RECON data set while you provide a replacement. Allocate secondary extents for the RECON data set when you define space for it.

If one RECON data set becomes full during online operation, IMS deallocates it. DBRC responds by copying and reorganizing the good RECON data set to a spare RECON data set, if one is available. If no spare RECON data set is available, the system runs in single-RECON mode.

When all active subsystems have deallocated the failing RECON data set, you can delete and redefine it offline using AMS. See "Replacing a Discarded RECON Data Set" on page 78 for information on replacing a discarded RECON data set. If you are in single mode, and a spare RECON data set is available, the next time DBRC accesses the RECON data set, it automatically enters dual-RECON mode. You do not have to enter the CHANGE.RECON command with the DUAL or REPLACE option.

60 DBRC Guide and Reference
Creating a RECON Data Set

Create a RECON data set using the DEFINE CLUSTER command. The keywords that are recommended when defining a RECON VSAM KSDS are discussed in the following list. Information on all the keywords can be found in the z/OS DFSMS Access Method Services for Integrated Catalog Facility.

CONTROLINTERVALSIZE

The values used with this keyword affect the total amount of storage used by DBRC for VSAM and internal buffers. DBRC uses the Local Shared Resources (LSR) option of VSAM to process the RECON data sets. If the number of index and data buffers created by DBRC is allowed to default, the amount of storage used for RECON buffers is:

\[ 60 \times (\text{index}\_\text{ci}\_\text{size} + 120) \times (\text{data}\_\text{ci}\_\text{size}) \]

This amount of storage is used when the index and data CI sizes are the same for all RECON data sets. You can change the default number of index or data buffers used by DBRC in an online or batch environment with the DSPBUFFS Buffer Size Specification Facility.

DBRC uses one set of internal buffers for each RECON data set to build logical records from smaller segments. The size of these buffers depends initially upon the CI size or the VSAM maximum record size (LRECL), whichever is smaller. After the initial allocation of these buffers, the logical record buffers grow over time as the largest logical record on the RECON data set grows. Because only the initial allocation of buffers is determined by the VSAM data set definition, the storage allocated to internal buffers cannot ultimately be affected by changes to either the CI or the record size.

DBRC divides its own records into segments, each of which is always smaller than a single control interval and which is seen by VSAM as a complete physical record. VSAM record spanning is not used. Segmenting allows a logical RECON record to be as large as 16M bytes independent of the VSAM RECORDSIZE parameter.

Recommendations: Initially set your minimum CI size to a minimum of eight KB. The allowable CI size is affected by the value you select for RECORDSIZE. Also, ensure that the smallest data CI size exceeds the largest index CI size by at least 2048 bytes. Failure to do so can seriously degrade your DBRC performance.


CYLINDERS

Specifies the amount of space to allocate to the cluster.

FREESPACEx

The default values of FREESPACEx(0 0) must not be used. While you are entering initial information in the RECON data set, you must specify a high control-interval percentage (for example, 70%) as free space. Later, you can lower the percentage with an Access Method Services (AMS) ALTER command.

INDEXED

Specifies that the cluster is defined for key-sequenced data.

KEYSX

KEYSX(32 0) is required.
Maintaining the RECON data set

NAME
Defines the cluster’s entry name and is required.

NONSPANNED
See RECORDSIZE.

RECORDSIZE
DBRC always writes physical VSAM records that are less than the CI size in length, even though the logical RECON records can be a long as 16 MB.

Recommendation: Set the maximum record size to be, at most, equal to the CI size minus seven bytes and ensure that NONSPANNED is specified. For example, if the CI size was defined as 8192 bytes, then RECORDSIZE (4086,8185) is appropriate.

SHAREOPTIONS
SHAREOPTIONS(3 3) must be specified. The first value is required with single-host processors. Both values are required with multiple-host processors.

SPEED
This is recommended because the initial load is faster.

NOWRITECHECK
Avoid using WRITECHECK. It can degrade the RECON data set I/O performance. Using dual RECON data sets eliminates the need for WRITECHECK.

Recommendation: Do not use authorization keywords because frequent operator prompting results.

Security Considerations for the RECON Data Set
All jobs that access the RECON data sets must have control-level authority to the RECON data sets. Control, rather than update-level authority, must be specified, because VSAM issues VERIFY macros. VERIFY uses control interval processing.

Even jobs with read intent for databases using DBRC must have control-level authority because even a job with read intent updates information on the RECON header record.

Initially Accessing the RECON Data Set
When a job needs to read the RECON data set, the RECON data set must be opened. When three RECON data sets exist, DBRC determines which two are the active RECON data sets and which is the spare data set. Table 6 on page 63 shows how this determination is made. The columns in Table 6 on page 63 have the following meanings:

• In the DD Statement column, RECONA, RECONB, and RECONC represent the RECON1, RECON2, and RECON3 DD statements in no particular sequence.
• The Data Set Status column indicates the status of the data set during open time.
• The Data Set Use column indicates how DBRC assigns the data set.
### Table 6. Determining Which RECON Data Sets Are Accessed

<table>
<thead>
<tr>
<th>Case</th>
<th>DD Statement</th>
<th>Data Set Status</th>
<th>Data Set Use</th>
<th>DBRC Selection Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RECONA</td>
<td>Create Mode</td>
<td>Copy1</td>
<td>Not selected until INIT.RECON command is specified</td>
</tr>
<tr>
<td></td>
<td>RECONB</td>
<td>Create Mode</td>
<td>Copy2</td>
<td>Not selected until INIT.RECON command is specified</td>
</tr>
<tr>
<td></td>
<td>RECONC</td>
<td>Create Mode</td>
<td>Spare</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RECONA</td>
<td>RECON</td>
<td>Copy1</td>
<td>Current copy of RECON</td>
</tr>
<tr>
<td></td>
<td>RECONB</td>
<td>RECON</td>
<td>Copy2</td>
<td>Current copy of RECON</td>
</tr>
<tr>
<td></td>
<td>RECONC</td>
<td>Create Mode</td>
<td>Spare</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>RECONA</td>
<td>RECON</td>
<td>Copy1</td>
<td>Current copy of RECON</td>
</tr>
<tr>
<td></td>
<td>RECONB</td>
<td>RECON</td>
<td>Copy2</td>
<td>Current copy of RECON</td>
</tr>
<tr>
<td></td>
<td>RECONC</td>
<td>Create Mode</td>
<td>Spare</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>RECONA</td>
<td>RECON</td>
<td>Copy1</td>
<td>Current copy of RECON</td>
</tr>
<tr>
<td></td>
<td>RECONB</td>
<td>RECON</td>
<td>Unused</td>
<td>Older copy than RECONA</td>
</tr>
<tr>
<td></td>
<td>RECONC</td>
<td>Create Mode</td>
<td>Copy2</td>
<td>Produced from Copy1</td>
</tr>
<tr>
<td>5</td>
<td>RECONA</td>
<td>RECON</td>
<td>Copy1</td>
<td>Current copy of RECON</td>
</tr>
<tr>
<td></td>
<td>RECONB</td>
<td>RECON</td>
<td>Unused</td>
<td>Older copy than RECONA</td>
</tr>
<tr>
<td></td>
<td>RECONC</td>
<td>RECON</td>
<td>Unused</td>
<td>Older copy than RECONA</td>
</tr>
<tr>
<td>6</td>
<td>RECONA</td>
<td>RECON</td>
<td>Copy1</td>
<td>Current copy of RECON</td>
</tr>
<tr>
<td></td>
<td>RECONB</td>
<td>RECON</td>
<td>Copy2</td>
<td>Current copy of RECON</td>
</tr>
<tr>
<td></td>
<td>RECONC</td>
<td>RECON</td>
<td>Unused</td>
<td>Older copy than RECONA</td>
</tr>
<tr>
<td>7</td>
<td>RECONA</td>
<td>RECON</td>
<td>Copy1</td>
<td>Current copy of RECON</td>
</tr>
<tr>
<td></td>
<td>RECONB</td>
<td>Create Mode</td>
<td>Copy2</td>
<td>Produced from Copy1</td>
</tr>
<tr>
<td>8</td>
<td>RECONA</td>
<td>Create Mode</td>
<td>Copy1</td>
<td>Not selected until INIT.RECON command is specified</td>
</tr>
<tr>
<td></td>
<td>RECONB</td>
<td>Create Mode</td>
<td>Copy2</td>
<td>Not selected until INIT.RECON command is specified</td>
</tr>
<tr>
<td>9</td>
<td>RECONA</td>
<td>RECON</td>
<td>Copy1</td>
<td>Current copy of RECON</td>
</tr>
<tr>
<td></td>
<td>RECONB</td>
<td>RECON</td>
<td>Copy2</td>
<td>Current copy of RECON</td>
</tr>
<tr>
<td>10</td>
<td>RECONA</td>
<td>RECON</td>
<td>Copy1</td>
<td>Current copy of RECON</td>
</tr>
<tr>
<td></td>
<td>RECONB</td>
<td>RECON</td>
<td>Unused</td>
<td>Older copy than RECONA</td>
</tr>
<tr>
<td>11</td>
<td>RECONA</td>
<td>Create Mode</td>
<td>None</td>
<td>Discontinue processing</td>
</tr>
<tr>
<td>12</td>
<td>RECONA</td>
<td>RECON</td>
<td>Copy1</td>
<td>Current copy of RECON</td>
</tr>
</tbody>
</table>

**Case 4** is the situation where two RECON data sets are available, but one is now out of date. DBRC does not use the out-of-date RECON data set. Instead, it copies the up-to-date RECON data set to the spare data set.

Only one RECON data set is available in cases 5, 10, and 12. If you have specified the STARTNEW parameter of the INIT.RECON or CHANGE.RECON command, processing continues with one RECON data set. Otherwise, processing ends.
Records in the RECON Data Set

The RECON data set contains many types of records. Some records, such as header records, exist primarily to control processing of the RECON data set. Other records exist to define the various data sets used in the recovery of DBDSs. Still others exist to record events related to the use of DBDSs.

Figure 10 shows the major relationships between RECON record types. These relationships are described in detail in the sections that follow.

**RECON Header Records**

The RECON data set has two record types that contain control information DBRC uses in its processing:

- **RECON header**
  
  In addition to internal control information that DBRC creates and uses, the RECON header record contains information provided by you in the parameters of the INIT.RECON and CHANGE.RECON commands.

- **RECON header extension**
  
  The RECON header extension record contains RECON data set configuration and state data that DBRC uses to process the RECON data sets.

Related Reading:

- See Appendix B, “Sample Listings from the RECON Data Set,” on page 469 for details of what the records contain.
- See the IMS V9: Diagnosis Guide and Reference for a complete listing of RECON keys.
Log Data Set Records

Three types of log data sets exist. Each type can have a primary log record, secondary log record, interim-primary log record, and interim-secondary log record. Log data set records for each log data set are:

- Recovery log data set
  - PRILOG (primary log record)
  - SECLOG (secondary log record)
  - IPRI (interim primary log record)
  - ISEC (interim secondary log record)

- System log data set
  - PRISLD (primary system log data set)
  - SECSLD (secondary system log data set)
  - PRITSLDS (primary RSR tracking site log data set)
  - SECTSLDS (secondary RSR tracking site log data set)
  - IPRISL (interim primary system log data set)
  - ISECSL (interim secondary system log data set)
  - IPRITSLD (interim primary RSR tracking site log data set)
  - ISECTSLD (interim secondary RSR tracking site log data set)

- Online log data set
  - PRIOLD (primary online log data set)
  - SECOLDS (secondary online log data set)
  - IPIROL (interim primary online log data set)
  - ISECOL (interim secondary online log data set)

Log records come in sets called PRILOG families. A PRILOG family consists of a PRILOG and one or more of the following: SECLOG, PRISLD, and SECSLD for a given time period and IMS subsystem. All records in this set have the same start and end times and normally have matching data set entries. The same LOGALL record applies to all members of the set.

DBRC creates the PRILOG and PRISLD records whenever an online IMS opens the first OLDS, and updates them each time an OLDS is archived. If you use dual archiving, DBRC creates SECLOG and SECSLD records when the first OLDS is archived and updates them each time an OLDS is archived.

Related Reading: See “Archiving an OLDS” on page 8 for more information about DBRC and the archiving of online logs.

Log data sets output from IMS batch jobs are recorded in PRILOG / SECLOG records even though, technically, they are SLDSs. These records are created whenever the output log is opened and updated when volume switches occur.

In addition, during log recovery processing, DBRC creates an IPRISL or IPIROL record for each interim primary-log data set and an interim secondary-log record for each interim secondary-log data set whenever the Log Recovery utility runs. An interim log record is an internal record that is used to reflect intermediate processing of the DUP function of the Log Recovery utility.

Maintaining the RECON data set

Although not part of normal DBRC operation, you can use the following commands to create log records (for example, to set up a test environment or for RECON data set repair purposes):

- NOTIFY.PRILOG
- NOTIFY.SECLOG

Database Recovery Records

In addition to the log data set records, the following types of records contain database recovery information:

- “Backout Record (BACKOUT)”
- “Change Accumulation Group Record (CAGRP)”
- “Change Accumulation Run Record (CA)”
- “Data Group Record (DBG, DBDSGRP, RECOVGRP)” on page 67
- “Database Records (DB)” on page 67
  - “DL/I Database Records” on page 67
  - “HALDB Records” on page 68
  - “Fast Path Database Records” on page 70
- “Global Service Group Record (GSG)” on page 71
- “Image Copy Record (IMAGE)” on page 71
- “Reorganization Record (REORG)” on page 71
- “Log Allocation Record (LOGALL)” on page 72
- “Database Allocation Record (ALLOC)” on page 72
- “Recovery Record (RECOV)” on page 72
- “Subsystem Records (SSYS)” on page 72

Backout Record (BACKOUT)

A BACKOUT record contains information about units of recovery including time stamp, associated PSB name, recovery token, and database name. The names of nonrecoverable databases are not stored in the BACKOUT record.

Change Accumulation Group Record (CAGRP)

A CAGRP record identifies a change accumulation (CA) group. This record includes up to 32,767 names of DBDSs whose change records are accumulated during one run of the Database Change Accumulation utility. Each DBDS (for which DBRC is controlling recovery) can be a member of only one CA group in order for the Database Change Accumulation utility to accumulate its changes. You specify the CAGRP name in the INIT.CAGRP command that you use to create a CAGRP record in the RECON data set.

The CAGRP record contains the name of a member of a partitioned data set. This member contains the skeletal JCL that is to be used to generate the JCL to run the Database Change Accumulation utility for this CA group. The CAGRP record also contains an indicator that specifies whether change accumulation data sets that correspond to this group can be reused. It also contains an indication of the maximum number of change accumulation data sets that are to be maintained for the group.

Change Accumulation Run Record (CA)

For each CAGRP record, there can be up to 1024 CA records. A CA record contains information about a change accumulation data set and can be either available or in use.
An available CA record is created by an INIT.CA command. This CA record contains the volume serial numbers and the data set name of a data set that is to be used for output from a subsequent run of the Database Change Accumulation utility. You can create available CA records only for those CA groups that are defined with the REUSE parameter.

An in-use change accumulation record is created by a run of the Database Change Accumulation utility. It can be a formerly available CA record that was used during a run of the Database Change Accumulation utility, or it can be a new record with information obtained from the JCL. The information in an in-use change accumulation record includes:

- Data set name
- Volume serial numbers
- Run time of the Change Accumulation utility
- Stop time of the last log volume that the Database Change Accumulation utility processed; or, if the CA processed a subset of logs, the start time of the first log that should be processed on the next execution

You can use a NOTIFY.CA command to create a CA record.

**Data Group Record (DBGRP, DBDSGRP, RECOVGRP)**

You can use a DBDSGRP record to define the following types of named groups:

- DBDSGRP, a group of DBDSs and DEDB areas
- DBGRP, a group of DL/I databases and DEDB areas that can be named in the DATAGROUP parameter for the /STA, /ST0, and /DBR commands
- RECOVGRP, a group of DL/I databases and DEDB areas that are logically related for database recovery purposes

All groups must have unique names. For example, a DBDSGRP cannot have the same name as a DBGRP.

You use the INIT.DBDSGRP, CHANGE.DBDSGRP, DELETE.DBDSGRP, and LIST.DBDSGRP commands to manipulate all three data group types.

**Recommendation:** Although any type of group can be named in the DATAGROUP parameter for the /STA, /ST0, and /DBR commands, the use of a DBDSGRP is not recommended because it is inefficient.

**Related Reading:** See “DBDS Group Considerations” on page 48 for more information on DBDS groups.

**Database Records (DB)**

DBRC treats DL/I, Fast Path, and HALDB (High Availability Large Database) database records differently. These types of records, their treatment, and contents are explained in the following sections.

**DL/I Database Records:** A DB record identifies a database that is registered and whose recovery is under the control of DBRC. This record contains information about the database and related recovery information including:

- Database name
- Database type
- Share level of database
- List of subsystems using the database
- Extended Error Queue Element (EEQE) counter
Maintaining the RECON data set

- IRLM identification of the first subsystem that authorized the database (if IRLM is used)

A DBDS record identifies a DBDS whose recovery DBRC is to control. This record contains information about the DBDS (such as its data set organization) and related recovery information including:

- Name of the CA group to which the DBDS belongs
- Maximum number of image copy data sets to be maintained for this DBDS
- Indication of whether image copy data sets are to be reused
- Period of time that image copy data sets are to be maintained for this DBDS
- Name of the implied skeletal JCL default member
- Extended Error Queue Elements (EEQEs)
- Names of the members of the partitioned data set of skeletal JCL to be used in order to generate JCL for utilities that are run for this DBDS

To describe DL/I databases and DBDSs, DBRC maintains logically related records, as shown in Figure 11.

![Figure 11. DBRC DL/I Records](image)

**HALDB Records:** From a RECON data set perspective, HALDBs consist of a HALDB master (TYPE=HALDB) and one or more HALDB partitions (TYPE=PART).

[Figure 12 on page 69](image) shows the relationship of the RECON data set records that represent a HALDB.
HALDB Master (DSPDBHRC)

The RECON data set stores information pertaining to the entire HALDB by using a DB header record. The DB header record includes the following:

- HALDB master name
- TYPE=HALDB
- Partition Selection exit routine name
- Number of partitions in the HALDB
- Data Set Group count
- Current Partition ID
- Current change version number
- Global DMB number
- Share Level
- RSR Global Service Group Name and tracking level
- RECOVABL | NONRECOV
- OLRCAP | OLRNOCAP

The TYPE=HALDB DB record stores information about the HALDB. Applications conduct DB activity at the HALDB master level; DBRC conducts the DB activity at the HALDB partition level. A subsystem authorizes a HALDB partition, not a HALDB master.

HALDB Partition

Each partition of the HALDB consists of the following RECON data set records:

- Partition record (DSPPTNRC)

  DSPPTNRC contains information that applies to the individual partition. The HALDB Partition Definition utility displays the partition record information. The LIST command does not display the partition record.

Figure 12. Record Structure for a HALDB in the RECON Data Set
Maintaining the RECON data set

- Partition DB record (DSPDBHRC)
  DSPDBHRC accesses the HALDB at the partition level. Like the TYPE=IMS DB record, the DB record for the HALDB partition records all sharing and recovery information. The partition name sets the database name field in this record. TYPE=PART has been defined for this record. The following fields have the same settings for each partition across the entire HALDB:
  - Global DMB number
  - Share Level
  - RSR Global Service Group Name and tracking level
  - RECOVABL | NONRECOV
  - HALDB master name
  - OLRCAP | OLRNOCAP

- Partition DBDS records (DSPDSHRC)
  Depending on the organization of the HALDB, there can be three types of DBDSs for each HALDB partition: data, index, and ILDS data sets. Multiple data DBDSs can exist, but only one of each of the others. Only data DBDSs can be recovered. The other DBDSs are rebuilt using the HALDB Index/ILDS Rebuild utility (DFSPREC0).

  Related Reading: See IMS V9: Administration Guide: Database Manager for information about the data set and DDN naming conventions for DBDS records.

Fast Path Database Records: To describe DEDBs, AREAs, and AREA Data Sets (ADSs), DBRC has a logical structure of records, as shown in Figure 13.

![DBRC Fast Path Database Records Diagram]

Figure 13. DBRC Fast Path Database Records

DBRC uses the DB and DBDS records to describe both DL/I databases and DEDBs; however, DBRC adds an ADS list to the Fast Path DBDS record giving information about each ADS. Each DEDB may contain multiple areas, and each area may contain up to seven ADSs.

The Fast Path DB record contains information similar to the information in a DL/I DB record, except that it describes a DEDB; and it does not contain a list of authorized subsystems. For Fast Path, this list is in the DBDS record, which is composed of an area authorization record and an area recovery record. The Fast Path DB record is displayed in the listing as the DBDS record.

  Related Reading: See the sample listing in Appendix B, “Sample Listings from the RECON Data Set,” on page 469.
When an area is registered in the RECON data set, it ensures that:

- The area names in the DEDB are unique
- The ADS ddnames in an area are unique
- No more than seven ADSs are defined for an area

**Global Service Group Record (GSG)**

GSG record defines a global service group and the service groups that make up the GSG. An RSR service group is made up of two service groups: the active and the tracker.

GSG records are created by the INIT.SG command and can be deleted by the DELETE.SG command. The INIT.SG and DELETE.SG commands add and remove service group definitions to and from the GSG record. The CHANGE.SG command modifies information about a service group.

**Image Copy Record (IMAGE)**

An IMAGE record contains information about an image copy data set and can be either available or in-use.

An available IMAGE record is created by an INIT.IC command. It describes data sets that are to be used for output from a subsequent run of an image copy utility. You can create available IMAGE records only for those DBDSs that are defined with the REUSE parameter.

An in-use IMAGE record is created by an execution of an image copy utility:

- If the DBDS or area is defined with REUSE, it is a formerly available IMAGE record or a reused in-use record.
- If the DBDS or area is defined with NOREUSE, it is a new record with the data set description that is obtained from the JCL.

In addition to the data set description, the in-use IMAGE record identifies the type of copy operation and contains the copy operation’s run time and, depending on the type of copy, the copy operation’s stop time.

If you request that the image copy utility create an image copy data set and a duplicate image copy data set, DBRC records information about both in the same image copy record. The first record is designated IC1, and the duplicate is designated IC2.

If you create a nonstandard image copy data set (one that the image copy utility did not create), you must use a NOTIFY.UIC command to record its existence in the RECON data set.

**Related Command:** See “NOTIFY.UIC” on page 342 for more information on the IMAGE COPY record.

**Reorganization Record (REORG)**

DBRC creates a REORG record each time you run the HISAM Reorganization Reload utility, HD Reorganization Reload utility, HALDB Online Reorganization utility for a registered DBDS or partition. For the HALDB Online Reorganization utility, the REORG record indicates that the type of REORG is online, includes a stop time, and indicates if the reorganized data set can be used for recovery. DBRC creates a REORG record in the RECON data set for each DBDS that you reorganize.
Maintaining the RECON data set

Related Command: See “NOTIFY.REORG” on page 328 for more information on the REORG record.

Log Allocation Record (LOGALL)
DBRC creates a log allocation (LOGALL) record for each PRILOG record. A LOGALL record identifies the registered DBDSs that were changed while the corresponding log data set was open.

Database Allocation Record (ALLOC)
For Fast Path, DBRC creates an ALLOC record when the area is placed in OPEN-for-update status. For DL/I, DBRC creates an ALLOC record when IMS updates a DBDS the first time during a run of IMS or the first time after you enter a /DBRECOVERY command. This record contains the time stamp of its creation and the time stamp of the opening of the corresponding log. This time stamp identifies the log data sets containing the database change records that are needed for recovery. If the DBDS is subsequently deallocated, DBRC adds the time stamp of the deallocation to the ALLOC record.

DBRC automatically deletes allocation records if they are older than the oldest image copy record and if DBRC no longer needs them for database recovery. As DBRC deletes an ALLOC record, it changes the associated LOGALL record. This is part of the DBRC Image Copy utility exit routine processing. This automatic deletion of ALLOC records for a DBDS does not occur under either of the following conditions:

• The ALLOC record has no deallocation time.
  A deallocation time is recorded when the database or area has had the /DBR command run against it. Otherwise, the ALLOC record uses the log close time as an implicit deallocation time.
• The PRILOG record associated with the ALLOC record is open (it has a STOPTIME of zero) but is not marked in error.

In cases such as these, you can use the DELETE.ALLOC command to delete unwanted ALLOC records from the RECON data set. DBRC automatically updates allocation time stamps during online and concurrent image copy utility exit routine processing to move the allocation time stamps forward in time.

Related Command: See “DELETE.ALLOC” on page 202 and “NOTIFY.ALLOC” on page 305 for more information on the ALLOC record.

Recovery Record (RECOV)
DBRC creates a RECOV record each time you run the Database Recovery utility to recover a DBDS. The RECOV record can indicate one of these types of recovery:

• A full recovery of a DBDS where the RECOV record contains the time stamp of the recovery.
• A time-stamp recovery where the RECOV record contains the time stamp of the run of the Database Recovery utility and the time stamp to which the DBDS was recovered.

Related Command: See “NOTIFY.RECOV” on page 326 for more information on the RECOV record.

Subsystem Records (SSYS)
The RECON data set uses the subsystem (SSYS) record to describe data sharing information.
An SSYS record is created when an IMS subsystem signs on to DBRC. This SSYS record contains information about the subsystem and related recovery information including:

- Subsystem name and type (online or batch)
- IRLM identification
- Abnormal-end flag and the recovery-process start flag
- List of authorized databases
- Time stamp that correlates the subsystem entry with the appropriate log records

**Related Command:** See “NOTIFY.SUBSYS” on page 341 for more information on the SSYS record.

---

**Maintaining the RECON Data Set**

You should perform periodic maintenance on the RECON data set to maintain data integrity and an acceptable level of performance. Maintenance of the RECON data set falls into these categories:

- “Backing Up the RECON Data Set”
- “Deleting Unnecessary RECON Log Records”
- “Reorganizing the RECON Data Set” on page 75
- “Procedure for Reorganizing the RECON Data Set” on page 76
- “Replacing Damaged RECON Data Sets” on page 76
- “Recovering the RECON Data Set” on page 76
- “Replacing a Discarded RECON Data Set” on page 78
- “RECON Loss Notification” on page 78
- “Preserving the Integrity of the RECON Data Set with DBRC Command Authorization Support” on page 79

**Backing Up the RECON Data Set**

Back up the RECON data sets frequently. They are a critical resource. Always make backup copies of the RECON data set after performing any RECON record maintenance, such as registering databases and adding or deleting change accumulation groups.

Use the BACKUP.RECON command to perform backup. This command issues the necessary RESERVE commands (reserving the device during backup processing) to ensure backup integrity. Then it invokes the AMS REPRO command to copy the data set. The BACKUP.RECON command copies only the first copy of the RECON data set. Its parameters determine whether it makes one or two copies.

**Related Reading:** See Chapter 8, “BACKUP Command,” on page 131 for additional information on the BACKUP.RECON command.

**Deleting Unnecessary RECON Log Records**

You can delete unnecessary RECON records using the following methods:

- Automatic deletion by the RECON data set
- PRILOG compression
- Manual log record deletion
Maintaining the RECON data set

Automatic Deletion of Extraneous Records
Normally you should not need to perform much record maintenance for database-related records.

When the RECON data set is notified of an image copy, it may delete or reuse the oldest in-use IMAGE record, and a later IMAGE record becomes the oldest IMAGE record. RECOV and REORG records with start times earlier than the (now) oldest IMAGE record, and ALLOC records with DEALLOC times earlier than that, are now extraneous, and are deleted from the RECON data set. This is the image copy cleanup process.

At the same time that extraneous IMAGE records are deleted from the RECON data set, all active ALLOC records are updated to the time of the first log volume necessary for recovery, based on the oldest image copy for the DBDS or area. When the cleanup process deletes an extraneous ALLOC record it changes the state of the associated LOGALL records. Once all the ALLOC records associated with the LOGALL record have been deleted (this may take place over many image copies for many databases), the PRILOG record associated with the LOGALL record becomes inactive.

Compressing the PRILOG Record
PRILOG record compression is the deletion of all inactive data set entries in the PRILOG record. A data set entry is defined as being inactive when it is older than all of the following criteria:

- Log retention period
- Oldest allocation (ALLOC) for any database updated on that log
- Earliest restart checkpoint for the online IMS

PRILOG record compression deletes inactive data set entries up to the oldest ALLOC on the log or the first gap in the data set entries. A gap occurs when an OLDS has not yet been archived.

When inactive data set entries are deleted from active PRILOGs, they are compressed to a single dummy data set entry that has the same start time as the start time of the log and the same stop time as the stop time of the last inactive data set entry deleted.

Automatic PRILOG Compression
PRILOG record compression is attempted automatically after an OLDS has been archived. At an RSR (remote site recovery) tracking site, automatic compression is attempted when a tracking log data set is opened by Log Router and recorded in the RECON data set.

Compression includes the deletion of all inactive data set entries in the PRILOG record. When applicable, corresponding entries in the SECLOG, PRISLD, and SECSLD records are also deleted.

Manual Process for PRILOG Compression
You can initiate PRILOG record compression manually by using the DELETE.LOG INACTIVE command. This command deletes inactive data set entries from active PRILOG records and deletes entire inactive PRILOG records.

Deleting Log Records
DBRC does not automatically delete RECON records that describe log data sets (PRILOG and SECLOG records). This design gives you control over which RECON
records associated with log data sets are deleted. You must periodically delete PRILOG and SECLOG records that are no longer needed for recovery.

Use the DELETE.LOG INACTIVE command to delete inactive PRILOG and SECLOG records.

Deleting log records does not prevent the RECON data sets from filling up because the space freed by deletions might not be reused by VSAM. However, if you delete the log records before backing up or reorganizing the RECON data set, you are able to reclaim the space during backup or reorganization.

Related Reading:
- See "DELETE.LOG (for OLDS)" on page 208 for more information on LIST.LOG.
- See "LIST.LOG (for a PRILOG Family)" on page 296 for more information on DELETE.LOG.

Reorganizing the RECON Data Set

You need to reorganize the RECON data sets periodically. Many of the record keys in the RECON data set include the date and time. DBRC recording of IMS log and database activity can cause CI and CA splits, that can degrade performance. In addition, deleting unnecessary records might not prevent the RECON data set from filling up because VSAM does not always reuse the space freed.

Using CHANGE.RECON to Reorganize a RECON Data Set

You can reorganize a RECON data set online if you are using dynamic allocation for the RECON data set. A spare RECON data set must be available. In this situation, you can issue the CHANGE.RECON command with the REPLACE option. This causes DBRC to copy and reorganize the active RECON data set (specified on the CHANGE command) to the spare data set. VSAM removes all CI and CA splits, and restores the original FREESPACE attributes.

The CHANGE command also deallocates the old RECON data set (the one that needed reorganization). Before you can delete and redefine this data set, however, you must wait for all other subsystems that are using it to deallocate it. If you redefine the data set with the same name it originally had, it is available to the online system for use as a spare data set. You can repeat this process to reorganize the second active RECON data set. If you use JCL in order to allocate data sets, dynamic deallocation does not occur.

If you do not use dynamic allocation or if a spare RECON data set is not available, you must wait for the online subsystem and all other subsystems that access the RECON data set to deallocate it before you can reorganize.

Backing up RECON Data Sets before Reorganization

It is recommended that you back up the RECON data sets before and after reorganizing, using the following procedure:

1. Copy the RECON data sets to temporary data sets with different data set names.
2. Verify that the copied data sets are uncorrupted.
3. Delete the original data sets.
4. Redefine the original data sets using the same data set names.
5. Copy the temporary data sets back to these original data sets.
Maintaining the RECON data set

Procedure for Reorganizing the RECON Data Set
To clean up your logs and reorganize your RECON data sets, follow this procedure. This process assumes that you are running DBRC in dual mode with RECON1 (as Copy1), RECON2 (as Copy2), and a spare.

1. Issue a LIST,LOG OPEN command to identify any open logs. Determine which of the identified logs should be open, and which should be closed or deleted. Close any logs that should be closed or deleted.
2. Issue a DELETE,LOG INACTIVE command to delete any inactive or unused logs. You are now ready for the RECON data set reorganization.
3. Issue a CHANGE,RECON with the REPLACE option for RECON1. DBRC then:
   a. Signals the start of the RECON data set reconfiguration with message DSP0380I
   b. Identifies the subsystems that are active at reconfiguration with message DSP0388I
   c. Signals that the process has completed with message DSP0381I.
4. Replace the discarded RECON data set. See “Replacing a Discarded RECON Data Set” on page 78 for more information.

Your RECON data sets have now been reorganized and the logs are now clean.

Replacing Damaged RECON Data Sets
If an I/O error occurs in the current RECON data set that is the only one remaining, DBRC stops the job. Any other jobs that are currently using the RECON data set continue to run if no other I/O error is encountered.

If an I/O error occurs in a RECON data set and two RECON data sets exist, DBRC attempts to locate a spare data set. If a spare is available, DBRC copies the RECON data set without the I/O error to the spare RECON data set. DBRC then establishes the spare as the Copy2 RECON data set.

Recommendation: After the spare RECON data set replaces the RECON data set that had the error, redefine the discarded RECON data set as quickly as possible. If you immediately replace the RECON with the I/O error, you are unlikely to experience a subsystem failure due to loss of all RECON data sets. See “Replacing a Discarded RECON Data Set” on page 78 for more information.

If DBRC cannot locate a spare RECON data set and you have specified the STARTNEW parameter of the INIT,RECON command, DBRC continues processing with one RECON data set. Otherwise, DBRC completes the current job but does not start new jobs until you define a spare RECON data set.

Recovering the RECON Data Set
Steps for recovering your RECON data set depend on the set up of your RECON data set and the situation. The following paragraphs discuss various solutions for recovering a RECON data set.

A Spare RECON data set is available: If an I/O error occurs on a RECON and a spare data set is available, DBRC copies the good RECON to the spare, and then activates the spare.

If, however, you want to analyze the RECON error, before deleting and redefining the discarded RECON data set, make a copy of it for later problem diagnosis.
A Spare RECON data set is not available: If a spare RECON is not available, all currently executing jobs continue processing using the RECON data set in single mode. If you specified the STARTNEW parameter in the INIT.RECON or CHANGE.RECON command, DBRC allows new jobs to start with only one RECON data set. This is not recommended as it jeopardizes the integrity of the system.

If one of the data sets in the set of RECON data sets becomes unusable by DBRC, you need to deallocate the RECON data set that is unusable and allocate a new spare.

In an RSR environment, the isolated log sender (ILS) starts its own copy of DBRC. If Automatic RECON Loss Notification is not active, you might need to stop ILS to terminate the DBRC in the transport manager address space. This causes the ILS’s DBRC to deallocate the RECON data sets so that you can replace the unusable RECON data set. Issue a STOP ILS(95g) command for each started ILS instance. Then issue START ILS(95g) to bring up ILS and DBRC again.

Both RECON data sets are not usable:

It is unlikely that both RECON data sets would be not usable. If, however, both RECON data sets ever become unusable, follow this procedure:
1. Stop all jobs that require access to the RECON data set.
2. Use the AMS REPR0 command to back up the RECON data sets if you can access both of them. This step is optional, but it is recommended.
3. Use the AMS utility to delete and redefine your RECON data sets.
4. Use the AMS REPR0 command to restore one of the RECON data sets.
5. Use the AMS REPR0 command to restore the other RECON data set from the first.
6. Use the LIST.RECON command to list one of the RECON data sets. Evaluate the list and determine which DBDSs have been updated since you made the backup in step 2. If you cannot determine which DBDSs have been updated, assume that all have been updated.
7. Use the CHANGE.IC command with the INVALID parameter to mark all image copy records that are in error for all applicable DBDSs in step 6.
8. Make an image copy of all applicable DBDSs in step 6.
9. Use the BACKUP.RECON command to make a backup copy of the RECON data sets.

The RECON data sets are now restored and resynchronized with the databases.

If you do not control an excessive number of databases, it might be easier to follow this procedure:
1. Stop all jobs that require access to the RECON data set.
2. Define new RECON data sets.
3. Initialize these RECON data sets.
4. Register the environment. Always keep a backup copy of the most recently initialized, but not-yet-used, RECON data set available.
5. Take image copies of all databases.

Finally, before you proceed with your regular operations, clean up the new RECON data set by, for example, closing any open, out-of-date OLDSs with the NOTIFY.PRILOG command.
Maintaining the RECON data set

Replacing a Discarded RECON Data Set

DBRC detects that a RECON data set is discarded only when some activity occurs that causes DBRC to access the RECON data sets. You can have multiple instances of DBRC whenever you have multiple IMS subsystems, online or batch. You cannot delete and redefine a discarded RECON data set until all instances of DBRC detect that a change has occurred and they deallocate the discarded data set. DBRC lists the subsystems that are active at reconfiguration in message DSP0388I. This message enables you to identify the subsystems that might need your help in detecting the status change of the RECON data sets.

To redefine a RECON data set after an I/O error has occurred, or in conjunction with the CHANGE.RECON REPLACE command, follow this procedure:

1. Allow all batch jobs using DBRC to finish.
2. Issue LIST.RECON STATUS in all online subsystems if you do not have Automatic Loss Notification active. Issuing the command causes the online subsystems to obtain the same Copy1 and Copy2 RECONs and to deallocate the discarded RECON data set. If you do have Automatic Loss Notification active, all subsystems are automatically notified to deallocate the discarded RECON data set. See "RECON Loss Notification" on page 78 for more information on RECON Loss Notification.
3. Use the AMS DELETE command to delete the discarded RECON data set.
4. Use the AMS DEFINE command to recreate the RECON data set as an empty VSAM KSDS data set. Use the same procedure that you used originally to create the RECON data set. See "Creating a RECON Data Set" on page 61.

RECON Loss Notification

All DBRC instances that are allocated to the RECON, using the same IMSplex, are notified of errors, at the time the errors occur, through the IMS Structured Call Interface (SCI). Then, any DBRC allocated to the discarded RECON will deallocate.

DBRC obtains the IMSplex name from the SCI Registration exit routine (DSPSCIX0), or from an EXEC parameter, IMSPLEX. DBRC records the IMSplex name in the RECON data set when the first use of RECON Loss Notification occurs for a given RECON data set. Any subsequent attempt at accessing a RECON data set using a different IMSplex, or no IMSplex, is rejected and message DSP1136I is issued. Otherwise, RECON Loss Notification is automatically invoked.

If you want to change the IMSplex associated with a set of RECONs, use the IMSPLEX() or NOPLEX parameter on the CHANGE.RECON command. See "CHANGE.RECON" on page 177 for more information. To change an IMSplex/RECON association when a DBRC instance is active, perform the following procedure to avoid having your command rejected and message DSP1137I issued:

1. Wait for all DBRC activity on the current IMSplex to cease.
2. Submit a Database Recovery Control utility job to change the IMSplex name.
3. Alter the IMSplex name in DSPSCIX0.
4. Ensure that the new IMSplex SCI is ready.
5. Resume DBRC activity on the new IMSplex.

Related Reading:
- See the IMS V9: Common Service Layer Guide and Reference for introductory information about IMSplexes.
Preserving the Integrity of the RECON Data Set with DBRC Command Authorization Support

The RECON data sets are critical to the integrity of IMS databases. Therefore, you should restrict access (control authorization) to the DBRC commands. Consider providing access to only a subset of the DBRC commands for those users who must issue them.

In the online region, IMS provides some authorization functions for commands issued. By default, IMS restricts /Rmxxxxxx commands (except for /RMLIST) to the Master Terminal Operator (MTO).

You can establish authorization control for DBRC commands through RACF® (or an equivalent product), a user exit routine, or both. For example, logon ID SMITH could have access to issue an INIT.DB command against DBD PAYROLL but not against DBD CUSTOMER.

**Related Reading:** For more information about using exit routines and RACF for command authorization, see:

- “Authorizing Commands with RACF” on page 80
- “Authorizing Commands with the DBRC Command Authorization Exit Routine (DSPDCAX0)” on page 80
- “Authorizing Commands with Both the DBRC Command Authorization Exit Routine (DSPDCAX0) and RACF” on page 80

The HALDB Partition Definition utility is an ISPF application that allows you to manage the definitions of IMS HALDBs and their partitions in the RECON data set, providing functionality equivalent to the following:

- INIT.DB
- INIT.PART
- CHANGE.DB
- CHANGE.PART
- CHANGE.DBDS
- DELETE.DB
- DELETE.PART
- LIST.DB

Authorization for the HALDB Partition Definition utility is controlled using the same resources as defined for these commands with one exception; the CHANGE.PART resource is used in place of the CHANGE.DBDS resource.

**Related Reading:** See the IMS V9: Utilities Reference: Database and Transaction Manager, SC18-7833 for more information about the HALDB Partition Definition utility.

Chapter 4. Initializing and Maintaining the RECON Data Set
Authorizing Commands with RACF

You can set up DBRC command authorization support with RACF by defining resource profiles (explicit or generic) that cover all of the DBRC commands and permitting appropriate user access to these profiles. The resource name corresponds directly to the command and consists of a high-level qualifier (HLQ) and up to three elements of the DBRC command. The resource name model has the following format:

- High Level Qualifier (for example, SAFHLQ)
- Command verb (for example NOTIFY)
- Command modifier (for example, PRILOG or IC)
- Command qualifier (for example, dbname or OLDS)


Different profiles can be used based on the set of RECON data sets being used, but only one HLQ name per RECON data set is allowed. For example, you might use an HLQ name of PRODRECON for the production RECON data sets and an HLQ name of TESTRECON for the test RECON data sets.

Other considerations for your definitions are as follows:
- Resource profiles protecting the DBRC commands must be defined in the FACILITY resource class
- User identifiers found in the ACEEUSRI field of the ACEE are used for all authorization checking and the user must be permitted READ access to the resource profile at a minimum
- DBRC command authorization is enabled or disabled using the CHANGE.RECON and INIT.RECON command with the CMDAUTH keyword. The HLQ name is required when enabling command authorization.

Authorizing Commands with the DBRC Command Authorization Exit Routine (DSPDCAX0)

You can use the DBRC Command Authorization exit routine (DSPDCAX0) to verify that a user is authorized to issue a particular command. If you decide to use this exit, you will make the appropriate selection when using the INIT.RECON or CHANGE.RECON commands. Refer to "INIT.RECON" on page 280 and "CHANGE.RECON" on page 177 for more information.

DSPDCAX0 is a required exit routine if it is specified to authorize DBRC commands (that is, if either sub-parameter EXIT or BOTH is specified in the CMDAUTH keyword on the INIT.RECON or CHANGE.RECON commands). DSPDCAX0 must be found in an authorized library or in LINKLST. If DSPDCAX0 is found in a concatenated STEPLIB or JOBLIB, only the data set containing DSPDCAX0 must be authorized. If DSPDCAX0 is found in LINKLST, no authorization check is performed.

Related Reading: See IMS V9: Customization Guide for more information on the DBRC Command Authorization exit (DSPDCAX0).

Authorizing Commands with Both the DBRC Command Authorization Exit Routine (DSPDCAX0) and RACF

The DBRC command authorization exit routine (DSPDCAX0) can be used in conjunction with RACF to provide command authorization. In such cases, the security product is invoked first. The return and reason codes are passed to the
exit routine. The return code that the exit routine issues ultimately determines the
success or failure of the command authorization; the exit routine overrides the
outcome of the security product. Therefore, DBRC messages that are put out as a
result of unsuccessfully invoking the security product will be suppressed.

Again, if EXIT or BOTH is specified in the CMDAUTH keyword on the INIT.RECON
or CHANGE.RECON commands, DSPDCAX0 is a required exit routine. Also,
DSPDCAX0 must be found in an authorized library or in LINKLST. If DSPDCAX0
is found in a concatenated STEPLIB or JOBLIB, only the data set containing
DSPDCAX0 must be authorized. If DSPDCAX0 is found in LINKLST, no
authorization check is performed.

Related Reading: See [IMS V9: Customization Guide] for more information on the
DBRC Command Authorization exit (DSPDCAX0).

Tracking Changes Made to the RECON Data Set

For administrative purposes, it is a good practice to keep a log of the changes
made to the RECON data set. You can provide your own exit routine which is to
be called every time RECON records are changed or read. This exit routine, also
called the RECON I/O exit routine (DSPCEXT0), allows you to keep track of
changes to RECON in the form of a journal.

Related Reading: See [IMS V9: Customization Guide] for detailed information about
the RECON I/O exit routine.
Chapter 5. Skeletal JCL

The following sections discuss various aspects of working with skeletal JCL.

In This Chapter:
- “Generating JCL and User-Defined Output”
- “Using IBM-Supplied Skeletal JCL” on page 84
- “Writing Your Own Skeletal JCL” on page 85
- “Contents of the Skeletal JCL Data Set” on page 85
- “Skeletal JCL Syntax” on page 86

Generating JCL and User-Defined Output

Eight GENJCL commands are supported by the Database Recovery Control utility. Seven of these commands generate the JCL and control statements necessary to run various IMS recovery-related utilities. You can use the remaining command, GENJCL.USER, to generate user-defined output, including JCL.

Table 7 lists the eight GENJCL commands and what they generate.

<table>
<thead>
<tr>
<th>Command (PDS Member)</th>
<th>What the Command Generates</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENJCL.ARCHIVE (ARCHJCL member)</td>
<td>Log Archive utility JCL and control statements</td>
</tr>
<tr>
<td>GENJCL.CA (CAJCL member)</td>
<td>Database Change Accumulation utility JCL and control statements</td>
</tr>
<tr>
<td>GENJCL.CLOSE (LOGCLJCL member)</td>
<td>Log Recovery utility JCL and control statements</td>
</tr>
<tr>
<td>GENJCL.IC (ICJCL member)</td>
<td>Database Image Copy or Database Image Copy 2 utility JCL and control statements</td>
</tr>
<tr>
<td>GENJCL.OIC (OICJCL member)</td>
<td>Online Database Image Copy utility JCL and control statements</td>
</tr>
<tr>
<td>GENJCL.RECEIVE (ICRCVJCL member)</td>
<td>Database Recovery utility JCL and control statements</td>
</tr>
<tr>
<td>GENJCL.RECOV (RECOVJCL member)</td>
<td>Database Recovery utility JCL and control statements</td>
</tr>
<tr>
<td>GENJCL.USER (DSPUPJCL member)</td>
<td>User-defined output, including JCL and control statements</td>
</tr>
</tbody>
</table>

Note: Sample JCL for the HALDB INDEX/ILDS Rebuild utility (DSPUPJCL) is shipped with IMS and can be used with the GENJCL.USER command, but other types of JCL can be used as well. No default is defined.

When you issue a GENJCL command, it uses a skeletal JCL execution member. The execution member is a model of the JCL or user output that you are producing. The execution member contains symbolic keywords. DBRC substitutes current information for the symbolic keywords. The substituted information comes from the RECON data set and from skeletal JCL default members, and from your USERKEY values. Typical of the information DBRC substitutes for symbolic
Using IBM-Supplied JCL

If you are generating JCL for the IMS recovery-related utilities using the IBM-supplied skeletal JCL execution members, the process is simple. It involves modifying the IBM-supplied skeletal JCL execution members. For the skeletal JCL execution members, see "IBM-Supplied Skeletal JCL Execution Members" on page 439. Before you use them:

- Add two DD statements (JCLPDS and JCLOUT) to the DBRC dependent address space procedure used for online IMS. JCLPDS identifies the partitioned data set containing the skeletal JCL execution members. JCLOUT identifies the data set to which the generated job is to be written. Output is in card image format. The output data set can be a punch file, a DASD data set that you plan to examine before submitting the job for execution, or directly to the z/OS internal reader. These two ddnames can be specified on the GENJCL command. When GENJCL is used, the two specified data sets are in effect for the GENJCL command only, and not for the life of the job. The JCLOUT data set is opened at the start of the command execution and closed at the end of the command execution. Consequently, if multiple GENJCL commands are concatenated in the job stream, the JCLOUT data set (if other than the z/OS internal reader) only contains the results from the last command that was processed.

- Add any STEPLIB and STEPCAT DD names, and job accounting information that your installation requires to the skeletal JCL execution member. Except for the skeletal JCL member for the JOB statement, do not add any JOBCAT, JOBLIB, and JES control statements to your skeletal JCL; doing so causes errors if multiple steps are generated.

- Change the default value for the REGION parameter on the skeletal JCL EXEC statement if the existing one is not correct for your installation.

- If you plan to generate JCL to run the Log Recovery utility (member LOGCLJCL), replace the DFSWADSO DD statement.

**Recommendation:** Exercise care when modifying the skeletal JCL, because DBRC does not verify any of the JCL that is generated.
Writing Your Own Skeletal JCL

The following sections describe the things you need to know before writing your own skeletal JCL or developing symbolic keywords to modify the IBM-supplied skeletal JCL. You must write your own skeletal JCL or symbolic keywords if the IBM-supplied skeletal JCL execution members do not meet your requirements or if you plan to use the GENJCL.USER command. IBM provides no skeletal JCL execution member for the GENJCL.USER command.

Contents of the Skeletal JCL Data Set

As shown in Figure 14 on page 86 the skeletal JCL data set contains the skeletal JCL members used by the GENJCL command processor to generate output. The two types of skeletal JCL members are execution members and default members.

Execution members are models of the output you are generating. Execution members can be IBM supplied (as described in “Using IBM-Supplied Skeletal JCL” on page 84) or supplied by you. Execution members contain symbolic keywords, which represent information DBRC provides.

Default members specify default values for symbolic keywords in the execution members. The use of default members is optional. You provide the default members. To use a default member, you specify the member on the GENJCL command. Or, in the case of DBDS and CA groups, you can implicitly specify the default member (see “Specifying Default Members” on page 104 for more information).
Skeletal JCL Syntax

Skeletal JCL execution members are models of the output to be produced. The execution members contain symbolic keywords and control keywords. Symbolic keywords are symbolic parameters that DBRC substitutes with a value in the output stream (for example, data set names), based upon what is specified on a GENJCL command and information obtained from the RECON data set.

Control keywords regulate what output is generated by DBRC, based upon what is specified on a GENJCL command (for example, what RECON records are to be used for keyword substitution).

The keywords in skeletal JCL members must be in uppercase.

Simple keywords and control keywords are explained in the following sections:

- “Symbolic Keywords”
- “Using Control Keywords” on page 88
- “Writing Control Keywords” on page 94
- “Understanding Skeletal JCL Default Members” on page 103

Symbolic Keywords

When JCL is generated, symbolic keywords in the skeletal JCL execution members are replaced with the current keyword value. For example, the IBM-supplied...
skeletal JCL execution members use `%TIME` as a symbolic keyword. When DBRC encounters `%TIME`, it replaces it with the time of day. Keyword substitution occurs each time DBRC encounters a symbolic keyword. Multiple symbolic keywords can exist in a skeletal JCL execution member.

Symbolic keywords must be assigned a value before you use them. Keyword values are assigned (or set) in several different ways as specified below:

- The `GENJCL` command specifies values for some of the symbolic keywords in skeletal JCL execution or default members. User-defined keywords are assigned a value in the `USERKEYS` parameter in the command. Other keyword values are set by various parameters on the command. For example, the `SSID` parameter sets the value for the `%SSID` keyword (the subsystem ID).
- Skeletal JCL default members set default values for keywords in skeletal JCL execution members.
- The `RECON` data set also provides keyword values. For example, when the `GENJCL.ARCHIVE` command is issued, the ddnames and data set names for the `OLDS` are obtained from the `PRIOLDS` and `SECOLDS` records.
- Some keyword values are implicitly known, for example the time of day.

If during the JCL generation process, a keyword is encountered that has not been assigned a value, no substitution takes place. Instead, DBRC issues a warning message.

When writing your own skeletal JCL execution members, you can define your own symbolic keywords as well as use the symbolic keywords already recognized by DBRC. For a list of the symbolic keywords that DBRC recognizes, see Appendix A, “Symbolic Keywords Recognized by DBRC,” on page 431. You can also define your own symbolic keywords and add them to the IBM-supplied skeletal JCL execution members.

Here are some conventions, restrictions, and other detail you should know when writing symbolic keywords:

- Keywords must begin with a percent (%) sign.
- The minimum keyword length is two characters, including the percent sign. The maximum length is eight characters, including the percent sign.
- Keywords must be written using uppercase letters only (A rather than a).
- The first character after the percent sign must be alphabetic (A-Z); the remaining characters must be alphanumeric (A-Z, 0-9). Keywords are delimited by a non-alphanumeric character or when the maximum length is reached.
- DBRC does not use any keywords beginning with `%W, %X, %Y,` or `%Z. You can, therefore, use these characters for your own keywords without conflicting with predefined keywords.
- User-defined symbolic keywords must be assigned a value with the `USERKEYS` parameter on the `GENJCL` command or with a skeletal JCL default member.
- Keyword substitution is performed on columns 1-71 of the skeletal JCL records. Columns 72-80 are not modified. If the keyword value is shorter than the keyword, the remaining data on the record is shifted to the left and filled with blanks. If the keyword value is longer than the keyword, the remaining data is shifted to the right. If any non-blank characters are shifted beyond column 71, a JCL continuation statement is generated. In some cases (for example, when the output is not a JCL statement), it might not be possible to generate a JCL.
continuation statement, because a comma or blank must exist in the output record for DBRC to split it. When DBRC cannot find a break in the statement, it splits the statement at column 71.

**Using Control Keywords**

Use the following control keywords to regulate what JCL (or other output) is generated:

- **%SELECT and %ENDSEL**
  The %SELECT keyword selects the RECON records that are needed in order to resolve symbolic keywords. The %ENDSEL keyword indicates the end of the records selected by the %SELECT keyword. These control keywords always occur in pairs. A %SELECT keyword is followed by one or more execution member records, which is followed by the %ENDSEL keyword. This sequence of records is called a control group or, more specifically, a select group.

- **%DELETE and %ENDDEL**
  The %DELETE keyword deletes records from the generated output stream. Deletion occurs based on a specific condition. The %ENDDEL keyword delimits the scope of the %DELETE keyword. These control keywords always occur in pairs. A %DELETE keyword is followed by one or more execution member records, which is followed by the %ENDDEL keyword. This sequence of records is called a control group or, more specifically, a delete group.

- **%SET MEMBER**
  The %SET MEMBER keyword specifies a different skeletal JCL execution member that is to be used in the next step of a multistep job.

- **%SET TIMEFMT**
  The %SET TIMEFMT keyword is used to specify a form for time stamps that appear in GENJCL output.

**Using the %SELECT and %ENDSEL Keywords**

Use the %SELECT keyword to select one or more records from the RECON data set. The selected records identify IMS data sets or events tracked by DBRC. Information from the selected records is used to resolve symbolic keywords in the select group. Symbolic keywords can occur in the execution member records or in the %SELECT keyword statement.

The format for a select group follows:

```
%SELECT record_type(selection_criteria)
   execution_member_record(s)
%ENDSEL
```

The record_type is the type of RECON record to be selected. You can select any of the following record_types:

- OLDS (PRIOLD)
- SLDS (PRISLD)
- RLDS (PRILOG)
- IC (IMAGE)
- CA (CA)
- ALLOC (ALLOC)
- DBDS (DBDS)
The selection criteria depend on the type of record you select, and can be time ranges and ddnames.

Both the record_type and selection_criteria can be symbolic keywords.

As RECON records are selected, information from them is used to set the values of symbolic keywords. Depending on the type of record being selected, the values of a particular group of symbolic keywords associated with the selected record are set. The group of symbolic keywords associated with particular records are discussed in "Writing Control Keywords" on page 94.

Any values assigned to a keyword before the select group is processed are overridden when the select group is processed. The keyword values in effect after the select group is processed are the values set from the last selected record. Keyword values remain unchanged if no records are selected. In this case, the records in the select group are not processed. The next records to be processed are those that appear just after the %ENDSEL statement. A select group can occur within a delete group. When this occurs and the delete group is deleted, the select group is not processed, and no keyword values are set (or changed).

The selection criteria for a select group can cause one or more RECON records to be selected. One execution member can be output more than once depending upon the type of records that have been selected.

When the output stream is JCL, a select group can generate either concatenated or repeated DD statements. The first execution member record of the select group determines which is to be generated. Repeated DD statements are generated if this record is a JCL DD statement and the ddname is a symbolic keyword. Otherwise, a concatenated DD statement is generated.

Example:

Assume that the first record is:

//DDNAME DD DSN= . . .

In this case, concatenated DD statements are generated. Alternatively, the first record might be:

//%DDNAME DD DSN= . . .

In this case, repeated DD statements are generated. When repeated DD statements are generated, you must provide some mechanism to ensure that the repeated ddnames are unique. When selecting OLDSs, DBRC uses the OLDS ddname, which is in the OLDS RECON record. DBRC does not track ddnames for any other type of data set. Therefore, DBRC might not be able to generate unique ddnames for data sets that are not OLDSs.

The two sections that follow explain the record_type and selection_criteria parameters in more detail.

Specifying the Record Type Parameter: The types of records that can be specified on the %SELECT keyword are shown in Table 8 on page 90.
Table 8. Records That Can Be Selected Using the %SELECT Keyword

<table>
<thead>
<tr>
<th>record_type</th>
<th>What Is Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLDS</td>
<td>Specifies that OLDSs are to be selected. If dual logging is in effect, both PRIOLDS and SECOLDS can be selected.</td>
</tr>
<tr>
<td>SLDS</td>
<td>Specifies that PRISLDS are to be selected. The PRISLD is selected unless the SLDS record in the RECON data set shows the SLDS has an error. In this case, the SECRLDS is selected. The SLDS is the one created by the Log Archive utility when archiving OLDSs, not the one created by an IMS batch region. To select an IMS batch SLDS, specify RLDS.</td>
</tr>
<tr>
<td>SSLDS</td>
<td>Specifies that SECSLDS are to be selected.</td>
</tr>
<tr>
<td>RLDS</td>
<td>Specifies that RLDSs are to be selected. The PRIRLDS is selected unless the PRILOG record in the RECON data set indicates the RLDS has an error. In this case, the SECRLDS is selected. RLDS refers to both the RLDS created by the Log Archive utility and the SLDS created by an IMS batch region.</td>
</tr>
<tr>
<td>SRLDS</td>
<td>Specifies that SECRLDSs are to be selected.</td>
</tr>
<tr>
<td>IC</td>
<td>Specifies that image copy data sets are to be selected.</td>
</tr>
<tr>
<td>CA</td>
<td>Specifies that change accumulation data sets are to be selected.</td>
</tr>
<tr>
<td>ALLOC</td>
<td>Specifies that DBDS allocation records are to be selected.</td>
</tr>
<tr>
<td>DBDS</td>
<td>Specifies that DBDS records are to be selected.</td>
</tr>
</tbody>
</table>

Understanding the Selection Criteria Parameter: The selection criteria depend on the record type you select. Selection criteria are described under the sections on selecting individual record types.

The following common terms, used for selection criteria, are used in the remainder of this chapter.

dbds qualifier
Specifications the DBDS with which the selected records are to be associated. The DBDS can be specified as dbname, ddbname, or CA group name. When a CA group name is specified, all DBDSs in the CA group are used for selection. The DBDS qualifier is used when selecting:

- RLDSs
- Change accumulation data sets
- Image copy data sets
- ALLOC records
- DBDSs

time qualifier
Specifications a time stamp or a range of time stamps.

DBRC selects RECON records by their record key. Many records contain a time stamp and the time that is contained in the record key is signified by an adjacent asterisk (*) in a listing. The time qualifier that is specified in a FROMTIME or TOTIME parameter determines what records DBRC selects.

Some records such as PRILOG or PRISLD records consist of multiple DSN entries, each of which has a start time and stop time. DBRC cannot select specific DSN entries without first selecting the entire log record. The
FROMTIME and TOTIME values must be specified such that the entire log record that contains desired DSN entries is selected based on the time stamp that is in the record key.

For example, if you specify a FROMTIME of 12:00, DSN entries with time stamps later than 12:00 (but that are included in a PRISLDS record with a start time of 11:00) would not be selected and displayed by DBRC, because the PRISLDS record itself has a time stamp earlier than the specified FROMTIME. You can specify a zero time value. The time qualifier can be specified in the forms described in "Standard Time Stamp Format" on page 124.

FIRST
   Specifies that the oldest record is to be selected.

LAST
   Specifies that the most recent record is to be selected.

(FROM(time),TO(time)) or (FROM(time)) or FROM(time) or (TO(time)) or TO(time)
   Specifies that all records with time greater than or equal to the FROM time and less than or equal to the TO time are to be selected.

ALL
   Specifies that all records are to be selected.

Using the %DELETE and %ENDDEL Keywords
Use the %DELETE and %ENDDEL keywords to delete records from the output stream based on a specific condition. The syntax of a delete group follows:

```
%DELETE (relational expression) execution_member_record(s) %ENDDEL
```

%DELETE statements cannot be nested. Each %DELETE keyword must be followed by a corresponding %ENDDEL before another %DELETE keyword is encountered.

The relational expression must be of the form %keyword op 'value' or %keyword op '%userkey' where:

- %keyword is any symbolic keyword.
- 'value' is any character string enclosed in single quotes. A null string ('') can be specified for the value. You can specify a zero time value. The time qualifier can be specified in the forms described in "Standard Time Stamp Format" on page 124.
- %userkey is any keyword defined via the USERKEYS parameter in the GENJCL command. The %userkey must be enclosed in quotes and the %userkey value must exclude leading zeros.
- op is one of the following operators:
  
  EQ     Equal
  NE     Not equal
  LT     Less than
  LE     Less than or equal
  GT     Greater than
  GE     Greater than or equal
Skeletal JCL Syntax

When a `%DELETE` keyword is encountered in a skeletal JCL execution member, the relational expression is evaluated. If the expression is true, the delete group is deleted from the output stream. If the expression is false, the applicable records are copied to the output stream after keywords are resolved. If a value has not been assigned to a keyword, the value is the null string ("'"). If an undefined keyword is encountered in the skeletal JCL, an error message is received and no substitution takes place.

**Specifying Complex Expressions:** You can specify complex expressions consisting of multiple relational expressions joined by connectives

**Definitions:** A connective is one of the following logical functions:

- `&` AND function
- `|` OR function

The following is an example of a complex expression:

```
%DELETE (relexp1 | relexp2 & relexp3)
```

The `DELETE` group is deleted when the entire complex expression is logically true. Complex expressions should have the following characteristics:

- The entire `DELETE` statement (including the `%DELETE`) is limited to 80 characters, within which up to five expressions are allowed.
- A connective must be the first character following a relational expression (blanks are optional).
- The statement is processed from left to right with no connective priority and no bracketing.

where:

```
relexpx = relational expression
```

This complex expression takes the results of the OR operation between `relexp1` and `relexp2` and performs the AND operation with `relexp3`.

**Using the %SET MEMBER Keyword**

The `%SET MEMBER` keyword can be used when you are generating multistep jobs (such as GENJCL.CA with the VOLNUM parameter specified). You use `%SET MEMBER` to specify a different skeletal JCL execution member than the one that is executed for the first step of the job. The execution member you specify is used in all job steps after the first. You can explicitly code various `%keywords` in the execution member that is used in job steps after the first one. For example, you can explicitly code the `%CA0DSN` keyword, which is the name of the input change accumulation data set.

The syntax of the `%SET MEMBER` keyword is:

```

```

The `%SET MEMBER` keyword can be placed anywhere in the current skeletal JCL execution member. However, it takes effect only after processing of the current execution member is complete. If you specify more than one `%SET MEMBER` keyword, the last one specified is the one that is used. In the new member, you can place a `%SET statement that specifies any member name.
newmbrname is the name of the skeletal JCL execution member that is to be used for all job steps after the first job step. newmbrname must reside in the library named in the JCLPDS DD statement. newmbrname is not used until it is necessary to begin processing of the new member. It is possible to specify an incorrect member name and not have an error condition occur until a GENJCL command is issued that causes enough steps to be generated to cause the member to be read.

Using the %SET TIMEFMT Keyword

The %SET TIMEFMT keyword is used to specify a format for time stamps that appear in GENJCL output. See “TIMEFMT Parameter” on page 125 for more information on time stamps.

For GENJCL.USER, the default is:

TIMEFMT(0,0,C,2,TEMP,1)

Note that the GENJCL TIMEFMT default values have been chosen to produce correct output with IBM-supplied skeletal JCL. If you use the %SET statement to change the TIMEFMT values in a way that affects the values substituted into the IBM-supplied JCL statements, the results might be invalid.

Example:

Here is an example of the %SET TIMEFMT keyword in skeletal JCL.

%SET TIMEFMT(,N)
%SELECT RLDS(%SSID,LAST)
LOGEND =%LOGETIM
%ENDSEL

And here is what the output from the preceding example of %SET would render:

LOGEND =960111315000

The next four examples are based on the following skeletal JCL member (called USER01) that is used with GENJCL.USER.

%SELECT RLDS(%SSID,LAST)
LOGETIM=%LOGETIM
%ENDSEL

- This sample output format was obtained by using the USER01 JCL, specifying SSID(XXXX), and using the default for TIMEFMT, which is:
  TIMEFMT(0,0,C,2,1)
  LOGETIM=960021315001-0700

- This sample output format was obtained by using the USER01 JCL, specifying SSID(XXXX), and using the default for TIMEFMT on an open log.
  LOGETIM=000000000000+0000

- This sample output format was obtained by using the USER01 JCL, specifying SSID(XXXX), and using the specification, TIMEFMT(,N).
  LOGETIM=960111314544

- This sample output format was obtained by using the USER01 JCL, specifying SSID(XXXX), and using the specification, TIMEFMT(,,P,4).
  LOGETIM=1996.011 13:15:00.0 -07:00

Restriction: The %SET TIMEFMT keyword affects GENJCL output only if it is issued via the GENJCL command or from a %SET statement in the skeletal JCL.

The syntax of the %SET TIMEFMT keyword is:
Writing Control Keywords

When writing control keywords, you need to observe the following conventions and restrictions:

- Control keywords must begin in column 1 of a skeletal JCL execution member record.
- Everything specified for the keyword must be contained on one record. Any data following the control statement is ignored.
- Any number of skeletal JCL execution member records can be contained in a control group.
- Delete groups and select groups cannot be nested. However, a select group can be contained within a delete group, or a delete group can be contained within a select group.
- Execution member records containing control keywords are not copied to the output stream.

Selecting OLDSs

The syntax of the %SELECT keyword to select OLDSs is as follows:

```plaintext
%SELECT(SSID,OLDS_qualifier)
```

- **ssid**
  Subsystem ID of the IMS online control region that created the OLDS.

- **olds_qualifier**
  Specifies the OLDSs that are to be selected as follows:
  
  **INUSE**
  Specifies that the OLDS that is currently in use by the specified subsystem is to be selected. If dual logging is in effect, both the primary and secondary OLDSs are selected.

  **LATEST**
  Specifies that the OLDS that was most recently opened by the specified subsystem is to be selected. If dual logging is in effect, both the primary and secondary OLDSs are selected.

  **UNARCH**
  Specifies that all unarchived OLDSs for the specified subsystem are to be selected. If dual logging is in effect, both the primary and secondary OLDSs are selected.

  **(DDNAME)**
  Specifies one or more OLDSs by ddname. If dual logging is in effect and both the primary and secondary OLDS are to be selected, both ddnames should be specified.

  **ALL**
  Specifies that all OLDSs for the specified subsystem are to be selected.
In the execution member records following the %SELECT keyword, you use symbolic keywords to specify the type of information to be gathered for each OLDS record that is selected. The types of information you can gather are:

- **%OLDSDDN**: The ddname of the OLDS.
- **%OLDSDSN**: The data set name of the OLDS.
- **%OLDSTYP**: The OLDS type. DBRC sets the %OLDSTYP to P for primary OLDS or S for secondary OLDS.
- **%OLDOTIM**: The time the OLDS was opened. DBRC sets %OLDOTIM in the form yydddhhmmss{offset}.
- **%OLDCTIM**: The time the OLDS was closed. DBRC sets %OLDCTIM in the form yydddhhmmss{offset}. If the OLDS has not been closed, DBRC sets the time to 000000000000+0000.
- **%OLDSSEL**: Set to YES if any OLDS was selected. Otherwise, set to NO.
- **%OLDFRID**: The log record sequence number of the first log record of the OLDS.
- **%OLDLRID**: The log record sequence number of the last log record of the OLDS.

**Example 1:** The following select group generates repeated DD statements for all unarchived OLDSs belonging to subsystem IMSA.

```plaintext
%SELECT
  OLDS(IMSA,UNARCH)
  //%OLDSDDN
  DD DSN=%OLDSDSN,DISP=SHR
%ENDSEL
```

The JCL generated by this select group might be:

```plaintext
//DFSOLP00 DD DSN=IMS.OLDSP00,DISP=SHR
//DFSOLS00 DD DSN=IMS.OLDS00,DISP=SHR
//DFSOLP01 DD DSN=IMS.OLDSP01,DISP=SHR
//DFSOLS01 DD DSN=IMS.OLDS01,DISP=SHR
```

**Example 2:** The following select group generates a list of all OLDSs belonging to subsystem IMSA:

```plaintext
%SELECT
  OLDS(IMSA,ALL)
  %OLDSTYPOLDS DD NAME=%OLDSDDN
  DSN=%OLDSDSN
  CLOSE TIME=%OLDCTIM
%ENDSEL
```

The output generated by this select group might be:

```plaintext
POLD DD NAME=DFSOLP00
  DSN=IMS.POLD00
  CLOSE TIME=842351638193+0012
POLD DD NAME=DFSOLP01
  DSN=IMS.POLD01
  CLOSE TIME=842351712246+0055
POLD DD NAME=DFSOLP02
  DSN=IMS.POLD02
  CLOSE TIME=000000000000+0000
```

**Selecting SLDSs**

The syntax of the %SELECT keyword to select SLDS is:

```plaintext
%SELECT—sls_type(ssid,time_qualifier)
```
slds_type
Can be specified as $LDS (for the PRISLD) or $SSLDS (for the SECSLD). This keyword selects the entire RECON record, not individual data sets. Therefore, all data sets identified in the SLDS record are selected.

If the selected PRILOG data set is marked in error, DBRC selects the associated secondary data set if one that is not also in error exists. If the associated data set exists but is also in error, DBRC selects the original record. If $SSLDS is specified, the SECLOG data set is selected regardless of whether it is marked in error. An SLDS record might not contain a DSN entry. In this case, the values for $SLDSDSN($LOGDSN), $SLDUNIT ($LOGUNIT), and $SLDVOLS ($LOGVOLS) are null. You must use the $DELETE statement in order to ensure that DBRC substitutes valid data in the generated JCL. See "Example 4" on page 97.

ssid
The subsystem ID (of the IMS online control region) that created the OLDSs that were archived to become SLDSs.

time_qualifier
The time qualifier as specified in "Understanding the Selection Criteria Parameter" on page 90.

In the execution member records following the $SELECT keyword, you specify (using symbolic keywords) the type of information to be gathered for each SLDS record that is selected. The types of information you can gather are:

%SLDSDSN The data set name of the SLDS.
%SLDUNIT The unit type of the SLDS.
%SLDVOLS The volume serial number of the SLDS.
%SLDFSEQ The file sequence number of the SLDS.
%SLDSTIM The start time of the SLDS. DBRC sets the $SLDSTIM in the form yyddhhmmmst{offset}.
%SLDETIM The stop time of the SLDS. DBRC sets the $SLDETIM in the form yyddhhmmmst{offset}.
%SLDSTIM The start time of the PRISLDS/SECSLDS. DBRC sets $SLDSTIM in the form yyddhhmmmst{offset}.
%SLDSSB Set to YES if any SLDS was selected. Otherwise, set to NO.
%SLDRMT Set to YES if the SLDS was created at the tracking site. Otherwise, set to NULL.
%SLDFRID The log record sequence number of the first log record of the SLDS.
%SLDLRID The log record sequence number of the last log record of the SLDS.
%SLDCHKT The Checkpoint Types value of the SLDS data set entry. This is a hexadecimal representation of the bit settings of the value where:
  • Bit 0 - simple checkpoint
  • Bit 1 - snapq checkpoint
  • Bit 2 - dumpq checkpoint
  • Bit 3 - purge checkpoint
  • Bit 4 - freeze checkpoint
  • Bit 5-7 - not used
Example 3: The following select group generates the most recent SLDS for subsystem IMSA.

```plaintext
%SELECT SLDS(IMSA,LAST)
LATEST SLDS:
  DSN=%SLDSASN
  STOP TIME=%SLDETIM
%ENDSEL
```

The output generated by this select group might be:

```plaintext
LATEST SLDS:
  DSN=IMS.SLDS
  STOP TIME=841230812339
```

If the SLDS record has more than one data set, then all the data sets to be selected and your output may look like this:

```plaintext
LATEST SLDS:
  DSN=IMS.IMSA.SLDSP.D97107.T1405235.V06
  STOP TIME=971071420469+0100
LATEST SLDS:
  DSN=IMS.IMSA.SLDSP.D97107.T1420469.V03
  STOP TIME=971071420579+0100
LATEST SLDS:
  DSN=IMS.IMSA.SLDSP.D97107.T1420579.V00
  STOP TIME=971071430087+0100
```

Example 4: The following select group generates a concatenated DD statement for all SLDSs for subsystem IMSA that have an open time greater than or equal to 840031903298.

```plaintext
%SELECT SLDS(IMSA,FROM(840031903298))
%DELETE (%SLDSASN EQ '')
//SLDS DD DSN=%SLDSASN,DISP=OLD,
//  UNIT=%SLDUNIT,
//  VOL=SER(%SLDVOLS),
//  LABEL=(1,SL)
%ENDDEL
%ENDSEL
```

The generated DD statements might be:

```plaintext
//SLDS DD DSN=IMS.SLDS1,DISP=OLD,
//  UNIT=3400,
//  VOL=SER(VOLUM1,VOLUM2,VOLUM3),
//  LABEL=(1,SL)
//DD DSN=IMS.SLDS2,DISP=OLD,
//  Unit=3400,
//  VOL=SER(VOLUM4,VOLUM5,VOLUM6,
//  VOLUM7,
//  VOLUM8,VOLUM9),
//  LABEL=(1,SL)
```

Attention: In this example, a JCL continuation card was generated. This is because the volume serial number list was longer than the output record.

The `%DELETE` statement prevents the JCL statement from being generated for an SLDS record that does not contain a DSN entry.

Selecting RLDSs

The syntax of the `%SELECT` keyword to select RLDSs can be specified as follows:
Skeletal JCL Syntax

Use SRLDS to request that secondary RLDS records be selected. Secondary RLDS records can be specifically requested only when you specify SSID. When you specify dbds_qualifier, you are specifically requesting primary RLDS records. If the primary RLDS is marked in error, DBRC selects the associated secondary data set if one that is not also in error exists. If no associated data set exists or if it is also in error, DBRC selects the original record.

**ssid**

The subsystem ID of the IMS online control region or an IMS batch region. PRILOG (or SECLOG) records corresponding to the specified SSID are selected to satisfy the specified search criteria. Because RECON records are selected, all data sets identified by the record are selected.

**time_qualifier**

Time qualifier as specified in "Understanding the Selection Criteria Parameter" on page 90.

**dbds_qualifier**

DBDS qualifier as specified in "Understanding the Selection Criteria Parameter" on page 90. When a dbds_qualifier is specified, only RLDSs that contain log records corresponding to the specified DBDS are selected. (In other words, those RLDSs for which an ALLOC record exists in the RECON data set.) Only primary RLDSs can be selected when the dbds_qualifier is specified.

**max_volumes**

The maximum number of log volumes to be selected. If max_volumes is specified, processing of the select group terminates when the specified number of log volumes is reached. If max_volumes is specified and a log merge situation exists, more than the specified number of volumes can be selected. This is to ensure that a valid subset of logs is selected.

In the execution member records following the %SELECT keyword, you use symbolic keywords to specify the type of information to be gathered for each RLDS record that is selected. The types of information you can gather are:

- `%LOGDSN` The data set name of the RLDS.
- `%LOGFSEQ` The file sequence number of the RLDS.
- `%LOGUNIT` The unit type of the RLDS.
- `%LOGVOLS` The volume serial number of the RLDS.
- `%LOGSTIM` The start time of the RLDS. DBRC sets `%LOGSTIM` in the form yydddhhmmst(offset).
- `%LOGETIM` The stop time of the RLDS. DBRC sets `%LOGETIM` in the form yydddhhmmst(offset). If the data set is still open, the time is set to 0000000000000000.
- `%LOGSEL` Set to YES if any log data sets were selected. Otherwise, set to NO.
- `%LOGMERG` Set to YES if a log merge is required. Otherwise, set to NO. %LOGMERG is always set to NO if SSID is specified.
- `%LOGONL` Set to YES if the RLDS is associated with an online region. Set to NO for batch logs.
- `%LOGRMT` Set to YES if the RLDS was created at the tracking site. Otherwise, set to NULL.
The log record sequence number of the first log record of the RLDS.

The log record sequence number of the last log record of the RLDS.

The start time of the PRILOG/SECLOG. DBRC sets %LOGOTIM in the form yydddhhmmsst{offset}.

Example 5: The following select group generates a DD statement for the most-recent RLDS for subsystem BATCHJOB. This example assumes the RLDS is still open.

```jcl
%SELECT  RLDS(BATCHJOB,LAST)
%DELETE  (%LOGETIM NE '000000000000+0000')
//LOGDD  DSN=%LOGDSN,DISP=OLD,
//       UNIT=%LOGUNIT,
//       VOL=SER=(%LOGVOLS),
//       LABEL=(%LOGFSEQ,SL)
%ENDDEL
%ENDSEL
```

If no RLDS is recorded in the RECON data set for the subsystem or if the most-recent RLDS has been closed, no DD statement is generated. Otherwise, the generated DD statement might be:

```jcl
//LOGDD  DSN=IMS.RLDS,DISP=OLD,
//       UNIT=3400,
//       VOL=SER=(VOLUM1,VOLUM2)
//       LABEL=(1,SL)
```

Selecting Image Copy Data Sets

The syntax of the %SELECT keyword to select image copy data sets is:

```jcl
%SELECT-IC(dbds_qualifier,time_qualifier)
```

dbdsQualifier and timeQualifier

DBDS qualifier and time qualifier are as specified in "Understanding the Selection Criteria Parameter" on page 90.

In the execution member records following the %SELECT keyword, you specify (using symbolic keywords) the type of information to be gathered for each image copy record that is selected. If the duplicate image copy is marked in error, the DBRC selects the primary image copy. The types of information you can gather are:

- %ICDSN The data set name of the image copy data set.
- %ICTYPE The image copy's type: BATCH, ONLINE, CIC, SMSCIC, or SMSNOCIC.
- %ICFSEQ The file sequence number of the image copy data set if it is a NONHSSP type; otherwise, ICFSEQ is null.
- %ICSEL Set to YES if any image copy data set was selected. Otherwise, ICSEL is set to NO.
- %ICSTOP The stop time of the image copy data set ID that is present; otherwise 1CSTOP is null.
- %ICTIME The run time of the image copy. DBRC sets %ICTIME in the format yydddhhmmsst{offset}.
- %ICUNIT The unit type of the image copy data set if it is a NONHSSP type; otherwise, ICUNIT is null.
The number of volumes of the image copy data set if it is a NONHSSP type; otherwise, ICVCNT is null.

The volume serial number list of the image copy data set if it is a NONHSSP type; otherwise, ICVOLS is null.

The update set identifier (USID).

Set to YES if the image copy is cataloged (HSSP CICs only). Otherwise, ICCAT is set to NO.

Set to YES if a duplicate image copy data set is associated with the selected image copy data set. Otherwise, IC2SEL is set to NO.

The following keywords are set only when a duplicate image copy data set exists; otherwise, they are null:

The data set name of the duplicate image copy data set.

The file sequence number of the duplicate image copy data set. If the IC was created by HSSP, IC2FSEQ is set to null.

The unit type of the duplicate image copy data set. If the IC was created by HSSP, IC2UNIT is set to null.

The number of volumes of the duplicate image copy data set. If the IC was created by HSSP, IC2VCNT is set to null.

The volume serial number list of the duplicate image copy data set. If the IC was created by HSSP, IC2VOLS is set to null.

Example 6: The following select group generates a DD statement for the oldest image copy data set for the DBDS with a database name of SHISAMDB and a ddbname of SHISAMDD.

```%
%SELECT IC(((SHISAMDB,SHISAMDD),FIRST)
///ICDD DD DSN=%ICDSN,DISP=OLD,
/// VOL=SER=(%ICVOLS),
/// UNIT=%ICUNIT,
/// LABEL=(%ICFSEQ,SL)
%ENDSEL
```

The generated DD statement might be:

```%
///ICDD DD DSN=SHISAMDB.SHISAMDD.IC,DISP=OLD,
/// VOL=SER=(VOLUM1),
/// UNIT=3400,
/// LABEL=(1,SL)
%ENDSEL
```

Selecting Change Accumulation Data Sets

The syntax of the %SELECT keyword to select change accumulation data sets is:

```
%SELECT—CA(dbds_qualifier,time_qualifier)
```

dbds_qualifier and time_qualifier

DBDS qualifier and time qualifier are as specified in “Understanding the Selection Criteria Parameter” on page 90.

In the execution member records following the %SELECT keyword, you use symbolic keywords to specify the type of information to be gathered for each change accumulation record that is selected. The types of information you can gather are:

The change accumulation data set name.
%CAFSEQ  The file sequence number of the change accumulation data set.
%CAUNIT  The unit type of the change accumulation data set.
%CAVCNT  The number of volumes of the change accumulation data set.
%CAVOLS  The volume serial number list of the change accumulation data set.
%CALGTM  The volume stop time of the last log volume that was used as input to the change accumulation data set. DBRC sets %CALGTM in the form yyydddhhmmss{offset}.
%CATIME  The change accumulation data set time in the form yyydddhhmmss{offset}.
%CASEL   Set to YES if any change accumulation data sets are selected. Otherwise, set to NO.

Example 7: The following select group lists all change accumulation data sets created since time 842310000000+0000 for CA group CAGRP1.

%SELECT CA((CAGRP1),FROM(842310000000+0000))
DSNAME=%CADSN
VOLUMES=%CAVOLS
RUNTIME=%CATIME
LOGTIME=%CALGTM
%ENDSEL

The generated output might be:

DSNAME=CAGRP1.DSN1
   VOLUMES=VOLUM1,VOLUM2,VOLUM3,
      VOLUM4,
   // VOLUM5,VOLUM6
   RUNTIME=842310618230
   LOGTIME=842302315557

DSNAME=CAGRP1.DSN2
   VOLUMES=VOLUM1,VOLUM2
   RUNTIME=842361824443
   LOGTIME=842360934519

In this example, the volume serial number list for the first data set does not fit on the output record. Therefore, a JCL continuation statement is generated (even though JCL is not being generated).

Selecting DBDS Allocation (ALLOC) Records
The syntax of the %SELECT keyword to select ALLOC records can be one of the following:

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>%SELECT—ALLOC(dbds_qualifier,time_qualifier)</td>
<td>%SELECT—ALLOC(PRILOG,time_qualifier)</td>
</tr>
</tbody>
</table>

dbds_qualifier and time_qualifier
DBDS qualifier and time qualifier are as specified in "Understanding the Selection Criteria Parameter" on page 90. When a dbds_qualifier is specified, ALLOC records corresponding to the specified DBDSs are selected. All ALLOC records for the specified DBDS with allocation times within the bounds of the specified time_qualifier are selected. When PRILOG is specified, all ALLOC records corresponding to PRILOG records within the specified time bounds are selected.
In the execution member records following the %SELECT keyword, you use symbolic keywords to specify the type of information to be gathered for each ALLOC record that is selected. The types of information you can gather are:

%DBNAME The database name.
%DBDDN The database ddname or area name.
%ALLTIME The allocation time stamp in the form yydddhhmmsst{offset}.
%DALTIME The deallocation time stamp in the form yydddhhmmsst{offset}. Set to 000000000000+0000 if there is no deallocation time stamp.
%ALLDSSN The data set sequence number.
%PLGTIME The start time of the corresponding PRILOG record.
%ALLSEL Set to YES if any ALLOC records are selected. Otherwise, ALLSEL is set to NO.
%ALLUSID The update set identifier (USID)

Example 8: The following select group generates a list of information about all ALLOC records for the DBDS with a database name of SHISAMDB and ddname of SHISAMDD:

```plaintext
%SELECT ALLOC((SHISAMDB,SHISAMDD),ALL)
  DBNAME   %DBNAME
  DDNAME   %DBDDN
  ALLOC time %ALLTIME
  DEALL time %DALTIME
  PRILOG time %PLGTIME
```

The generated output might be:

```
DBNAME    SHISAM
DDNAME    SHISAM
ALLOC TIME 832560800000+0000
DEALL TIME 000000000000+0000
PRILOG TIME 832560630000+0000
```

Selecting DBDS Records
The syntax of the %SELECT keyword to select DBDS records is:

```
%SELECT DBDS(dbds_qualifier)
```

dbds_qualifier

DBDS qualifier is as specified in "Understanding the Selection Criteria Parameter" on page 90. For DEDBs, the select group is processed once for each defined area data set (ADS) for each specified area. For other types of databases, the select group is processed once for each specified DBDS.

In the execution member records following the %SELECT keyword, you use symbolic keywords to specify the type of information to be gathered for each DBDS record that is selected. The types of information you can gather are:

%DBNAME The database name.
%DBDDN The DBDS ddname or DEDB area name.
%DBTYPE Set to FP when the selected DBDS is an area of a Fast Path database. Set to DLI for DBDSs of non-HALDBs. Set to PDATA for data DBDSs of HALDBs. Set to PINDEX for primary index DBDSs of HALDBs. Set to PILDS for ILDS DBDSs of HALDBs.
%DBDSN The data set name of the DBDS or ADS.

%DBADDN For DEDBs, the ddname of the ADS. For other types of databases, DBADDN is set to null.

%DBADSAV For DEDBs, set to AVAIL if the ADS is indicated as available in the RECON data set. Set to UNAVAIL if the ADS is unavailable. For other types of databases, DBADSAV is set to null.

%DBDSSEL Set to YES if any DBDS records are selected. Otherwise, DBDSDEL is set to NO.

%DBUSID For DEDBs, set to AVAIL if the ADS is indicated as available in the RECON data set. Set to UNAVAIL if the ADS is unavailable. For other types of databases, DBUSID is set to NULL.

%DBDSNRV Set to YES if the DBDS is non-recoverable. Otherwise, DBDSNRV is set to NO.

Example 9: The following select group generates a series of DD statements for available area data sets for the area named DBHVSAM1. This area is in the DEDB named DIVNTZ04.

%SELECT DBDS((DIVNTZ04, DBHVSAM1))
%DELETE (%DBADSAV ne 'AVAIL')
//%DBADON DD DSN=%DBDSN, DISP=OLD
%ENDDEL
%ENDSEL

The generated output might be:
//FP1ADD1 DD DSN=IMS.FP1ADD1, DISP=OLD
//FP1ADD2 DD DSN=IMS.FP1ADD2, DISP=OLD

Understanding Skeletal JCL Default Members

Skeletal JCL default members are used to set default values for keywords you have defined in the skeletal JCL execution members. The use of default members is optional. You must supply any default members to be used.

Writing Default Members

Default members can have two types of records: assignment records or comment records. Assignment records assign default values to user-defined keywords. Assignment records must contain a percent sign (%) in column 1. If a record does not contain a percent sign in column 1, it is a comment record, which DBRC ignores.

%user defined keyword='value'

The value assigned to the keyword can be any text string, including the null string (' '). If the value contains a single quote, you must use two single quotes. The entire value must be contained on one record. Any data following the closing single quote is ignored. A closing single quote is required. If a closing single quote is missing, an error message is generated and the GENJCL command fails.

Example 10: A default member contains these records:
%DEDBNAM = 'DIVNTZ04'
%AREANAM = 'DBHVSAM1'

A skeletal JCL member contains:
DATABASE NAME = %DEDBNAM
AREA NAME = %AREANAM
Skeletal JCL Syntax

Using the DEFAULTS parameter and assuming the values are not overridden, the GENJCL command generates the following:

```plaintext
DATABASE NAME = DIVNTZ04
AREA NAME = DBHVSAM1
```

Specifying Default Members

You can specify default members either explicitly or implicitly.

Members are explicitly specified using the DEFAULTS parameter on the GENJCL command. Up to 10 default members can be specified.

Implicit specification can be used for the GENJCL commands that apply to a DBDS (GENJCL.IC, GENJCL.01C, and GENJCL.RECOV) or CA group (GENJCL.CA). In addition, implicit specification can be used on the GENJCL.USER command. The default members to be implicitly used are specified using the DEFLTJCL parameter on the INIT.DBDS, CHANGE.DBDS, INIT.CAGRP, and CHANGE.CAGRP commands. Only one default member is allowed per DBDS or CA group.

The use of an implicit default member can be overridden with the NODEFLT parameter on the GENJCL command. When both explicitly and implicitly specified default members are used, explicitly specified members have precedence. That is, if a keyword is assigned a value in both members, the value assigned by the explicitly specified member is used.

If a keyword is assigned a value both in a default member and in the USERKEYS parameter of the GENJCL command, the latter value is used. USERKEYS parameter values override default member values.

Using User-Supplied or Modified Skeletal JCL

Before using your skeletal JCL execution or default members, you must do the following:

- Add the JCLPDS and JCLOUT ddnames to the JCL required to run DBRC. JCLPDS identifies the partitioned data set containing the skeletal JCL execution members. JCLOUT identifies the data set to which the generated job is to be written. Output is in card image format. The output data set can be a punch file, a DASD data set that you plan to examine before submitting the job for execution, or directly to the z/OS internal reader.
- Add to the skeletal JCL execution member any STEPLIB and STEPCAT ddnames and job accounting information that your installation requires. If the DD statements contain comments or continuation characters, they are ignored when the JCL is generated. Except for the skeletal JCL member for the JOB statement, do not add to your skeletal JCL any JOBCAT, JOBLIB, and JES control statements; doing so causes errors if multiple steps are generated.
Chapter 6. Hints and Tips for DBRC

This chapter provides task-oriented instructions for frequently-used procedures, including the following topics:

In This Chapter:
- "Changing the RECON Data Set to Output Time Stamps in Local Time of Origin" on page 105
- “Locating the Last SLDS Stop Time in RECON” on page 106
- “Adjusting GENMAX When It Is Reached or It Is Too High” on page 107
- “Getting PRILOG Compression to Work” on page 108
- “Getting Notified When the PRILOG Record Exceeds the Maximum Record Size” on page 109
- “Closing an Open Online PRILOG” on page 109
- “Deleting Log Records” on page 110
- “Working with Subsystem Records (SSYS)” on page 111
- “Removing Authorization Inconsistency Between the SSYS from DB/AREA Records” on page 112
- “Getting Change Accumulation to Start Processing Logs Again” on page 112
- “Getting Change Accumulation Working When It States Nothing to Process” on page 112
- “Moving Log Data Sets” on page 113
- “Cataloging Data Sets” on page 113
- “Performing Multiple Cold Starts in a Test Environment” on page 114
- “Avoiding Some Causes of RECON Data Set Enqueue Problems” on page 115

Changing the RECON Data Set to Output Time Stamps in Local Time of Origin

If the listing of the RECON data set does not list the offset value for time stamps, the record might have been created when the offset value was different from the current offset value. Using DBRC commands to change or delete such a record will result in a “record-not-found” condition because the current offset is applied if it is not included in the time stamp parameter of the command.

To change the RECON data set to output time stamps in local time of origin, when they were created, use one of the following:

CHANGE.RECON TIMEFMT(0,0,P,4,PERM)

or

CHANGE.RECON TIMEFMT(0,0,P,2,PERM)

This will permanently change the RECON data set to output time stamps when the records were originally created with the appropriate offsets in punctuated four-digit or two-digit year formats. The time stamp listed can be used in commands such as the following:

CHANGE.PRILOG STARTIME('2002.063 16:11:58.1 -08:00') -
DSSTART('2002.063 16:11:58.1 -08:00') ERROR RLDS
Locating the Last SLDS Stop Time in RECON

Follow this procedure to find the last SLDS stop time in the RECON data set using the GENJCL.USER command. This can be done when IMS is still running and also works if the PRISLD is already closed.

1. Create a skeletal JCL execution member first. Here is an example of a member named SECLSDL:

   ```
   %SELECT SLDS(%USSID,LAST)
   %ENDSEL
   
   SLDSDD BEG=%SLDSTIM
   END=%SLDETIM
   VOL=%SLDVELS
   UNIT=%SLDUNIT
   DSN=%SLDSDSN
   ```

2. Issue the GENJCL.USER command indicating the execution member to run with. Below is a sample of the GENJCL.USER command indicating that the command is to be run with member SECLSDL:

   ```
   GENJCL.USER MEMBERSLDS
   NOJOB USERKEYS((%USSID,'IMS1'))
   JCLOUT(SYSPRINT)
   ```

3. Below is the SLDS listing:

   ```
   DSN=IMSVS.SLDSP.IMS1.D04194.T0859467.V00 UNITS=SYSDA
   START = 04.194 08:59:46.7 FIRST DS LSN= 0000000000000001
   STOP = 04.194 09:02:33.9 LAST DS LSN= 0000000000000B70
   FILE SEQ=0001 #VOLUMES=0001
   CHECKPOINT TYPES=80: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=N
   
   VOLSER=000000 STOPTIME = 04.194 09:00:49.0
   CHKPTCT=1 CHKPT ID = 04.194 08:59:48.4
   LOCK SEQUENCE#  000000000000
   
   DSN=IMSVS.SLDSP.IMS1.D04194.T0900490.V00 UNITS=SYSDA
   START = 04.194 09:00:49.0 FIRST DS LSN= 0000000000000B71
   STOP = 04.194 09:01:17.5 LAST DS LSN= 000000000000010AD
   FILE SEQ=0001 #VOLUMES=0001
   CHECKPOINT TYPES=00: SIMPLE=N SNAPQ=N DUMPQ=N PURGE=N FREEZE=N
   
   VOLSER=000000 STOPTIME = 04.194 09:01:17.5
   CHKPTCT=0 CHKPT ID = 04.194 08:59:48.4
   LOCK SEQUENCE#  000000000000
   
   DSN=IMSVS.SLDSP.IMS1.D04194.T0901175.V00 UNITS=SYSDA
   START = 04.194 09:01:17.5 FIRST DS LSN= 0000000000000163E
   STOP = 04.194 09:01:41.2 LAST DS LSN= 000000000000163F
   FILE SEQ=0001 #VOLUMES=0001
   CHECKPOINT TYPES=08: SIMPLE=N SNAPQ=N DUMPQ=N PURGE=N FREEZE=Y
   ```
Below is the output that results from issuing the previous sample GENJCL.USER command:

```
SLDSDD  BEG=041940901412-0700
       END=041940902339-0700
       VOL=000000
       UNT=SYSDA
       DSN=IMSVS.SLDSPI.1M4194.T0901412.V00
```

### Adjusting GENMAX When It Is Reached or It Is Too High

Prior to image copy utility execution, verification exit processing may determine that one of the following situations is true when image copy data sets are being reused:

- **Situation 1** - GENMAX Value is Too Low
  - The GENMAX value was reached.
  - The oldest image copy cannot be reused because it is within the recovery period.
  - No available image copy exist that can be used.
  - Message DSP0063I was displayed.

See [“Solutions When GENMAX is Too Low” on page 107](#) on how to correct this.

- **Situation 2** - GENMAX Value is Too High
  - GENMAX has not been reached so the in use image copy data sets cannot be used.
  - No available image copy data sets exist that can be used.
  - Message DSP0084I was displayed.
  - No recovery period is defined.

See [“Solution When GENMAX is Too High” on page 108](#) on how to correct this.

### Solutions When GENMAX is Too Low

Either of the following will solve Situation 1 as described in [“Adjusting GENMAX When It Is Reached or It Is Too High”](#):

- Increase GENMAX and define additional image copy data sets.
- Change the recovery period to allow the oldest image copy data set to be used.

Issue the `CHANGE.DBDS` command to increase the GENMAX and INIT.IC parameters to define the additional image copy data sets:

1. Create a JCL job similar to one of the following jobs. These examples assume that your DBD name is THISDB, that your area name is AREA1, and that your previous GENMAX value was 2.

   ```
   //CHNGDBDS JOB
   
   //SYSIN DD *
   CHANGE.DBDS DBD(THISDB) AREA(AREA1) -
       GENMAX(4)
   /*
   //INITIC JOB
   
   //SYSIN DD *
   INIT.IC DBD(THISDB) DDN(DDN1) -
   ```
Hints and Tips

- ICDSN(IMS.*.NEWICDSN)
- INIT.IC DBD(THISDB) DDN(DDN1) - 
  (IMS.*.NEWICDSN2)
  /*
  Issue CHANGE.DBDS RECOVPD to reduce the recovery period.
  //CHNGDBDS JOB
  //SYSIN DD*
  CHANGE.DBDS DBD(THISDB) AREA(AREA1) RECOVPD(10)
  /*
  2. Run the job.
  3. Check the JES log to ensure that the job ran successfully.
  4. Run a LIST command to see the new GENMAX value (optional).
  5. Run your image copy job.

Solution When GENMAX is Too High

Either of the following steps will resolve Situation 2 as described in “Adjusting GENMAX When It Is Reached or It Is Too High” on page 107.

- Issue an INIT.IC command to define a new image copy data set for the identified database data set or area data set
  -OR-
  - Issue a CHANGE.DBDS command to lower the GENMAX value.

Attention: If GENMAX is lowered using the CHANGE.DBDS command, the new GENMAX value is recorded regardless of whether the oldest image copies are undeletable, because they are within the recovery period.

Getting PRILOG Compression to Work

If your PRILOG is not getting compressed, you will receive message DSP1150I because of one of the following reasons:

- EARLIEST CHECKPOINT
  If this time stamp is within the first DSN entry in the PRILOG, then compression is not possible. Most often, the oldest checkpoint needed for restart causing the problem is the checkpoint to rebuild the message queues. If so, take a SNAPQ checkpoint. When the OLDS that records this SNAPQ checkpoint is archived, the earliest checkpoint time stamp is updated.

- EARLIEST ALLOC TIME
  The databases listed in the LOGALL record are in earliest allocation time stamp order. The earliest allocation time listed in the LOGALL record is associated with the first databases in the listing.
  Concurrent image copies (CICs) and Online image copies (OICs) update the allocation time to reflect the earliest log that is needed for recovery. The checkpoint IDs and checkpoint counts on the logs prior to the start of the CIC are used to determine where the new allocation time is set. If the allocation times are not being moved forward-in-time, ensure that checkpoint IDs and counts are being recorded in the DSN entries.
  Old allocations for any databases in the LOGALL record can cause compression to fail. Image copy databases on a regular basis to allow old allocations to be deleted.

- LOG RETENTION TIME
  Indicates the minimum amount of time that DBRC is to keep log records in the RECON data set.
Getting Notified When the PRILOG Record Exceeds the Maximum Record Size

DBRC ABENDs when the PRILOG record exceeds the 16M maximum record size. DBRC issues warning messages indicating that a PRILOG record is approaching its maximum size. You can control the timing of those messages, taking into account the typical rate at which your system fills OLDSs and the average number of volumes in each archive data set.

The current size of the PRILOG record is printed on the listing of the PRILOG record.

Closing an Open Online PRILOG

During time stamp recoveries, when DBRC reports that there are open allocations that are not needed for the recovery, it might be easier to close the open PRILOG rather than deleting the individual allocations. Use the NOTIFY.PRILOG command to close the open PRILOG.

If an open PRILOG record is found for an online IMS subsystem, and the PRILOG record is not for the current run of IMS, it indicates that the last OLDS for this online IMS has not yet been archived. If the OLDS is no longer available and you need to close the open PRILOG record, the following commands can be used to create a dummy DSN entry in the PRILOG:

```
NOTIFY.PRILOG SSID(IMS1) RLDS DSN(DUMMY) FIRSTREC(00) -
  STARTIME(042010652499) VOLSER(VOL001)
NOTIFY.PRILOG SSID(IMS1) RLDS -
  STARTIME(042010652499) RUNTIME(042010753260) LASTREC(500)
CHANGE.PRILOG STARTIME(042010652499)
  ERROR - DSSTART(042010653260)
```

Below is the PRILOG as it appears in the RECON data set before issuing the NOTIFY.PRILOG command to close it:

```
PRILOG
START = 04.201 06:52:49.9   RECORD SIZE= 464
STOP = 00.000 00:00:00.0   SSID=IMS1
GSNAME=**NULL**   VERSION=9.1
FIRST RECORD ID= 0000000000000001
EARLIEST CHECKPOINT = 04.201 06:52:52.0

DSN=IMSVS.RLDSP.IMS1.D04201.T0652499.V00   UNIT=SYSDA
START = 04.201 06:52:49.9
STOP = 04.201 06:53:12.7
FILE SEQ=0001   #VOLUMES=0001

DSN=IMSVS.RLDSP.IMS1.D04201.T0652499.V00   UNIT=SYSDA
DSN=IMSVS.RLDSP.IMS1.D04201.T0653127.V00   UNIT=SYSDA
```
Below is the PRILOG as it appears in the RECON data set after issuing the NOTIFY.PRILOG command to close it (differences are highlighted):

```
PRILOG RECORD SIZE= 624
START = 04.201 06:52:49.9 * SSID=IMS1 VERSION=9.1
STOP = 04.201 07:53:26.0 #DSN=3
GSGNAME=**NULL**
FIRST RECORD ID= 0000000000000001 PRILOG TOKEN= 0
EARLIEST CHECKPOINT = 04.201 06:52:52.0

DSN=IMVS.RLDSP.IMS1.D04201.T0652499.V00 UNIT=SYSDA
START = 04.201 06:52:49.9 FIRST DS LSN= 0000000000000001
STOP = 04.201 06:53:12.7 LAST DS LSN= 00000000000000642
FILE SEQ=0001 #VOLUMES=0001

VOLSER=000000 STOPTIME = 04.201 06:53:12.7
CKPTCT=1 CHKPT ID = 04.201 06:52:52.0
LOCK SEQUENCE#= 000000000000

DSN=IMVS.RLDSP.IMS1.D04201.T0653127.V00 UNIT=SYSDA
START = 04.201 06:53:12.7 FIRST DS LSN= 00000000000000643
STOP = 04.201 06:53:26.0 LAST DS LSN= 000000000000007A9
FILE SEQ=0001 #VOLUMES=0001

VOLSER=000000 STOPTIME = 04.201 06:53:26.0
CKPTCT=0 CHKPT ID = 04.201 06:52:52.0
LOCK SEQUENCE#= 000000000000

DSN=DUMMY ERROR UNIT=3400
START = 04.201 06:53:26.0 FIRST DS LSN= 0000000000000000
STOP = 04.201 07:53:26.0 LAST DS LSN= 000000000000001F4
FILE SEQ=0001 #VOLUMES=0001

VOLSER=VOL001 STOPTIME = 04.201 07:53:26.0
CKPTCT=0 CHKPT ID = 00.000 00:00:00.0
LOCK SEQUENCE#= 204000000000

LOGALL
START = 04.201 06:52:49.9 *
EARLIEST ALLOC TIME = 04.201 06:53:03.2
DBDS ALLOC=1 -DBD- -DDN- -ALLOC-
DBVHDJ05 CJVHDG1E 1
```

### Deleting Log Records

As time passes, the RECON data set accumulates old records associated with IMS log data sets. You can use the `DELETE.LOG` command to eliminate unnecessary PRILOG families and their associated LOGALL records.

#### Related Reading:
See “DELETE.LOG (for OLDS)” on page 208 and “DELETE.LOG (for RLDS and SLDS)” on page 208 for descriptions of the available `DELETE.LOG` commands and their parameters.

Use the `DELETE.LOG` INACTIVE command periodically to keep the RECON data set free of unnecessary PRILOG family records. Refer to the description of the `DELETE.LOG` command INACTIVE parameter for the conditions that must be met before DBRC deletes a PRILOG family of records. The `DELETE.LOG` command, specified with either the INACTIVE or the TOTIME parameter, does not delete active log records.
Common problems are old open (nonzero stop time) PRILOG records that were created by jobs that terminated abnormally. These jobs abended without closing their logs. You can find these open logs by issuing the LIST.LOG command with the OPEN parameter. You can use the DELETE.LOG command with the STARTIME parameter to remove old, unnecessary, open PRILOG family records.

Here is a list of some other factors that can affect the PRILOG family of records:

- GENMAX
- Frequency of creating DB Image copies
- RECOVPD
- Log Retention Period

Working with Subsystem Records (SSYS)

Have you ever wondered what those subsystem names represent in the RECON data set? If authorization for a database fails and you list the database and all the subsystem records in the RECON data set, you might wonder how the subsystem names (SSIDs) were created. The following sections discuss different aspects of the SSIDs.

Naming Conventions for SSIDs in RECON Subsystem Records

The SSYS record is created when a signon call is issued to DBRC. The SSID must be unique. Normal conventions for creating the subsystem name are:

- IMS and DBCTL subsystems = IMSID value from IMSCTRL SYSGEN macro (four characters)
- BATCH and UTILITY SUBSYSTEMS = JOBNAME
- XRF = RSENAME (Recovery Service Element) of the IMS systems (active and alternate)

Exceptions include the following:

- ONLINE IMAGE COPY - XXXTZZZZ where
  - XXXT is the DMB number of the database + DCB number translated to 0-9,A-Z
  - ZZZZ is the control region IMSID

Online image copy is the only BMP that creates its own subsystem record in the RECON data set.

Batch Backout

If batch backout is processing a log created by a batch subsystem, the SSID that is used is the job name of the subsystem that is being backed out. The job name of the batch backout utility job is not used.

If batch backout is processing a log that was created by an online subsystem, the job name of the backout utility job is used as the SSID.

Deleting a Subsystem Record

When you no longer need a particular subsystem record in the RECON data set, the following commands can be used to delete the SSYS:

```
CHANGE.SUBSYS SSID(XXXX) STARTRCV
CHANGE.SUBSYS SSID(YYYY) ENDRECOV <--(removes auth from DB)
DELETE.SUBSYS SSID(XXXX)
```
Hints and Tips

Subsystem Record Size
The subsystem record grows in size. If the subsystem record grows to exceed the 16M maximum record size, IMS abends.

Removing Authorization Inconsistency Between the SSYS from DB/AREA Records
To change database authorization or if authorization fails, a LIST.DB shows that the DB or area is authorized to a non-existent subsystem and backout is not required. Use the command below to remove the authorization. You can also use the command below to remove a database/area name from the list in a subsystem record when a LIST.SUBSYS shows it authorized and a LIST.DB does not show it authorized.

```
CHANGE.DB DBD(dbname) SSID(XXXX) UNAUTH  (or)
CHANGE.DB DBD(dbname) AREA(areaame) SSID(XXXX) UNAUTH
```

Getting Change Accumulation to Start Processing Logs Again
Change accumulation erroneously stops processing logs and the source of the problem is traced to an OLDS left in the ARC STARTED state. An IPL of z/OS is done on the CPU where the archive job was run and the status of the OLDS remains in this state. GENJCL.ARCHIVE jobs do not include any OLDS in ARC STARTED status. The online IMS that created the OLDS is not running on this CPU.

If the online IMS is run on the same CPU as the archive job when IMS is restarted, it resets the status of all OLDS in ARC STARTED status to ARC NEEDED. The OLDS is included in the next archive job. To avoid this problem, run archive jobs on the same CPU as their corresponding IMS systems.

In cases where manual intervention is required, use the following command to change the status of the OLDS to archive needed (ARNEEDED):

```
CHANGE.PRILOG OLDS(DFSOLPXX) ARNEEDED SSID(XXXX)
```

Getting Change Accumulation Working When It States Nothing to Process
Change accumulation has issued a message stating that it found no logs to process. A review of the listing reveals that there are many logs that need to be processed.

DBRC selects all logs needed to satisfy the allocations for each DBDS since the effective image copy time. The logs are put in log volume start time order as a way for DBRC to keep track of which logs it has processed. DBRC processes all available logs and truncates the list of log volumes when it encounters a log in error, an open log, or when it detects an unarchived OLDS. A message indicating the condition that was encountered (such as an open log) is issued.

If the last change accumulation execution record shows that a SUBSET of logs was input to it last, the STOPTIME reflects the start time of the first unavailable log volume. Find this log volume time stamp in the listing of the RECON data set. If no log volume exists with this start time, there is an unarchived OLDS.

The following time line illustrates this open OLDS condition:
If GENJCL.CA is run at T7, it includes Log1(DSN1) and has a STOPTIME of T4 and is a SUBSET. T4 is the start time of the first unavailable log. After this JCL is executed, any attempts to run GENJCL.CA result in issuing two messages; one stating that an unarchived OLDS exists and another stating that there are no logs to process until the open OLDS is closed and archived.

### Moving Log Data Sets

DBRC records information about IMS logs in the RECON data set. This information includes the log data set name (DSN), its starting and ending times, and the volume serial (VOLSER).

The best way to avoid lost logs is to use cataloged log data sets and the DBRC CATDS option.

For non-cataloged log data sets, inform DBRC about changes by using the CHANGE.PRILOG or CHANGE.SECLOG command.

**Related Reading:** See the CHANGE.PRILOG commands starting with "CHANGE.PRILOG (for OLDS)" on page 166 and the CHANGE.SECLOG commands starting with "CHANGE.SECLOG (for OLDS)" on page 187.

### Cataloging Data Sets

You can indicate whether image copy, change accumulation, and log data sets are cataloged by using the CATDS or NOCATDS keywords on the CHANGE.RECON or INIT.RECON command.

The CHANGE.RECON and INIT.RECON commands update the RECON header record accordingly if you specify either CATDS or NOCATDS. The only difference between the two commands in respect to cataloging is that INIT.RECON can be issued only once for each set of RECON data sets (3) required for an IMS region.

To specify that these data sets are cataloged specify:

```plaintext
//CHGRECON JOB

.. SYSIN DD *
   CHANGE.RECON CATDS
/*

-OR-

//INITRCON JOB

.. SYSIN DD *
   INIT.RECON CATDS
/*
```
The data set must have been initialized as cataloged for CATDS to be effective with the CHANGE.RECON command.

Specifying NOCATDS on the CHANGE.RECON or INIT.RECON command establishes that these data sets, regardless of their cataloged status, are not to be treated as cataloged.

To specify that these data sets are not to be treated as cataloged specify:

```
//CHGRECON JOB
...
//SYSIN DD *
  CHANGE.RECON NOCATDS
/*
```

-OR-

```
//INITRECON JOB
...
//SYSIN DD *
  INIT.RECON NOCATDS
/*
```

Related Reading: See “INIT.RECON” on page 280 and “CHANGE.RECON” on page 177 for more information.

---

Performing Multiple Cold Starts in a Test Environment

Many times in a test environment, you may want to cold start IMS. In order to cold start IMS the last OLDS must be closed. If you have no need to close the OLDS, you can use the following commands to close the OLDS and mark it archived in the RECON data set so it may be reused:

```
NOTIFY.PRILOG STARTIME(042052037526) RUNTIME(042052137526) -
  LASTREC(2218) OLDS(DFSOLP02) SSID(IMS1)
CHANGE.PRILOG OLDS(DFSOLP02) ARCHIVED SSID(IMS1)
```

The entries for each OLDS (such as: DFSOLP00, DSPOLP01, and DSPOLD02) in the PRIOLD record are built when the OLDSs are used (if they do not already exist in the RECON data set). If you also need to delete the OLDS from the RECON data set, the following commands can be used:

```
DELETE.LOG OLDS(DFSOLP00) SSID(IMSA)
DELETE.LOG OLDS(DFSOLP01) SSID(IMSA)
DELETE.LOG OLDS(DFSOLP02) SSID(IMSA) LASTCLOS
```

Note that the LASTCLOS is necessary to delete the last OLDS used by IMS. The PRIOLD record is also deleted when the last DDNAME entry is removed.

The PRIOLD record before issuing the commands:

```
PRIOLD
SSID=IMS1   # DD ENTRIES=3
EARLIEST CHECKPOINT = 04.205 20:37:02.3

DDNAME=DFSOLP00 DSN=IMSTESTL.IMS01.OLDSP0
START = 04.205 20:36:58.7 FIRST DS LSN= 0000000000000001
STOP = 04.205 20:37:29.1 LAST DS LSN= 0000000000000642
LOCK SEQUENCE# = 000000000000
STATUS=ARC COMPLT FEOV=YES AVAIL
PRILOG TIME=04.205 20:36:58.7 ARCHIVE JOB NAME=JT203738
VERSION=9.1
```
Avoiding Some Causes of RECON Data Set Enqueue Problems

The following sections describe some situations that can cause RECON data set enqueue problems and contain suggestions for avoiding these problems.

Related Reading: See “RECON Data Set Serialization” on page 57 for more information.

In a Shared DASD Environment

Operating in a shared DASD environment, the most common cause of RECON data set enqueue problems is failing to follow the recommendation to catalog each RECON data set in its own ICF catalog on the same volume as the RECON data set.
If you catalog each RECON data set in its own ICF catalog on the same volume as the RECON and still have problems; examine your GRS, (or equivalent) RESERVE conversion list, to determine how you process SYSIGGV2 and DSPURI01 QNAMEs. A couple of combinations might lead to deadlocks.

In a Non-Shared DASD Environment

If you are operating in a non-shared DASD environment and are having problems, this is probably not caused by deadlock but rather by contention and slow performance. Here are a few things to look at in this situation:

1. Minimize the amount of time any job holds the RECON data set.
   One way to minimize that time is to tune the LSR buffer pool DBRC uses when accessing the RECON data sets. There is a CSECT, DSPBUFFS, that contains the values used for online, batch, and a procedure in the [IMS V9: Customization Guide] on how to zap it to change the values. The defaults might be low for your usage.

2. Analyze the other contents of the volumes where the RECON data set resides. Consider isolating these volumes, to prevent interference from other I/O activity (and vice versa). Consider placing the RECON s on high performance cached DASD, or perhaps solid state DASD.
# Part 2. DBRC Command Reference

## Chapter 7. DBRC Commands

| DBRC Batch Commands | 119 |
| DBRC Online Commands | 120 |
| DBRC Command Syntax | 120 |
| Separators | 121 |
| Continuation Characters | 121 |
| Comments | 121 |
| Commands | 121 |
| Parameters | 121 |
| How to Read Syntax Diagrams | 122 |
| DBRC Time Stamps | 124 |

## Chapter 8. BACKUP Command

| BACKUPRECON | 131 |
| BACKUPRECON Usage | 131 |

## Chapter 9. CHANGE Commands

| CHANGE Commands In This Chapter | 133 |
| CHANGE.ADS | 133 |
| CHANGE.BKOUT | 134 |
| CHANGE.CA | 136 |
| CHANGE.CAGR | 138 |
| CHANGE.DB | 141 |
| CHANGE.DBDS | 149 |
| CHANGE.DBDSGRP | 157 |
| CHANGE.IC | 159 |
| CHANGE.PART | 161 |
| CHANGE.PRILOG (for OLDS) | 166 |
| CHANGE.PRILOG (for RLDS) | 168 |
| CHANGE.PRILOG (for SLDS and TSLDS) | 172 |
| CHANGE.RECON | 177 |
| CHANGE.SELO (for OLDS) | 187 |
| CHANGE.SELO (for RLDS) | 188 |
| CHANGE.SELO (for SLDS and TSLDS) | 192 |
| CHANGE.SG | 197 |
| CHANGE.SBSYS | 198 |
| CHANGE.UIC | 200 |

## Chapter 10. DELETE Commands

| DELETE Commands In This Chapter | 201 |
| DELETE.ADS | 201 |
| DELETE.ALOC | 202 |
| DELETE.BKOUT | 203 |
| DELETE.CA | 203 |
| DELETE.CAGR | 204 |
| DELETE.DB | 204 |
| DELETE.DBDS | 205 |
| DELETE.DBDSGRP | 206 |
| DELETE.GS | 206 |
| DELETE.IC | 207 |
| DELETE.LOG (for OLDS) | 208 |
| DELETE.LOG (for RLDS and SLDS) | 208 |
| DELETE.PART | 211 |
| DELETE.RECOV | 212 |
| DELETE.REORG | 212 |
| DELETE.SG | 213 |
| DELETE.SBSYS | 214 |
| DELETE.UIC | 214 |

## Chapter 11. GENJCL Commands

| GENJCL Commands In This Chapter | 217 |
| GENJCL.ARCHIVE | 217 |
| GENJCL.CA | 221 |
| GENJCL.CLOSE | 225 |
| GENJCL.IC | 229 |
| GENJCL.OIC | 238 |
| GENJCL.RECEIVE | 243 |
| GENJCL.RECOV | 247 |
| GENJCL.USER | 253 |

## Chapter 12. INIT Commands

| INIT Commands In This Chapter | 259 |
| INIT.ADS | 259 |
| INIT.CA | 260 |
| INIT.CAGR | 261 |
| INIT.DB | 263 |
| INIT.DBDS | 266 |
| INIT.DBDSGRP | 271 |
| INIT.GS | 273 |
| INIT.IC | 274 |
| INIT.PART | 275 |
| INIT.RECON | 280 |
| INIT.SG | 285 |

## Chapter 13. LIST Commands

| LIST Commands In This Chapter | 287 |
| LIST.BKOUT | 287 |
| LIST.CAGR | 288 |
| LIST.DB | 289 |
| LIST.DBDS | 290 |
| LIST.DBDSGRP | 292 |
| LIST.GS | 293 |
| LIST.HISTORY | 294 |
| LIST.LOG (for a PRILOG Family) | 296 |
| LIST.LOG (for a Category of Records) | 298 |
| LIST.RECON | 300 |
| LIST.SBSYS | 302 |

## Chapter 14. NOTIFY Commands

| NOTIFY Commands In This Chapter | 305 |
| NOTIFY.ALOC | 305 |
| NOTIFY.BKOUT | 307 |
| NOTIFY.CA | 308 |
| NOTIFY.IC | 310 |
| NOTIFY.PRILOG (for OLDS) | 313 |
| NOTIFY.PRILOG (for RLDS) | 317 |
| NOTIFY.PRILOG (for SLDS and TSLDS) | 322 |
| NOTIFY.RECOV | 326 |
| NOTIFY.REORG | 328 |
| NOTIFY.SECLOG (for OLDS) | 331 |
Chapter 7. DBRC Commands

This chapter contains a description of the DBRC commands. Use these commands to add to, change, and delete information in the RECON and to generate the JCL and the control statements you need to run the various IMS utilities used in database recovery.

In This Chapter:
- “DBRC Batch Commands”
- “DBRC Online Commands” on page 120
- “DBRC Command Syntax” on page 120

DBRC Batch Commands

Here is a summary of the DBRC commands:
- Use the BACKUP.RECON command to create a backup copy of the RECON.
- Use the CHANGE commands to modify information in the RECON.
- Use the DELETE commands to delete information from the RECON.
- Use the GENJCL commands to generate jobs for the various IMS recovery utilities.
- Use the INIT commands to make the following changes to a RECON:
  - Create entries
  - Create CA groups
  - Register DBDSs or DEDB areas
  - Define global service groups
  - Register HALDB partitions
  - Initialize a RECON
- Use the LIST commands to produce a formatted printout of all or selected parts of the RECON.
- Use the NOTIFY commands to add to the RECON information that is normally written there automatically.
- Use the RESET.GS6 command after an unplanned RSR takeover to remove obsolete recovery information about RSR-covered databases and areas from the original active site RECONs.

You can also issue a variation of some of these commands online using the 
/RMxxxxxx command (see ”DBRC Online Commands” on page 120).

CICS users can execute DBRC commands using the CICS-supplied transaction CDBM, that provides a command interface to DBCTL.

Related Reading:
- See the command chapters for valid parameters and usage notes for each command. See [IMS V9: Command Reference] for details pertaining to the syntax of the online commands (/RMxxxxxx).
- See [CICS Transaction System for z/OS CICS Supplied Transaction, SC34-5992] for a description of CDBM, and [CICS Transaction System for z/OS V2R2 IMS Database Control Guide, SC34-6010], for details of the DBCTL and DBRC commands you can use with the CDBM transaction.
DBRC Online Commands

Certain DBRC commands can be issued from an online IMS environment (/RMxxxxx commands). The online DBRC commands are:

- /RMCHANGE
- /RMDELETE
- /RMGENJCL
- /RMINIT
- /RMLIST
- /RMNOTIFY

Related Reading: See [IMS V9: Command Reference](#) for details on DBRC commands that can be entered online and the supported DBRC parameters (modifiers) that are associated with those commands.

After these commands successfully pass through IMS security, IMS passes these commands to DBRC where they are processed. These commands are processed in a similar manner as the regular DBRC commands of similar name. For example, DBRC processes a /RMCHANGE command in a similar manner as a CHANGE command. The authorization of these online commands can also be controlled in a similar manner as the batch DBRC commands, namely you can use the DBRC Command Authorization exit (DSPDCAX0), a security authorization program like RACF, or both.

Related Reading:
- For more information about DBRC command authorization support, see “Preserving the Integrity of the RECON Data Set with DBRC Command Authorization Support” on page 79.
- For the details of using the DBRC Command Authorization Exit (DSPDCAX0), see the [IMS V9: Customization Guide](#).

DBRC Command Syntax

All DBRC commands adhere to the syntax described here. This syntax is standard, command-language syntax and is similar to that of TSO and Access Method Services.

You can enter commands in uppercase, lowercase, or mixed case format. DBRC translates most command input into uppercase format before processing, regardless of the format that is used. However, DBRC does not translate keyword values and string values into uppercase format. For example, the 'value' portion of the USERKEYS parameter on GENJCL commands, and the UDATA parameter on the NOTIFY.UIC and CHANGE.UIC commands are processed in the exact format in which they are entered.

The following sections discuss additional aspects of the syntax of DBRC commands.

- “Separators” on page 121
- “Continuation Characters” on page 121
Separators
A blank, a comma, or a comment can be interchanged in a command wherever a separator is needed. More than one separator can be used between parameters.

Continuation Characters
Continuation characters are used to continue commands and comments that do not fit on a single line of input.

The two continuation characters used by DBRC are the minus sign (−) and the plus sign (+):

+ Deletes the leading separators from the continued line.
− Does not delete the leading separators from the continued line.

Restriction: A character string enclosed in single quotation marks (for example, ('c...c')) can be continued only with a minus continuation character because separators are meaningful in a character string between single quotation marks. DBRC assumes such a character string to be ended if the end of a line is reached before the ending quotation mark is encountered and the last non-blank character in the line is not a minus continuation character. The maximum length of a character string is 256 characters.

Comments
Comments consist of alphanumeric character strings beginning with the symbols (/*) and ending with the symbols (*/).

A comment is assumed to have ended if the end of a line is reached before the character string (*/) is encountered and if the last character in the line is not a continuation character.

Commands
A command consists of a verb, a modifier, and, in most cases, a list of parameters. Exactly one period (.) follows the verb, with no other characters between the verb and the modifier.

Commands can be entered anywhere in columns 1 through 72 of the DBRC SYSIN input stream. Commands can be continued on multiple lines by entering a continuation character as the last non-blank character of the command line.

Columns 73 through 80 of the SYSIN input stream are ignored.

Parameters
Most DBRC commands have both required and optional parameters.

In general, there are no positional keyword parameters in DBRC commands. The keyword parameters are of the following types:
Command Syntax

A keyword by itself
A keyword with a value:
  keyword(v)
A keyword with a list of values:
  keyword(v1,v2..)
A keyword with a repeating list of values:
  keyword((v1,v2..),(v1,v2..)..)
When you enter a repeating list of values only once for this type of keyword, you can omit the outer set of parentheses like this:
  keyword(v1,v2)

Some commands have mutually exclusive parameters. In the parameter descriptions, a vertical bar (|) separates the mutually exclusive parameters. If you specify more that one of these parameters on the same command, you will not get an error, but DBRC will use the last one specified and ignore the others.

Restrictions: For DBRC parameters:

- Certain keywords require values in a specific form.
- The format for hexadecimal input is X'xxx', where X'x' can be any of the characters 0-9 and A-F.
- Any character can be part of a character string.
- A character string that contains special characters must be enclosed in apostrophes.
- Data set names, data set ddnames, and volume serial numbers can include hyphens.
- Data set names follow the conventions specified in the z/OS JCL manuals.

Except where otherwise noted, optional keywords with values have the following defaults.

| Numeric values | 0 |
| Character values | blank |

If a particular parameter is encountered more than once within a command, the last occurrence of the parameter is used. If mutually exclusive parameters are encountered within a command, the last occurrence is used.

How to Read Syntax Diagrams

The following rules apply to the syntax diagrams that are used in this information:

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
  - The >>>-- symbol indicates the beginning of a syntax diagram.
  - The --> symbol indicates that the syntax diagram is continued on the next line.
  - The >-- symbol indicates that a syntax diagram is continued from the previous line.
  - The -->< symbol indicates the end of a syntax diagram.
- Required items appear on the horizontal line (the main path).
• Optional items appear below the main path.

If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.

• If you can choose from two or more items, they appear vertically, in a stack. If you must choose one of the items, one item of the stack appears on the main path.

• An arrow returning to the left, above the main line, indicates an item that can be repeated.

If the repeat arrow contains a comma, you must separate repeated items with a comma.

A repeat arrow above a stack indicates that you can repeat the items in the stack.

• Sometimes a diagram must be split into fragments. The syntax fragment is shown separately from the main syntax diagram, but the contents of the fragment should be read as if they are on the main path of the diagram.
DBRC Time Stamps

Time stamps are points in time recorded in the RECON. Correctly interpreting time stamp formats found in RECON listings, messages, and dumps will help you enter the appropriate time stamp in commands.

Standard Time Stamp Format

Certain parameters require a time stamp, which may be entered in one of the following formats:

Compressed: \[yydddhhmmsst\] [offset]

Punctuated: \[[yy]yy|ddd|hh|mm|ss|t [offset] \]

where:

\(yyyy\) Is the year (0000 to 9999)
\(ddd\) Is the day (000 to 366)
\(hh\) Is the hour (0 to 23)
\(mm\) Is the minute (0 to 59)
\(ss\) Is the second (0 to 59)
\(t\) Is the tenth of a second (0 to 9)
\(|\) Can be any non-numeric character delimiter including blank except the single quotation mark. If the time stamp contains any blanks, commas, or parentheses, it must be enclosed in single quotation marks. For example,

LIST.LOG STARTIME('94.252 08:24:45.7-8')

or

LIST.LOG STARTIME('94,252 08:24:45.7 PST')

offset Can be one of the following:

1. Omitted. The current TIMEZIN value is used.

Related Reading: See “CHANGE.RECON” on page 177 for more information on the TIMEZIN parameter.
2. A numeric offset in the form ±h[h:mm]] or ±h[h:mm]] that, when added to UTC, gives local time. h[h] is a numeric value from 0 to 14. For the compressed format if mm is specified, then hh must also be specified. mm is a value from the set {00, 15, 30, 45}.

±hh:mm is valid only between the values -11:45 to +14:45.
±hhmm is valid only between the values -1145 to +1445.

3. A symbolic time zone label.

The time stamp value might have elements truncated on the right, in which case the omitted element’s digits are assumed to be zeros.

You can truncate the input at the beginning of any element after ddd; so, yyyy|ddd is acceptable, as is yyyy|ddd|hh. Part of an element cannot be entered: for example, yyyy|ddd|h is invalid.

If only two digits are entered for the year, the two high order digits are extrapolated by using the sliding-window method described in “Extrapolation of Two-digit Year Input” on page 128.

The same time stamp value could be entered in the following ways:

942520824457
942520824457-0800
94.252/08:24:45.7

or with blank, commas, or parentheses:

'94.252 08:24:45.7 -8'     '94/252-08.24.45.7 -8:00'
'94,252 08:24:45.7 PST'    '94/252-16.24.45.7 UTC'
'1994 252 16.24.45.7 +0'    

**TIMEFMT Parameter:** Use the optional TIMEFMT parameter to define the form in which time stamps appear in messages, displays, and listings from DBRC. Each parameter associated with TIMEFMT is optional and can be omitted by including only the comma.

The TIMEFMT parameter can be used on CHANGE.RECON, LIST.xxxx, and GENJCL.xxxx commands or can be specified in a skeletal JCL member.

**Restriction:** The parameters that are associated with TIMEFMT are positional.

The format of the TIMEFMT parameter sublist is as follows:

```
TIMEFMT( offset , offset_display A )
```

A:
Notes:
1 duration can only be set on the CHANGE.RECON command.
2 precision is only valid on the %SET statement.

If items of the sublist are omitted, the current values from the RECON header are used.

offset
Specifies the offset that is to be applied to the UTC internal time before display.
  U None—that is, display UTC when the event occurred.
  O Offset of origin—that is, display local time when and where the event occurred.
  L Current local offset—that is, display the current-local-time equivalent.

offset_display
Specifies the display format of the offset that is appended to the time.
  L Specifies that the offset is to be displayed in label format, if a label has been defined for it. If no label is defined, the offset is displayed in numeric format.
  O Specifies that the offset is to be displayed in the numeric (+|- HH:MM) format.
  N Specifies that no time zone information is to be displayed.

form
Specifies whether the time stamp is to be displayed in punctuated or compressed form.
  P Specifies that the time stamp is to be displayed in punctuated form.
  C Specifies that the time is to be displayed in compressed form.

year_size
Specifies whether all four digits of the year are to be displayed or only the two low-order digits.
  2 Only the units and tens digits of the year are displayed.
  4 All four digits of the year are to be displayed.

duration
Specifies the scope of these choices to be either limited to the current job or used as global overrides to the system defaults. The duration subparameter can only be specified on a CHANGE.RECON command.

PERM
Indicates that the specified options are to be in effect for any subsequent DBRC utility job running with the same RECON; that is, these values become the defaults for subsequent jobs.
**TEMP**

Indicates that the specified options are in effect only for the job in which the command is entered. If neither PERM nor TEMP is coded, TEMP is the default.

**precision**

Coded only on a %SET statement in skeletal JCL as a number from 1 to 6. You can use it to control the number of low-order digits contained in time stamps that are output by GENJCL. The default is 1, which gives tenths of seconds. DBRC normally ignores digits of a lower order than tenths of seconds. Full time-stamp precision is not required nor accepted for IMS utility JCL; it is only intended for GENJCL output generated for user-specified purposes (normally GENJCL.USER).

(As with other parameters, coding a null value causes the corresponding TIMEFMT value to be reset to the GENJCL default. TIMEFMT() resets all values.)

**TIMEFMT in Skeletal JCL:** Besides its use on the CHANGE.RECON command, the TIMEFMT parameter can be coded on any LIST.xxx or GENJCL.xxx DBRC command. It can also be specified in a skeletal JCL member as:

%SET TIMEFMT(....)

Here is an example of the %SET keyword in skeletal JCL:

```plaintext
%SET TIMEFMT(,N)
%SELECT RLDS(SSID, LAST)
LOGEND =%LOGETIM
%ENDSEL
```

And here is what the output from the preceding example of %SET would render:

LOGEND =96011315000

**Related Reading:** See “Using the %SET TIMEFMT Keyword” on page 93 for more examples of %SET output specifications.

**TIMEFMT Subparameter Order of Precedence:** The order of precedence of the TIMEFMT subparameters, from lowest to highest, is as follows:

1. **RECON defaults**
   
   The RECON defaults are established by the INIT.RECON command. These defaults can be reset using the PERM option of the CHANGE.RECON command.

2. **Job-level override**
   
   The RECON defaults can be overridden for the following commands in a SYSIN command stream using either PERM or TEMP options of the CHANGE.RECON command.

3. **%SET statements in skeletal JCL members**
   
   The TIMEFMT settings in a %SET statement override the GENJCL default settings, as well as settings from a previous %SET statement in the same member.

4. **Command override**
   
   The TIMEFMT settings in a DBRC command override all of the above settings within the scope of that command invocation. For example, a GENJCL.xxx command TIMEFMT setting overrides the TIMEFMT settings on any %SET statement in the applicable skeletal JCL member.

When coded on any command other than CHANGE.RECON, the last parameter, duration, has no meaning; any TIMEFMT values coded override the values that are
Command Syntax

currently in effect only for the duration of the command. For any omitted values, the values currently in effect from the most recent CHANGE.RECON command with a PERM duration, which override the INIT.RECON defaults, remain in effect.

Default Settings: The values set in the RECON by the INIT.RECON command are TIMEFMT(O,N,P,2,TEMP).

The defaults used by GENJCL commands are TIMEFMT(O,O,C,2,TEMP,1).

Recommendation: If your installation operates multiple IMS systems with different time zones and they share data and a RECON; use UTC or LOC for RECON listings, at least, so that all the time stamps listed have a common base.

Specifying Zero Time Stamp Values
A zero time stamp is valid only where explicitly allowed for a command keyword value. A zero time stamp value, where permitted, can be entered in either punctuated or compressed format, or can be condensed into a single digit. For example, you can specify a zero time stamp in the following ways:

```
000000000000
'00 000 00:00:00.0'
0
```

If an offset value is given, it is ignored.

Extrapolation of Two-digit Year Input
Input time stamps on DBRC commands are accepted with the two-digit year notation. In this case the century digits are derived for the use of the internal time stamp. The century digits in the internal time stamp are determined from the two entered digits by the following process:

The entered year digits are compared to a sliding window of 100 years. The lower limit of the window is the current year minus 70. The upper limit is the current year plus 29. A “breakpoint” is defined as the two low-order digits of the lower limit.

If the entered digits are equal to, or greater than, the breakpoint; the century digits of the lower limit are used. Otherwise, the century digits of the upper limit are used. The upper and lower limits of the window and the breakpoint are established when IMS is initialized. When the year changes, these values are recomputed.

Time Stamp Conversions and Examples
In the examples that follow, local time is PST (offset -08:00).

An event occurring at 1998.030 10:00:0.1 -08:00 (punctuated format - yyyy ddd hh mm ss) or 98030100001 -0800 (compressed format - yyddhhmmssst) is recorded in 12 bytes in the RECON as 1998030F|18000012|3456032d| (yyyyyddd hhmmssst xxxxxfqqq), where:

- **hh** UTC, 8 hours later than local time
- **x xxxx** Low-order digits ignored by DBRC (except for GENJCL.USER output with precision)
- **fqqq** Offset:
  - **f** Flag bits, normally 0
  - **qq** Quarter hours (32/4=8)
  - **s** Sign (D is negative, C is positive)
**Note:** The offset is always added to UTC to obtain local time. To obtain UTC from local time, reverse the sign and add. The time stamp from a record in the RECON data set, that is PRILOG, lists as:

'1998.030 10:0:00.1 -08:00'

When the time stamp above is entered in a command with the quotes, DBRC finds the record in the RECON data set with the recorded time as:

1998030F 18000012 3456032D

DBRC ignores the low-order time digits and the offset when searching. The UTC date and time is determined.

If no offset is supplied in the time stamp, DBRC uses the TIMEZIN value which might not be correct if a clock change has occurred since the time when the record was written in RECON.

Time stamps from different RECON listings, even with time stamps displayed in different formats, can be used freely, so long as the offsets are included.

**Standard Default Settings for Time-Stamp Values**
The RECON data set time-stamp formats are the standard default.

From the RECON header:

TIMEZIN=%SYS

OUTPUT FORMAT: DEFAULT = LOCORG NONE PUNC YY
No user-defined labels are defined.

Listing an allocation record without offset values lists the time stamp as:

99.027 19:03:47:0

The allocation record was written prior to daylight savings change. The time stamp is entered on a command to list the record without an offset. With TIMEZIN=%SYS set in RECON, the offset is obtained from current z/OS clock which has changed. The record is not found in RECON.

**DBRC Commands Affected by the Time-Stamp Format**
These DBRC commands have time stamps as part of their syntax.

- CHANGE.BKOUT
- CHANGE.CA
- CHANGE.IC
- CHANGE.PRILOG
- CHANGE.SECLOG
- CHANGE.UIC
- DELETE.ALLOC
- DELETE.CA
- DELETE.IC
- DELETE.LOG
- DELETE.RECOV
- DELETE.REORG
- DELETE.UIC
- GENJCL.CA
- GENJCL.RECOV
- GENJCL.USER
- LIST.HISTORY
Command Syntax

- LIST.LOG
- NOTIFYALLOC
- NOTIFYBKOUT
- NOTIFYCA
- NOTIFYIC
- NOTIFYPRILOG
- NOTIFYRECOV
- NOTIFYREORG
- NOTIFYSECLOG
- NOTIFYUIC

**Related Reading:** See [“Standard Time Stamp Format” on page 124](#) for more information on the standard time-stamp format.
Chapter 8. BACKUP Command

Use the BACKUP.RECON command to create backup copies of the RECON from the Copy1 RECON.

In This Chapter:
* "BACKUP.RECON"

BACKUP.RECON

“BACKUP.RECON Usage” contains the details for the BACKUP.RECON command.

BACKUP.RECON Usage

The BACKUP.RECON command first opens the RECON and any needed cleanup is done. The command then invokes the IDCAMS REPRO command, using its normal defaults, to create the backup copy. Any restrictions applicable to the normal use of the REPRO command apply to this command. The data set receiving the backup copy must be empty.

If your RECON RECORDSIZE is greater than 32 KB, the BACKUP.RECON command will fail if the back up is to a sequential data set. Perform the following steps to back up a RECON to a sequential file (for example, to a tape):

1. Create a backup KSDS using the BACKUP.RECON command
2. Use DFSMSdss™ to copy the backed-up KSDS to the sequential file

BACKUP.RECON Syntax

```
BACKUP.RECON RECON1 | RECON2 | BOTH
```

Parameters for BACKUP.RECON

RECON1 | RECON2 | BOTH

A required parameter that specifies the backup data set to which the RECON is copied.

RECON1

Copies RECON to the backup data set specified by the BACKUP1 DD statement of your JCL.

RECON2

Copies RECON to the backup data set specified by the BACKUP2 DD statement of your JCL.

BOTH

Copies RECON to the data sets specified by the BACKUP1 and BACKUP2 DD statements of your JCL.

Example of Creating Backups of a RECON

In this example, two backup copies of the Copy1 RECON are created.
//BKUP JOB
//BACKUP1 DD . .
//BACKUP2 DD . .

:.
//SYSIN DD *
   BACKUP.RECON BOTH
/*

To create a sequential backup, the BACKUPx DD statement must include appropriate DCB parameters. The BLKSIZE specified must be larger than the maximum RECORDSIZE defined in the RECON, but less than 32K. For example, DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760).
Chapter 9. CHANGE Commands

Use the CHANGE commands to modify information in a RECON.

CHANGE Commands In This Chapter

The following is a list of all the CHANGE commands:

- “CHANGE.ADS”
- “CHANGE.BKOUT” on page 134
- “CHANGE.CA” on page 136
- “CHANGE.CAGRP” on page 138
- “CHANGE.DB” on page 141
- “CHANGE.DBDS” on page 149
- “CHANGE.DBDSGRP” on page 157
- “CHANGE.IC” on page 159
- “CHANGE.PART” on page 161
- “CHANGE.PRILOG (for OLDS)” on page 166
- “CHANGE.PRILOG (for RLDS)” on page 168
- “CHANGE.PRILOG (for SLDS and TSLDS)” on page 172
- “CHANGE.RECON” on page 177
- “CHANGE.SECLOG (for OLDS)” on page 187
- “CHANGE.SECLOG (for RLDS)” on page 188
- “CHANGE.SECLOG (for SLDS and TSLDS)” on page 192
- “CHANGE.SG” on page 197
- “CHANGE.SUBSYS” on page 198
- “CHANGE.UIC” on page 200

CHANGE.ADS

Use the CHANGE.ADS command to change DEDB ADS information in RECON. The CHANGE.ADS command fails if you issue it while the area is in use. To create an ADS entry, refer to “INIT.ADS” on page 259.

CHANGE.ADS Syntax

```
CHANGE.ADS ADDN(name) AREA(name) DBD(name) ADDNNEW(name)
```

Parameters for CHANGE.ADS

ADDN(name)

Required parameter you use to specify the area data set ddname of the ADS being changed.
CHANGE.ADS

AREA(name)
   Required parameter you use to identify the ADS being changed, by area name.

DBD(name)
   Required parameter you use to identify the ADS being changed, by database name

ADDNNEW(name)
   Optional parameter you use to identify the ADS being changed, by new ddbname.

ADSN(name)
   Optional parameter you use to identify the ADS being changed, by new dataset name.

AVAIL | UNAVAIL
   Mutually exclusive, optional parameters you use to change the ADS record to indicate its availability.

   AVAIL
   Indicates that the ADS is available. The CHANGE.ADS AVAIL command fails if the area needs to be recovered.

   UNAVAIL
   Indicates that the ADS is unavailable.

   If neither AVAIL nor UNAVAIL is specified but ADSN is specified, the value defaults to UNAVAIL.

Example of Changing an ADS Record
In this example, an ADS record in RECON is being changed.

//CHGADS   JOB
;
//SYSIN    DD  *
   CHANGE.ADS DBD(DBD001) AREA(AREA002) -
         ADSN(ADSN004) ADDN(ADDN004)
/*

CHANGE.BKOUT

Use the CHANGE.BKOUT command to add, change, or delete a unit of recovery (UOR) in the backout record that is associated with a specified subsystem.

Recommendation: Use the CHANGE.BKOUT command when a dynamic backout failure has occurred and certain known backout records are invalid. Using the CHANGE.BKOUT command incorrectly can result in a loss of recovery integrity.

CHANGE.BKOUT Syntax

CHANGE.BKOUT—SSID(name)—UOR(uor)—UORTIME(time_stamp)

DELETE(UOR—ALL—name)

PSB(name)

DBD(name)—name
Parameters for CHANGE.BKOUT

SSID(name)
Required parameter that specifies the subsystem for which the backout record is to be changed. The name is an alphanumeric string, with up to eight characters, that represents any valid subsystem name.

UOR(uor)
Required parameter that you use in conjunction with the UORTIME parameter to identify a unit of recovery in the backout record. The recovery token (uor) is 16-byte field that describes a specific UOR in the backout record. The value for this parameter must be expressed as 32 hexadecimal digits.

This parameter can specify a unit of recovery that currently exists in the backout record or one that is to be added to the record.

UORTIME(time_stamp)
Required parameter that specifies the time of the UOR described above. The value is the beginning time of the UOR (found in the X'5607' log record). The time stamp must be in standard form.

DELETE(UOR | ALL | name...)
Optional parameter used to delete some or all of the information related to the unit of recovery that is specified by the required parameters described above.

UOR
Deletes the entire UOR defined by the required UOR and UORTIME parameters described above. If you do not specify DELETE(UOR), CHANGE.BKOUT assumes you are changing an existing UOR or adding a UOR that is not currently in the backout record.

If you specify DELETE(UOR), all other optional parameters are ignored.

If the UOR does not exist in the backout record, the command fails.

ALL
Specifies that the database entries for the specified UOR and UORTIME are deleted, but that the UOR prefix information is left intact, if you also specify database names in the DBD parameter, the BKO parameter, or both. If you do not specify database names in DBD or BKO, CHANGE.BKOUT acts as if DELETE(UOR) is specified. You can use the ALL option to replace, or substantially alter, a database list within a UOR entry of a backout record without disturbing the control data in the UOR’s prefix. You can also use the ALL option to delete all database entries in the UOR except those that are listed in the DBD or BKO parameters.

name
Specifies up to eight database names for use with the DELETE parameter. Use a comma to separate each specified name. If you list all of the databases associated with the specified unit of recovery, CHANGE.BKOUT acts as if DELETE(ALL) is specified.

If any listed database name is not in the specified UOR, the command fails.

HALDB Restriction: When using the CHANGE.BKOUT command for HALDBs, name specifies the name of a HALDB partition.
The following optional parameters can only be used if you do not specify DELETE(UOR). If the UOR already exists in the backout record, you must provide at least one of the optional parameters. If the UOR does not exist in the backout record, it is added. In this case, you must specify the PSB and either the DBD parameter or the BKO parameter.

You can specify either the BKO parameter, the DBD parameter, or both. However, the same database name cannot appear in both the BKO and the DBD parameters, because a database cannot be both backed out and require a backout at the same time.

**PSB(name)**
Optional parameter that identifies the PSB associated with the UOR. To add a UOR to the backout record, you must specify PSB(name). If the UOR defined by the required parameters already exists in the backout record, the specified PSB name replaces the current PSB name.

**DBD(name...)**
Optional parameter that identifies databases associated with the specified UOR. Up to eight database names can be listed with the DBD parameter. The database names listed here identify the databases that require backout for this unit of recovery. This parameter can be used to change the status of an existing database entry to backout required.

**HALDB Restriction:** When using the CHANGE.BKOUT command for HALDBs, *name* specifies the name of a HALDB partition.

**BKO(name...)**
Optional parameter that identifies databases to which the UOR applies. Use BKO to identify databases that have already been backed out from this UOR. Up to eight database names can be specified with the BKO parameter. This parameter can be used to change the status of an existing database entry to backout completed.

**HALDB Restriction:** When using the CHANGE.BKOUT command for HALDBs, *name* specifies the name of a HALDB partition.

**Example of Using the CHANGE.BKOUT Command**

```
//CHGBKOUT JOB
:
//SYSIN DD *
CHANGE.BKOUT SSID(SYS3)—UOR(E2E8E2F34040400000006000000003)—UORTIME(930931345027) DELETE(ALL)—DBD(DATA1,DATA2,DATA3C)—BKO(DATA4,DATA5,DATA3A)
/*
```

**CHANGE.CA**

Use the CHANGE.CA command to change information about a specified run of the Change Accumulation (CA) utility for an identified CA group in RECON.

**CHANGE.CA Syntax**

```
CHANGE.CA—GRPNAME(name)—RECTIME(time_stamp)—CASN(name)
```
Parameters for CHANGE.CA

GRPNAME(name)
Required parameter you use to specify the name of the CA group for which information is to be changed.

RECTIME(time_stamp)
Required parameter you use to identify the change accumulation run record that you are changing.

Use the STOP time marked with an asterisk (*) from the listing of the CA record. The time stamp must be in standard form (see "Standard Time Stamp Format" on page 124).

CADSN(name)
Optional parameter you use to specify the new name of the change accumulation data set in the identified record.

FILESEQ(value)
Optional parameter you use to specify a new file sequence number that is to be recorded in the identified record.

INVALID | VALID
Mutually exclusive, optional parameters you use to specify whether a change accumulation data set is to be used as input for a subsequent run of change accumulation or database recovery.

INVALID
Specifies that the change accumulation data set is not to be used as input for a subsequent run of change accumulation or database recovery. If an invalidated change accumulation data set is subsequently reused for output by change accumulation, it is automatically marked as valid and is used. See "INIT.CAGRP" on page 261 for an explanation of the REUSE parameter.

VALID
Specifies that the previously invalidated change accumulation data set is available for use as input to a subsequent run of change accumulation or database recovery. Use this parameter only if the change accumulation data set was previously marked as invalid and is now valid.

Specifying the INVALID parameter causes the STOPTIME and RUNTIME of the change accumulation record to be swapped. This prevents duplicate records in the RECON. Specifying the VALID parameter causes the STOPTIME and RUNTIME to be swapped back.

UNIT(3400 | unittype)
Optional parameter you use to change the unit type of the volumes on which the change accumulation data set resides. The unit type can be up to eight alphanumeric characters long.
VOLLIST(volser)
Optional parameter that can be listed, that you use to replace the volume serial numbers of the change accumulation data set in the specified change accumulation run record. You can substitute up to 255 volume serial numbers in the variable field; each can be up to six alphanumeric characters long.

SUBSET | COMP
Mutually exclusive, optional parameters you use to indicate the change accumulation status.

SUBSET
Indicates that when the CA was created, a subset of logs were processed and the CA’s stop time is the start time of the first unprocessed log volume.

COMP
Indicates that when the CA was created, a complete set of logs were processed and the CA’s stop time is the stop time of the last log volume that was processed.

You do not need to use this parameter under normal conditions. Checking is not done to verify that the use of this parameter is consistent with the value of the CA stop time. This parameter value is used by the GENJCL.CA and GENJCL.RECOV processes. Incorrect use can result in invalid generated JCL.

Example of Changing a Change Accumulation Run Record
In this example, a change accumulation run record in RECON is being changed. The INVALID parameter indicates that the identified data set is not to be used as input to a subsequent run of a utility. The VOLLIST, FILESEQ, and CADSN parameters indicate additional fields in the record that are to be changed.

```
//CHGCA JOB

; ;
//SYSIN DD *
 CHANGE.CA GRPNAME(CAGRP2) RECTIME(820650204335) -
      INVALID CADSN(IMS.CAGRP2.CA.CADSN2) -
      VOLLIST(VOLCA1) FILESEQ(4)
/*
```

CHANGE.CAGRP
Use a CHANGE.CAGRP command to modify information contained in a specified CA group record in RECON. You can change the names of DBDSs that are members of a CA group with this command.

Restriction: HALDB, ILDS, and PHIDAM Primary Index DBDSs are not recoverable and changes to them are not logged. The CHANGE.CAGRP command does not support these data sets.

CHANGE.CAGRP Syntax
Parameters for CHANGE.CAGRP

**GRPNAME**(name)
Required parameter you use to specify the name of the CA group whose record you want to modify.

**ADD**(dbname,ddname)  |  **DELETE**(dbname,ddname)
Mutually exclusive, optional parameters you use to specify the member you want to add to or delete from the specified CA group.

**ADD**
Specifies members you are adding to the identified CA group. A group cannot have more than 32767 members.

*Restriction:* User-recoverable databases cannot be added to a CA group.

**DELETE**
Specifies members you are deleting from the identified CA group. When you have deleted a member from a CA group, DBRC has no knowledge of previous change accumulation activities on that DBDS.

Specify a list of one or more members in the variable field; each member is a pair of names enclosed in parentheses. dbname is the member’s database name. For HALDBs, dbname specifies the name of a HALDB partition. ddname is the symbolic name of the DD statement.

If you delete all the members of a group, the record of that group is deleted from RECON.

**CAJCL**(member)
Optional parameter you use to change the name of a member of a partitioned data set of skeletal JCL. The member is used to generate the JCL for a run of the Change Accumulation utility when you issue a GENJCL.CA command for the specified CA group.

**DEFLTJCL**(member)  |  **NODEFLT**
Mutually exclusive, optional parameters you use to specify the implicit skeletal JCL default member for the CA group.

**DEFLTJCL**
Specifies an implicit skeletal JCL default member for the CA group. GENJCL.CA uses the default member you specify in order to resolve keywords you have defined. For more information, see "GENJCL.CA" on page 221.

**NODEFLT**
Specifies that no skeletal JCL default member is to be used for the CA group.

**GRPMAX**(value)
Optional parameter you use to modify the number of change accumulation data sets to be maintained for the specified CA group. The value you substitute in the variable field must be a decimal number from 2 to 1024.

*Restriction:* User-recoverable databases cannot be added to a CA group. If you are increasing the GRPMAX value and REUSE is specified, you should use the INIT.CA command to add additional change accumulation data sets. If
the number of data sets does not equal GRPMAX, reuse does not take place and you eventually run out of available data sets for the utility.

**NOREUSE | REUSE**

Mutually exclusive, optional parameters you use to indicate whether change accumulation data sets should be reused.

**NOREUSE**
Indicates that change accumulation data sets that were already used for the specified CA group cannot be reused as output data sets in subsequent runs of the Change Accumulation utility. Any existing, unused change accumulation run records for the specified CA group are deleted when you specify the NOREUSE keyword.

**REUSE**
Indicates that change accumulation data sets that were already used for the specified CA group can be reused as output data sets in subsequent runs of the Change Accumulation utility.

If GRPMAX is higher than the number of existing data sets for the group, you should use the INIT.CA command to add additional data sets; otherwise reuse does not take place. See the explanation for GRPMAX.

For additional information about reusing change accumulation data sets, see the explanation of the REUSE parameter “INIT.CAGRP” on page 261.

**Examples of Using the CHANGE.CAGRP Command**

Here are several examples of things you can do using the CHANGE.CAGRP command.

**Example of Adding DBDSs to the Existing CA Group CAGRP1:** In this example, the DBDSs identified by the dbname and ddname parameters are to be added to the existing CA group, CAGRP1.

```
//CHGCAGRP JOB
;
//SYSIN DD *
  CHANGE.CAGRP GRPNAM(CAGRP1) ADD((DB1,DD1),(DB2,DD2))
```

**Example of Deleting DBDSs from the CA Group CAGRP1:** In this example, the DBDSs identified by the dbname and ddname parameters are to be deleted from the CA group, CAGRP1.

```
//CHGCAGRP JOB
;
//SYSIN DD *
  CHANGE.CAGRP GRPNAM(CAGRP1) DELETE((DB3,DD3),(DB4,DD4))
```

**Example of Changing a CA Group Record:** In this example, a CA group record in RECON is being changed. The changes are made to prevent the reuse of CA data sets by the Change Accumulation utility. It also renames the member of the partitioned data set of skeletal JCL that is used for generating the JCL that is needed for the Change Accumulation utility for this CA group.

```
//CHGCAGRP JOB
;
//SYSIN DD *
  CHANGE.CAGRP GRPNAM(CAGRP1) NOREUSE CAJCL(JCLCA)
/*
CHANGE.DB

Use a CHANGE.DB command to change the information about a database or a Fast Path DEDB area. This information is contained in a database record or area record in the RECON. If you specify the parameters SHARELVL, TYPEFP, or TYPEIMS while the database or an area of the DEDB is in use, the command fails.

You can also use CHANGE.DB to remove a rare authorization inconsistency between the subsystem (SSYS) record and the database or area (DB/AREA) record. This inconsistency occurs when either the SSYS record still has an entry for the DB/AREA in its authorized databases/areas list but the DB/AREA record no longer contains the SSID entry in its associated subsystem information list or the SSID entry is still in the DB/AREA but the SSYS record either no longer exists in RECON or no longer contains an entry for the DB/AREA. Use the AUTH parameter with the syntax shown above.

Restrictions:

- The following restrictions apply when you use the UNAUTH parameter:
  - If any parameters other than DBD, AREA, SSID, and ACTIVE | TRACKING are specified for UNAUTH, the command fails.
  - If AREA, ACTIVE, or TRACKING is specified without UNAUTH, the command fails.
  - If the inconsistency between the SSYS and DB/AREA records as described above does not exist, the command fails.
  - If the ACTIVE or TRACKING parameter do not match the SS ROLE field in the associated subsystem information entry of the specified database or area, the command fails.
- When issuing a CHANGE.DB command during a HALDB online reorganization:
  - CHANGE.DB can only be specified for a HALDB partition (TYPE=PART).
  - HALDB online reorganization must have already been attempted for this partition (the M-through-V data sets must exist).
  - The ALL parameter can only be used when issuing a CHANGE.DB ALL OLRCAP command. All HALDBs that are available to be marked as online reorganization capable are marked “OLRCAP.” Those unavailable for change (such as HALDBs that are currently authorized) are skipped and are not marked as OLRCAP.
- If you change a VSO DEDB from SHARELVL (2 or 3) to SHARELVL (0 or 1), MAS is reset to NOMAS. See [“Parameters for CHANGE.DBDS” on page 151] for a description of MAS and NOMAS.

CHANGE.DB Syntax

Use one of the following two diagrams for the CHANGE.DB command.
CHANGE.DB

- PINIT - NOPINIT
- READON - READOFF
- SHARELVL(0, 1, 2, 3)
- NONRECOV - RECOVABL - USERRCOV
- ICREQ - NOICREQ
- TYPEFP - TYPEIMS
- GSGNAME(gsgname)
- NOTCOVER
- RCVTRACK
- DBTRACK
- PARTSEL(pgmname)
- HIKEY
- (1)
- REORGI - NOREORGI

Notes:
1. Applies only to full function databases. Not valid with the ALL parameter.

Or:

CHANGE.DB - UNAUTH - DBD(name) - SSID(name) - AREA(name)

Notes:
1. Required if using this command for a Fast Path DEDB

Parameters for CHANGE.DB

DBD(name) | ALL
Mutually exclusive, optional parameters you use to identify the database for which the record is to be changed.

DBD
Specifies that you are changing the record of a single database.

Restrictions for HALDBs: For HALDBs, name specifies the name of a HALDB partition or the HALDB master (if you want to change all partitions of the HALDB master). For HALDBs, you can use the CHANGE.DB command only as defined in Table 9 on page 142.

Table 9. Parameter Definitions for HALDB Partition and Master

<table>
<thead>
<tr>
<th>CHANGE.DB</th>
<th>name = TYPE=HALDB</th>
<th>name = TYPE=PART</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTH</td>
<td>NOAUTH</td>
<td>Yes (Global)</td>
</tr>
<tr>
<td>IREQ</td>
<td>NOICREQ</td>
<td>Yes</td>
</tr>
<tr>
<td>NOBACK</td>
<td>BACKOUT</td>
<td>No</td>
</tr>
<tr>
<td>NORECOV</td>
<td>RECOVABL</td>
<td>Yes</td>
</tr>
<tr>
<td>OLRCAP</td>
<td>OLRNOCAP</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Notes: 1. Applies only to full function databases. Not valid with the ALL parameter.

Or:

CHANGE.DB - UNAUTH - DBD(name) - SSID(name) - AREA(name)

Notes: 1. Required if using this command for a Fast Path DEDB

Parameters for CHANGE.DB

DBD(name) | ALL
Mutually exclusive, optional parameters you use to identify the database for which the record is to be changed.

DBD
Specifies that you are changing the record of a single database.

Restrictions for HALDBs: For HALDBs, name specifies the name of a HALDB partition or the HALDB master (if you want to change all partitions of the HALDB master). For HALDBs, you can use the CHANGE.DB command only as defined in Table 9 on page 142.

Table 9. Parameter Definitions for HALDB Partition and Master

<table>
<thead>
<tr>
<th>CHANGE.DB</th>
<th>name = TYPE=HALDB</th>
<th>name = TYPE=PART</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTH</td>
<td>NOAUTH</td>
<td>Yes (Global)</td>
</tr>
<tr>
<td>IREQ</td>
<td>NOICREQ</td>
<td>Yes</td>
</tr>
<tr>
<td>NOBACK</td>
<td>BACKOUT</td>
<td>No</td>
</tr>
<tr>
<td>NORECOV</td>
<td>RECOVABL</td>
<td>Yes</td>
</tr>
<tr>
<td>OLRCAP</td>
<td>OLRNOCAP</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 9. Parameter Definitions for HALDB Partition and Master (continued)

<table>
<thead>
<tr>
<th>CHANGE.DB</th>
<th>name = TYPE=HALDB</th>
<th>name = TYPE=PART</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLRRGON</td>
<td>OLRRGOFF</td>
<td>No</td>
</tr>
<tr>
<td>OLRMSID(name) NOOWNER</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>OLRDBDS(A</td>
<td>M)</td>
<td>No</td>
</tr>
<tr>
<td>PARTSEL</td>
<td>HIKEY</td>
<td>Yes</td>
</tr>
<tr>
<td>PINIT</td>
<td>NOPINIT</td>
<td>Yes</td>
</tr>
<tr>
<td>READON</td>
<td>READOFF</td>
<td>No</td>
</tr>
<tr>
<td>REORGI</td>
<td>NOREORGI</td>
<td>Yes</td>
</tr>
<tr>
<td>SHARELVL</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>TYPEFP</td>
<td>TYPEIMS</td>
<td>No</td>
</tr>
<tr>
<td>GSGNAME</td>
<td>NOTCOVER</td>
<td>Yes</td>
</tr>
<tr>
<td>RCVTRACK</td>
<td>DBTRACK</td>
<td>Yes</td>
</tr>
<tr>
<td>UNAUTH</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Restriction:** If you specify the UNAUTH parameter, you must specify the DBD name. The ALL parameter is not valid with UNAUTH.

**ALL**
- Specifies that you are changing all the databases registered in RECON.
- If any of the databases are HALDB masters or partitions, and you have specified any of the restricted parameters (for example, the PINIT or NOPINIT parameters), a warning message is issued and command processing continues.

**AUTH | NOAUTH**
- Mutually exclusive, optional parameters you use to specify whether the database is authorized to participate in data sharing.
  - **AUTH**
    - Indicates that authorization processing for data sharing is permitted for the database.
  - **NOAUTH**
    - Indicates that authorization processing for data sharing is prohibited for the database.

**GSGNAME(gsgname) | NOTCOVER**
- Mutually exclusive, optional parameters you use to assign the remote site recovery attributes of a DL/I database, including HALDBs.
  - **GSGNAME**
    - Assigns the database to a global service group (GSG).
  - **NOTCOVER**
    - Discontinues remote site recovery for the database.
- You cannot use CHANGE.DB to change the state of a database from non-RSR-covered to RSR-covered at a remote site. Message DSP1044I is issued if you attempt to change the covered state of the database with this command. In order to change a database not covered at a remote site, delete and reinitialize the database. Neither Fast Path DEDBs nor non-recoverable databases can be assigned to a GSG. Neither GSGNAME nor NOTCOVER can be specified while the database is in use.
ICREQ | NOICREQ
Mutually exclusive, optional parameters used to specify whether DBRC should enforce the requirement of taking an image copy of a non-recoverable or user-recoverable database after it has been loaded or reorganized. This option affects all DBDSs within the DB.

ICREQ
Used to reset the IMAGE COPY NEEDED option. When this option is active, DBRC will set IC NEEDED ON in the DBDS record and increment the IMAGE COPY NEEDED COUNT in the DB record following initial load (PROCOPT=L) or a reorganization of the DBDS.

NOICREQ
Used to disable the IC NEEDED requirement. DBRC will not set IC NEEDED ON in the DBDS record or increment the IMAGE COPY NEEDED COUNT in the DB record following initial load or a reorganization of the DBDS. The IC NEEDED flag can still be set manually with the CHANGE.DB ICON command.

Restrictions:
- If the DB is recoverable, NOICREQ cannot be specified without also specifying NONRECOV or USERRCOV.
- NOICREQ will reset all DBDS IC NEEDED values to OFF and the DB IMAGE COPY NEEDED COUNT to 0.
- If the DB is changed to RECOVABL, this option is reset to ICREQ and IC NEEDED is set ON in all DBDSs; and the DB IMAGE COPY NEEDED COUNT is set accordingly.
- ICREQ | NOICREQ cannot be specified with the ALL parameter.

NOBACK SSID(name) | BACKOUT(value) SSID(name)
Mutually exclusive, optional parameters you use to specify whether the database needs backout by any subsystem. Do not use these parameters for a DEDB.

NOBACK
Indicates that the specified subsystem does not need to back out the database. You use this parameter to delete backout information in the specified database record.

If the held AUTH state and ENCODED state are zero, and if the BACKOUT-NEEDED flag is on, using the NOBACK parameter causes the associated subsystem information to be deleted from the database record.

BACKOUT
Indicates that the specified subsystem needs to back out the database the specified number of times. You need to specify the subsystem name with the SSID parameter. If you do not specify the SSID parameter with the BACKOUT parameter, this command fails.

SSID(name)
Required parameter specifying which subsystem encountered the backout errors.

name is any valid subsystem name.

With UNAUTH, SSID indicates which entry is to be removed from the associated subsystem information list of the database specified with the DBD parameter, or which SSYS record is to be changed by removing the specified DB/AREA from the authorized databases/areas list.
SSID is required with the NOBACK, BACKOUT or UNAUTH parameters
If SSID is not included with NOBACK, BACKOUT or UNAUTH, the
command fails.

NONRECOV | RECOVABL | USERRCOV
Mutually exclusive, optional parameters used to specify whether update logs
of the specified database are recorded in the RECON data set.

NONRECOV
Specifies that no recovery is to be performed on the database.

RECOVABL
Specifies that the database is recoverable and all updates performed for the
DBDSs or areas are to be registered in the RECON data set.

USERRCOV
Specifies that update allocations for the Fast Path DEDB are not to be
recorded in the RECON data set. Database updates are logged by IMS, but
DBRC cannot be used to manage recovery for the database. Recovery is
allowed with JCL manually generated by the user. This parameter is valid
only for Fast Path DEDBs.

Notes for USERRCOV:
1. USERRCOV should only be used if a critical need to reduce database
   open time exists and recoverability is require.
2. USERRCOV requires the user to provide their own manually-created
   JCL for change accumulation and recovery.

Restrictions for NONRECOV, RECOVABL, and USERRCOV:
• HALDB master databases are supported by these parameters, but HALDB
  partitions are not.
• These parameters are not valid when ALL is specified with the CHANGE.DB
  command.
• You cannot take concurrent image copies of nonrecoverable or
  user-recoverable databases.
• If the database is registered as RECOVABL or USERRCOV, VIO data sets
  cannot be used for the output log (IEFRDER) in any job that updates the
  database. Temporary log data sets, such as VIO, are deleted at job
  termination, and therefore are not usable for recovery.
• NONRECOV or USERRCOV cannot be specified for RSR-covered databases.
• NONRECOV does not support DEDBs with SDEPs. For a more detailed
  description of this restriction, see the “Non-Recovery Option” section in the

OLRCAP | OLRNOCAP
Mutually exclusive, optional parameters that specify if the HALDB can be
reorganized online using HALDB OLR. There is no default and these
parameters are valid only when the DBD is a HALDB master.

OLRCAP
Specifies (in the master database record and the partition
database record) that the HALDB database is capable of being
reorganized online.

OLRNOCAP
Specifies (in the master database record and the partition
database record) that the HALDB database is not capable of
being reorganized online. If the HALDB OLR cursor is active
or if any M-through-V and Y DBDSs are active, the command
fails. If the command completes successfully and the
M-through-V and Y DBDSs and all of their associated records exist in the RECON data set, the M-through-V and Y DBDSs and all of their associated records are deleted from the RECON data set for all partitions.

OLRIMSID

Mutually exclusive, optional parameters that specify whether a HALDB OLR is in progress. These parameters are valid only for a HALDB partition.

OLRIMSID(name)

Mutually exclusive, optional parameters that specify or change the name of the IMS that owns the HALDB OLR.

OLRDBDS(A | M)

Optional parameter that indicates the active set of DBDSs for this partition is either A-through-J and X or M-through-V and Y.

A
Specifies that the A-through-J and X data sets are active.

M
Specifies that the M-through-V and Y data sets are active.

PINIT | NOPINIT

Mutually exclusive, optional parameters used to specify whether a HALDB partition needs to be initialized. Use this parameter after deleting and redefining one or more partition data sets without changing the database definition.

If the database specified by DBD parameter is a HALDB master, the change is made to all partitions. Otherwise, only the partition that is specified is changed.

Restriction: The PINIT | NOPINIT parameters are not valid with ALL.

PINIT
Indicates that the partition needs to be initialized using the DB Prereorganization utility or the HALDB Partition Data Set Initialization utility.

NOPINIT
Indicates that the partition does not need to be initialized.

RCVTRACK | DBTRACK

Mutually exclusive, optional parameters you use to specify the type of RSR tracking (shadowing) for DL/I databases that are assigned to a GSG.

RCVTRACK
Indicates recovery-readiness tracking.
DBTRACK
Indicates database-readiness tracking.

Neither RCVTRACK nor DBTRACK can be specified for Fast Path DEDBs. For DL/I databases, RCVTRACK or DBTRACK can only be specified if the database is assigned to a GSG and is not currently in use, including all the partitions of a HALDB.

READON | READOFF
Mutually exclusive, optional parameters you use to specify whether the database can be restricted to read only processing only. Do not use either parameter for a DEDB.

READON
Specifies that the database can be authorized only for read processing.

READOFF
Specifies that the database can be authorized for both read processing and update processing.

SHARELVL(0 | 1 | 2 | 3)
Optional parameter you use to specify the level of data sharing for which authorized subsystems can share a database. You cannot specify this parameter for authorized DL/I databases.

The numbers 0, 1, 2, and 3 define the four types of data sharing levels.

0 Indicates that the database cannot be shared.

1 Indicates that the database can be shared by one IMS subsystem authorized for update and other IMS subsystems authorized only for read processing (no integrity processing). 1 can also indicate that the database can be shared by multiple IMS subsystems that have been authorized only for read processing. Level 1 is known as database-level sharing.

2 Indicates that the database can be shared by multiple, concurrent subsystems that have been authorized for update in a single-host processor environment. Level 2 is known as intrahost, block-level sharing.

3 Indicates that the database can be shared by multiple, concurrent subsystems that have been authorized for update in a multiple-host processor environment. Level 3 is known as interhost, block-level sharing.

For more information on data sharing levels and dynamic allocation, see [IMS V9: Utilities Reference: System]

Restrictions:

- The SHARELVL parameter must be greater than 0 for concurrent image copies.
- If you are using IRLM, and have specified SHARELVL 2 or 3, ensure that the VSAM SHAREOPTIONS (3 3) parameter is also specified.
- For more information on coordinating VSAM data set definitions with share options, see [IMS V9: Administration Guide: System]
- The SHARELVL parameter applies to all areas in the DEDB.
- If you change a DEDB from level 0 or 1 to level 2 or 3, the first coupling facility structure name (CFSTR1) for all VSO areas in the DEDB is set to the
name of the area. If you change a DEDB from level 2 or 3 to level 0 or 1, DBRC resets any specified coupling facility structure names to zeros, resets the LKASID parameter to NOLKASID, and resets the MAS parameter to NOMAS. See “CHANGE.DBDS” on page 149 for explanations of the CFSTR1, LKASID, NOLKASID, MAS, and NOMAS parameters.

**TYPEFP | TYPEIMS**

Mutually exclusive, optional parameters you use to change the RECON record structure to a Fast Path DEDB or a DL/I database.

**TYPEFP**

Specifies that the database is a Fast Path DEDB and that the record structure in RECON must be changed from IMS to Fast Path. TYPEFP cannot be specified for an RSR-covered DL/I database.

**TYPEIMS**

Specifies that the database is a DL/I database and that the record structure in RECON must be changed from Fast Path to IMS. TYPEIMS cannot be specified if any area of a DEDB is covered by RSR.

**UNAUTH**

Removes an entry from the associated subsystem information list in the database specified by the DBD parameter, or removes an entry from the authorized databases/areas list in the SSYS record specified by the SSID parameter. You must specify the following parameters when you use UNAUTH:

- **DBD(name)**: For the database name
- **AREA**: If the database is a Fast Path DEDB
- **SSID(name)**: For the IMS subsystem ID
- **TRACKING**: If the IMS subsystem is an RSR tracking subsystem

**AREA(name)**

Required when UNAUTH is specified for a Fast Path DEDB. The name is the name of the DEDB area. If you specify AREA without UNAUTH, the command fails.

**ACTIVE | TRACKING**

Indicates the role of the specified subsystem when UNAUTH is specified. These parameters are ignored unless UNAUTH is specified.

**ACTIVE**

Specifies that the subsystem is an RSR active subsystem.

**TRACKING**

Specifies that the subsystem is an RSR tracking subsystem.

**PARTSEL(pgmname) | HIKEY**

Mutually exclusive, optional parameters used to either identify or delete a user Partition Selection Exit program name for a HALDB. Deleting the partition selection exit program name causes all of the partition selection strings to be cleared. Adding or changing a partition selection exit program name will leave the partition high keys or partition selection strings intact. In all cases, all of the partitions are marked PINIT (partition initialization needed). The command will fail if NOPINIT is also specified. Use the CHANGE.PART command to specify new partition high keys or partition selection strings.

**PARTSEL(pgmname)**

Identifies a user Partition Selection Exit program name for a HALDB. The value is a program name that is up to 8 characters long.
Restriction: This parameter is not valid with ALL.

**HIKEY**
Removes an existing Partition Selection Exit program associated with a HALDB.

Restriction: This parameter is not valid with ALL.

**REORGI | NOREORGI**
Mutually exclusive, optional parameters used by online reorganization products (other than the HALDB Online Reorganization function that is delivered with IMS) to inform DBRC they are active (they intend to reorganize the database) to provide data integrity for full function databases. When the REORG INTENT flag is set to ON, database authorization fails for other reorganization products, batch access, and online image copy processes.

**REORGI**
Indicates that the database is going to be reorganized and sets the REORG INTENT flag to ON in the database record of the RECON data set if the following conditions have been met:
- Database type is valid. Must be full function including HALDBs.
- The REORG INTENT flag is off.
- No batch jobs are authorized to the database.
- HALDB Online Reorganization (OLR) cannot have cursor active set (OLREORG CURSOR ACTIVE=YES).
- The database cannot be owned by an IMS OLRIMSID=imsid.

**NOREORGI**
Indicates that the database is not going to be reorganized (or has just completed reorganization) and sets the REORG INTENT flag to OFF in the database record of the RECON data set if the following conditions have been met:
- Database type is valid. Must be full function including HALDBs.
- The REORG INTENT flag is on.

**Example of Changing a Record for a DB Identified with the DBD Parm**

This example specifies changes to be made to a record in RECON for the database identified with the DBD parameter. The level of data sharing is specified, and the database needs one backout.

```
//CHGDB JOB

SYSIN DD *
CHGDB DBD(THISDBD) NOAUTH READOFF SHARELVL(2) -
       BACKOUT(1) SSID(IMSID1)
```

**CHANGE.DBDS**

Use a CHANGE.DBDS command to change the information about a DBDS. This information is contained in a DBDS record in RECON. If you specify DSN, DDNNEW, or AREANEW while the database or an area of a DEDB is in use, the command fails.

Some attributes of a HALDB DBDS can be changed with this command. See "CHANGE.PART" on page 161 if you want to change all the DBDSs of a HALDB partition.
Restrictions for HALDBs: For HALDBs, you can use this command only as indicated in Table 10 on page 150.

Table 10. CHANGE.DBDS for HALDBs

<table>
<thead>
<tr>
<th>CHANGE.DBDS</th>
<th>Partition Data DBDS</th>
<th>Partition Index/ILDS DBDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDEQE</td>
<td>DELEQE</td>
<td>Yes</td>
</tr>
<tr>
<td>CFSTR1</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>CFSTR2</td>
<td>NOCFSTR2</td>
<td>N/A</td>
</tr>
<tr>
<td>AUTH</td>
<td>NOAUTH</td>
<td>N/A</td>
</tr>
<tr>
<td>DDNEW</td>
<td>AREANEW</td>
<td>No</td>
</tr>
<tr>
<td>DEFLTJCL</td>
<td>NODEFLT</td>
<td>Yes</td>
</tr>
<tr>
<td>DSN</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>GENMAX</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>GSGNAME</td>
<td>NOTCOVER</td>
<td>N/A</td>
</tr>
<tr>
<td>ICJCL</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>ICON</td>
<td>ICOFF</td>
<td>Yes</td>
</tr>
<tr>
<td>NOREUSE</td>
<td>REUSE</td>
<td>Yes</td>
</tr>
<tr>
<td>OICJCL</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>LKASID</td>
<td>NOLKASID</td>
<td>N/A</td>
</tr>
<tr>
<td>PRELOAD</td>
<td>NOPREL</td>
<td>N/A</td>
</tr>
<tr>
<td>PREOPEN</td>
<td>NOPREO</td>
<td>N/A</td>
</tr>
<tr>
<td>RCVTRACK</td>
<td>DBTRACK</td>
<td>N/A</td>
</tr>
<tr>
<td>RECOV</td>
<td>NORECOV</td>
<td>Yes</td>
</tr>
<tr>
<td>RECOVJCL</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>RECOVPD</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>RECVJCL</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>VSO</td>
<td>NOVSO</td>
<td>N/A</td>
</tr>
</tbody>
</table>

CHANGE.DBDS Syntax

```
CHANGE.DBDS DBD(name) DDN(name) AREA(name)
  ADDEQE(value)
  DELEQE(value)

CFSTR1(name) CFSTR2(name) AUTH NOAUTH DDNEW(name) AREANEW(name)

DEFLTJCL(member) NODEFLT DSN(name) GENMAX(value) GSGNAME(gsgname) NOTCOVER
```
Parameters for CHANGE.DBDS

**DBD(name)**
Required parameter you use to identify by its database name the DBDS or DEDB area whose record is to be changed.

**Restrictions for HALDBs**: For HALDBs, *name* specifies the name of a HALDB partition.

**DDN(name)** | **AREA(name)**
Mutually exclusive, required parameters you use to identify the DBDS or DEDB area whose record is to be changed. When you specify DDN, you specify the ddbname of the DBDS. When you specify AREA, you specify the name of the area.

**ADDEQE(value)** | **DELEQE(value)**
Mutually exclusive, optional parameters you use to change the error queue elements of a DBDS.

**ADDEQE**
Adds error queue elements to a DBDS. Error queue elements indicate that an I/O error occurred on the data set and that the data set therefore needs to be recovered. Processing continues except for that part of the data set described by the error queue element. Error queue elements can be added only when the DBDS is not in use. The value specified in the variable field is one or more 4-byte hexadecimal values enclosed in quotes; for example, ADDEQE(‘X’00002345’, ‘X’00012345’, …).

**DELEQE**
Deletes error queue elements from a DBDS. Deletion of error queue elements indicates that recovery processing has occurred on that data set. Error queue element deletions can be done only when the DBDS is not in use. The value specified in the variable field is one or more 4-byte hexadecimal values enclosed in quotes; for example, DELEQE(‘X’00002345’, ‘X’00012345’, …).

When you add an EQE to a database, the RECOV-NEEDED flag is automatically turned on. When you delete an EQE, and no other EQE flags exist for the database, the RECOV needed flag is turned off. Do not specify RECOV or NORECOV when specifying the ADDEQE or DELEQE parameters.

**CFSTR1(name)**
Optional parameter you specify to change the name of the first coupling facility structure for the identified VSO area. Adhere to the z/OS coupling facility structure naming conventions. This parameter is valid only for VSO areas of DEDBs that are defined with SHARELVL(2 | 3). The area name is the
default if the area is changed to VSO and the DEDB is SHARELVL(2 | 3). CFSTR1 is not allowed if the area is authorized, unless it is also being changed from NOVSO to VSO.

**CFSTR2(name) | NOCFSTR2**

Mutually exclusive, optional parameters you use to update or remove the name of the second coupling facility structure for the identified VSO area. These parameters are valid only for VSO areas of DEDBs that are defined with SHARELVL(2 | 3). These parameters are allowed for an authorized area only if it is being changed from NOVSO to VSO.

**CFSTR2(name)**

Specifies the new name of the second coupling facility structure. Adhere to the z/OS naming conventions for coupling facility structures.

**Related Reading:** See [IMS V9: Administration Guide: Database Manager](#) for details on coupling facility structure naming conventions.

**Restriction:** CFSTR2 cannot be specified with MAS or cannot be specified if the area is already specified as a multi-area coupling facility structure.

**NOCFSTR2**

Removes the name of the second coupling facility structure (CFSTR2).

**AUTH | NOAUTH**

Mutually exclusive, optional parameters you use to specify whether the area is available for authorization processing. The AUTH and NOAUTH parameters are valid only if you have specified the AREA parameter.

**AUTH**

Specifies that the area is available for authorization processing.

**NOAUTH**

Specifies that authorization processing is prohibited for the area.

If you specify the `CHANGE.DBDS AREA(name) RECOV` command, all ADSs that belong to that area are set to unavailable status at the same time.

**DDNNEW(name) | AREANEW(name)**

Mutually exclusive, optional parameters you use to change either the database ddname of the specified DBDS or the area name of the specified Fast Path DEDB area in RECON.

When you specify this parameter, the new ddname replaces the existing ddname for all records in RECON that correspond to the specified DBDS.

You must supply a ddname for the IMS DBDLIB data set in the JCL for the `CHANGE.DBDS` command. The new ddname must be defined in the IMS DBD library and its numeric data set identifier must be unchanged; it also cannot already exist in RECON.

AREANEW is valid only if you have specified the AREA parameter.

**DEFLTJCL(member) | NODEFLT**

Mutually exclusive, optional parameters you use to specify an implicit skeletal JCL default member for the DBDS.

**DEFLTJCL**

Specifies the new implicit skeletal JCL default member for the DBDS. The specified member is used by the GENJCL.1C, GENJCL.01C, and GENJCL.RECOV commands to resolve keywords you have defined.
NODEFLT
   Removes the default JCL name from the DBDS record when you do are not
   using a default JCL member.

DSN(name)
   Optional parameter you use to change the data set name of the identified
   DBDS. You cannot use this parameter for a DEDB area.

GENMAX(value)
   Optional parameter you use to change the maximum number of image copy
   data sets DBRC is to maintain for the specified DBDS or DEDB area. value
   must be a decimal number from 2 to 255.

   If the value you specify is smaller than the number of image copy data sets
   currently existing for the specified DBDS, records of image copy data sets that
   are beyond the recovery period are deleted from RECON until the number
   reaches the specified GENMAX value. Records of image copy data sets with
   the oldest time stamps are deleted until the number that remains equals the
   specified GENMAX value.

   If you are increasing the GENMAX value and REUSE is specified, use the
   INIT.IC command to create additional image copy records in RECON. If the
   number of data sets does not equal GENMAX, reuse of the data sets does not
   take place and you eventually run out of available data sets for the utility.

GSGNAME(gsgname) | NOTCOVER
   Mutually exclusive, optional parameters you use to assign the remote site
   recovery attributes of a DEDB area.

   GSGNAME
      Assigns the area to a global service group (GSG).

   NOTCOVER
      Discontinues remote site recovery for the area.

   GSGNAME and NOTCOVER are only valid if AREA is specified.

   You cannot use CHANGE.DBDS to change the state of an area from
   non-RSR-covered to RSR-covered. Message DSP1044I is issued if you attempt
   to change the covered state of the area with this command. In order to change
   an area from non-RSR-covered to RSR-covered, issue the following two
   commands for the area:
      • DELETE.DBDS
      • INIT.DBDS

ICJCL(member)
   Optional parameter you use to change the name of the member of the
   partitioned data set of skeletal JCL. The $GENJCL.1C command uses this name to
   generate the JCL for a run of the Database Image Copy utility for the specified
   DBDS or DEDB area.

ICON | ICOFF
   Mutually exclusive, optional parameters you use to specify whether a database
   needs an image copy.

   ICON
      Specifies that a DBDS needs to have an image copy taken. If the image
      copy status of a DBDS is IC RECOMMENDED, issuing a CHANGE.DBDS ICON
      command changes the image copy status (in the RECON) from IC
CHANGE.DBDS

RECOMMENDED to IC NEEDED. In the associated database record, the IC NEEDED counter increases to indicate how many DBDSs need an image copy.

ICOFF
Specifies that a DBDS does not need an image copy. If the image copy status of a DBDS is IC RECOMMENDED or IC NEEDED, issuing a CHANGE.DBDS ICOFF command resets (turns off) the image copy status. If the image copy status was IC NEEDED before issuing the CHANGE.DBDS ICOFF command, the IC NEEDED counter in the associated database record is decreased by one after the CHANGE.DBDS ICOFF command is issued. The IC NEEDED counter indicates the number of DBDSs that need an image copy.

NOREUSE | REUSE
Mutually exclusive, optional parameters you use to indicate whether image copy data sets can be reused for subsequent image copy jobs.

NOREUSE
Indicates that image copy data sets already used for the specified DBDS or DEDB area are not to be reused for subsequent image copies. Any existing, unused image copy data set records for the specified DBDS or DEDB area are deleted.

REUSE
Indicates that image copy data sets already used for the specified DBDS or DEDB area can be made available for reuse by subsequent image copies. You cannot specify REUSE if RECON contains any nonstandard image copy data set records for the DBDS or DEDB area.

If GENMAX is higher than the number of existing data sets for the group, use the INIT.IC command to add additional data sets; otherwise, reuse of the data sets does not take place. See “INIT.IC” on page 274 for more information.

For additional information about reusing image copy data sets, see “INIT.DBDS” on page 266 for an explanation of the REUSE parameter.

OICJCL(member)
Optional parameter you use to change the name of the partitioned data set member of skeletal JCL. You cannot use this parameter for a DEDB area. The GENJCL.OIC command uses this name to generate the JCL for a run of the Online Database Image Copy utility for the specified DBDS.

LKASID | NOLKASID
Mutually exclusive optional parameters you use to specify whether local data caching for the specified area is to be used for buffer lookaside on read requests. The LKASID option is valid only for VSO areas that are specified as SHARELVL(2 | 3). These parameters are allowed for an authorized area only if it is being changed from NOVSO to VSO.

LKASID
Indicates that buffer lookaside is to be performed on read requests for this area.

NOLKASID
Indicates that buffer lookaside is not to be performed on read requests for this area.

MAS | NOMAS
Mutually exclusive, optional parameters you use to specify whether a shared VSO area is to reside in a multi-area coupling facility structure.
MAS  Indicates that the area is to reside in a multi-area coupling facility structure. A second coupling facility structure name (CFSTR2) cannot be specified if MAS is specified. If a second structure is already defined, the second structure’s name is removed.

Restrictions: MAS can only be specified if the DEDB is specified as SHARELVL (2 or 3) and either VSO is specified or the area is already specified as a VSO DEDB. Also, CFSTR2 cannot be specified with MAS or cannot be specified if the area is already specified as a multi-area coupling facility structure.

NOMAS  Indicates that the coupling facility structure (or structures) should contain data for only this area.

PRELOAD | NOPREL
Mutually exclusive, optional parameters you use to specify whether a VSO DEDB area is to be loaded the next time it is opened.

PRELOAD
Indicates that the area is to be loaded into the data space or coupling facility structure the next time that it is opened. Selecting this option also causes the area to be preopened.

NOPREL
Indicates that the VSO area is not to be loaded into the data space or coupling facility structure the next time that it is opened. CIs are copied into a data space when they are read for the first time.

PREOPEN | NOPREO
Mutually exclusive, optional parameters you use to specify whether a VSO DEDB area is to be opened either, after the first checkpoint following the next control region initialization, or when the next /STA AREA command is processed.

PREOPEN
Indicates that the area is to be opened the next time the control region is started or a /STA AREA command is processed. This option is valid for both VSO and non-VSO areas.

NOPREO
Indicates that the area is not to be preopened the next time the control region is started or a /STA AREA command is processed. You cannot specify this parameter with the PRELOAD parameter.

RCVTRACK | DBTRACK
Mutually exclusive, optional parameters you use to specify the type of RSR tracking (shadowing) for an area assigned to a GSG.

RCVTRACK
Indicates recovery-readiness tracking.

DBTRACK
Indicates database-readiness tracking.

Restriction:
RCVTRACK and DBTRACK can only be specified if AREA is specified and the area is assigned to a GSG.
RECOV | NORECOV
Mutually exclusive, optional parameters you use to specify whether a DBDS or DEDB area needs to be recovered.

RECOV
Specifies that the DBDS or area needs to be recovered. A RECOVER-NEEDED counter in the associated database record is increased to indicate the number of DBDSs that need to be recovered.

NORECOV
Specifies that the DBDS or DEDB area does not need to be recovered. A RECOVERY-NEEDED counter in the associated database record is decreased to indicate the number of DBDSs that have been recovered.

RECOVJCL(member)
Optional parameter you use to change the name of a member of the partitioned data set of skeletal JCL. The GENJCL.RECOV command uses the member to generate the JCL for a run of DBRC for the specified DBDS or DEDB area.

RECOVPD(value)
Optional parameter you use to change the recovery period for a specified DBDS or DEDB area.

value must be a number from 0 to 999 that represents the number of days the image copies are to be kept in RECON. A 0 indicates no recovery period.

RECVJCL(member)
Optional parameter you use to specify the name of the skeletal JCL member to be used for the GENJCL.RECEIVE command.

RECVJCL can be specified for both RSR-covered and non-RSR-covered DL/I DBDSs and Fast Path areas.

VSO | NOVSO
Mutually exclusive, optional parameters you use to specify whether an area resides in virtual storage the next time the control region is initialized or when the next /STA AREA command is processed.

VSO
Indicates that the area is to reside in virtual storage. Areas defined with SHARELVL(0 | 1) are read into and written from a z/OS data space. Areas defined with SHARELVL(2 | 3) use the coupling facility to share data between connected subsystems.

NOVSO
Indicates that this area is not to reside in virtual storage.

If an area was previously specified as SHARELVL(2 | 3), changing the area to NOVSO clears the coupling facility structure names and resets the LKASID setting to NOLKASID. NOVSO cannot be specified if the area is in use.

Example of Changing a Record for a Fast Path DEDB
This example specifies changes to be made to the record in RECON for the Fast Path DEDB that is identified by the DBD and AREA parameters. The image copy data sets for the specified DEDB area are not reused, and the maximum number of images that DBRC maintains is two. In addition, the image copy data sets for the specified DBDS are kept in RECON for at least 15 days.
CHANGE.DBDS

//CHGDBDS JOB
:
://SYSIN DD *
   CHANGE.DBDS DBD(DB3) AREA(DD3) NOREUSE -
       GENMAX(2) RECOVPD(15)
/*

CHANGE.DBDSGRP

Use a CHANGE.DBDSGRP command to change the information about a DBDS or
database group. This information is contained in a DBDS group record in RECON.

CHANGE.DBDSGRP Syntax

```
GRPCODE
```

Notes:
1 Full-function database name
2 Fast Path DEDB name

Parameters for CHANGE.DBDSGRP

GRPCODE

Required parameter you use to identify the DBDSGRP to be changed. A record
with that name must already exist.

ADDDB(dbname | areaname) | DELDB(dbname | areaname) |
ADDMEM(dbname,ddname | areaname) | DELMEM(dbname,ddname | areaname) |

Chapter 9. CHANGE Commands  157
CHANGE.DBDSGRP

ADDRECOV(dbname,areaname) | DELRECOV(dbname,areaname)
Mutually exclusive, optional parameters you use to identify the member or members to be added to or deleted from the group. A member can belong to any number of DB groups or DBDS groups but can belong only to one recovery group.

ADDBD(dbname)
Identifies one or more database or area names to be added to a DB group. The name specified can be a full-function database name, a Fast Path DEDB name, or an area name.

DELBDB(dbname)
Identifies one or more database or area names to be deleted from a DB group. The name specified can be a full-function database name, a Fast Path DEDB name, or an area name.

ADDMEM(dbname,ddname)
Identifies one or more members to be added to a DBDS group, where dbname is the database name and ddname is the DD statement name or the area name.

DELMEM(dbname,ddname)
Identifies one or more members to be deleted from a DBDS group, where dbname is the database name and ddname is the DD statement name or the area name.

ADDRECOV(dbname,areaname)
Identifies one or more databases or DEDB areas to be added to a recovery group, where dbname is the database (or DEDB) name and areaname is the DD statement name or the area name.

If a DEDB area is to be added to the recovery group, both dbname and areaname must be specified. If the group specified is not a recovery group, the command fails with message DSP0077I.

A database or area can belong to only one recovery group. If any of the members specified by ADDRECOV already belong to another recovery group, the command fails with message DSP0078I.

DELRECOV(dbname,areaname)
Identifies one or more databases or DEDB areas to be deleted from a recovery group, where dbname is the database (or DEDB) name and areaname is the DD statement name or the area name.

If a DEDB area is to be deleted from the recovery group, both dbname and areaname must be specified. If the group specified is not a recovery group, the command fails with message DSP0077I.

If you delete all the members of a group, the record of that group is deleted from RECON.

Restrictions for HALDBs: For HALDBs, dbname specifies the name of a HALDB partition or the HALDB master. When specifying HALDB database names, use the CHANGE.DBDSGRP command only as defined in Table 1.1.

Table 11. Supported dbname specifications for HALDBs on CHANGE.DBDSGRP command parameters

<table>
<thead>
<tr>
<th>CHANGE.DBDSGRP</th>
<th>dbname = HALDB Master</th>
<th>dbname = Partition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDDB</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 11. Supported dbname specifications for HALDBs on CHANGE.DBDSGRP command parameters (continued)

<table>
<thead>
<tr>
<th>CHANGE.DBDSGRP</th>
<th>dbname = HALDB Master</th>
<th>dbname = Partition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELDB</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ADDMEM</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DELMEM</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>ADDRECOV</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DELRECOV</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Example of Changing a Group of DBDSs

In this example, a group of DBDSs is changed.

```plaintext
//CHGDBGRP JOB :

//SYSIN DD *
CHANGE.DBDSGRP GRPNAME(GRP1) ADDMEM((DB1,DD1),(DB2,DD2)) /*
```

CHANGE.IC

Use a CHANGE.IC command to modify information contained in an image copy record in RECON.

**CHANGE.IC Syntax**

```
CHANGE.IC DBD(name) DDN(name) AREA(name) RECTIME(time_stamp)
FILESEQ(value) FILESEQ2(value) ICDSN(name) ICDSN2(name)
INVALID VALID INVALID2 VALID2 RECDCT(value) UNIT(3400 unittype)
UNIT2(3400 unittype) VOLLIST(volser)
VOLLIST2(volser) STOPTIME(time_stamp)
```

**Parameters for CHANGE.IC**

**DBD(name)**

Required parameter you use to identify the database name of the DBDS whose image copy record is to be modified.

**Restrictions for HALDBs**: For HALDBs, `dbname` specifies the name of a HALDB partition.
**CHANGE.IC**

**DDN**(name)  |  **AREA**(name)

Mutually exclusive, required parameters you use to identify the name of the DBDS or DEDB area to which the image copy record being modified is related.

**RECTYPE**(time_stamp)

Required parameter you use to identify the specific image copy data set record to be changed.

Use the Run time marked with an asterisk (*) from the listing of the IMAGE record. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

**FILESEQ**(value)

Optional parameter you use to change the file sequence number in the record of the identified image copy data set.

**FILESEQ2**(value)

Optional parameter you use to change or add the file-sequence number in the record of the identified duplicate image copy data set.

**ICDSN**(name)

Optional parameter you use to change the data set name of the identified image copy data set.

**ICDSN2**(name)

Optional parameter you use to change or add the data set name of the identified duplicate image copy data set in an image copy record.

To change the name of the duplicate image copy data set, a record of the first image copy data set must exist in RECON.

**INVALID**  |  **VALID**

Mutually exclusive, optional parameters that you use to prevent or permit the use of an image copy data set as input to a subsequent run of the Database Recovery utility (DFSURDB0).

**INVALID**

Prevents the use of the specified image copy data set as input to a subsequent run of the Database Recovery utility. If the invalidated image copy data set is reused, it is automatically marked as valid. See “INIT.DBDS” on page 266 for an explanation of the REUSE parameter.

**VALID**

Permits the use of a previously invalidated image copy data set as input to a subsequent run of the Database Recovery utility.

The Database Recovery utility expects this image copy data set as input unless it is marked as INVALID, in which case the utility expects a duplicate image copy data set as input.

**INVALID2**  |  **VALID2**

Mutually exclusive, optional parameters that you use to prevent or permit the use of a duplicate image copy data set as input to a subsequent run of the Database Recovery utility (DFSURDB0).

**INVALID2**

Prevents the use of the specified, duplicate image copy data set as input to a subsequent run of the Database Recovery utility. If the invalidated, duplicate image copy data set is subsequently reused, it is automatically marked as valid. See “INIT.DBDS” on page 266 for an explanation of the REUSE parameter.
VALID2
Permits the use of a previously invalidated, duplicate image copy data set as input to a subsequent run of the Database Recovery utility.

If both INVALID2 and VALID2 are specified, the last one specified is used.

RECDCT(value)
Optional parameter you use to change the count of the records in the corresponding image copy data set in the specified image copy record. CHANGE.IC.value must be a decimal number up to 2,147,483,647.

UNIT(3400 | unittype)
Optional parameter you use to change the unit type that is recorded in the specified image copy record. The unit type can be up to eight alphanumeric characters.

UNIT2(3400 | unittype)
Optional parameter you use to change the unit type that is recorded in the specified duplicate image copy record. The unit type can be up to eight alphanumeric characters.

VOLLIST(volser)
Optional parameter you use to change, in the image copy record, the volume serial numbers of the volumes on which the identified image copy data set resides.

VOLLIST2(volser)
Optional parameter you use to change or add, in the image copy record, the volume serial numbers of the volumes on which the identified duplicate image copy data set resides.

STOPTIME(time_stamp)
Optional parameter you use to specify the time when an image copy has completed. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124) and cannot be less than the image copy start time. If this is an HSSP CIC that is in progress, specifying a valid stop time terminates the HSSP CIC and resets the in-progress indicators in the IC record and the DBDS record.

Example of Changing an Image Copy Record
In this example, information in an image copy record that is identified by the DBD, DDN, and RECTIME parameters is to be changed in RECON. The new data set names of both image copy data sets (specified in the ICDSN and ICDSN2 parameters) follow the default naming convention. The volume serial numbers on which the image copy data sets reside are also to be changed as specified in the VOLLIST and VOLLIST2 parameters.

```
//CHGIC   JOB
:
  //SYSIN   DD  *
  CHANGE.IC DBD(DBDKS051) DDN(DONKDS051) -
          ICDSN(IMS.DBDKDS051.DONKDS051.IC.ICDSN02) -
          ICDSN2(IMS.DBDKDS051.DONKDS051.IC2.ICDSN02) -
          VOLLIST(ICVOL1,ICVOL2,ICVOL3) FILESEQ2(2) -
          VOLLIST2(ICVOL4) RECTIME(820921314143)
/*
```

CHANGE.PART
Use a CHANGE.PART command to change attributes of a HALDB partition. The changes apply to all the DBDSs of the partition. Some attributes such as skeletal
JCL member names can be changed for a single DBDS of a partition by a CHANGE.DBDS command. See “CHANGE.DBDS” on page 149 for a list of what can be changed.

Restriction: The IMS DBDLIB data set must be identified in the job stream for the RECOVERY Control utility with a ddname of IMS if KEYSTRING is being set.

Use the DISABLE parameter to temporarily remove a partition from active use. Use the ENABLE parameter to return the partition to active use. You would normally DISABLE a partition prior to deleting it (see “DELETE.PART” on page 211), but do not want to remove all the recovery related information that DBRC maintains until the changes have been tested. A partition marked as DISABLE can be deleted or made active again by marking it ENABLE. A partition that is changed from DISABLE to ENABLE must be recovered before it can be used.

Restriction: No other parameters are allowed on a CHANGE.PART command if either DISABLE or ENABLE is specified.

Related Reading: See the [IMS V9: Administration Guide: Database Manager] for information on designing databases, especially the following:
- “Designing Databases”
- “Naming Conventions”
- “Procedure for Changing PHDAM, PHIDAM, and PSINDEX Partition Definitions”

CHANGE.PART Syntax

Use one of the following diagrams for the CHANGE.PART command.

Or:

CHANGE.PART—DBD(name)—PART(name)
Parameters for CHANGE.PART

**DBD**(name)
Required parameter used to identify the HALDB.

**PART**(name)
Required parameter you use to identify the partition.

**KEYSTRING**(char or hex value)
Optional parameter used to change a HALDB partition high key value or a selection string for use by a partition selection exit. Specified as a character value up to 256 characters long or a hexadecimal value up to 512 characters long. Character values must be alphanumerics (with no embedded blanks or commas unless the string is enclosed by single quotes). Unless enclosed by single quotes, the character string will be folded to upper case. Hexadecimal values must be enclosed by single quotes and preceded by the letter X, for example: KEYSTRING('X'D7C1D9E3D2C5E8').

If no partition selection routine is specified for the HALDB, KEYSTRING defines a partition high key value which cannot be longer than the root key length. If the high key length is less than the defined root key length, the high key value is padded with hex 'FF's up to the defined root key length. The partition high key values must be unique for each partition within a HALDB.

If a partition selection routine is specified for the HALDB, KEYSTRING defines a Partition Selection String which is passed to the partition selection routine. The Partition Selection String may be up to 256 bytes long and consist of simple character information. If it contains non-printable characters, it must be identified using hex notation. A hex character string is enclosed by single quotation marks and prefixed with an X.

The command fails if the partition is in use, or any other partition affected by a change to the high key value is in use. The partition and all other partitions affected by a change to the high key value are marked as needing partition initialization (PINIT is set on) and can not be used until initialized (see HALDB Partition Data Set Initialization utility (DFSUPNT0) in [IMS V9: Utilities](#)).

**Requirement:** When you issue a CHANGE.PART KEYSTRING command, the IMS DDLIB data set must be identified in the job stream for the Database Recovery Control utility with a ddname of IMS.

**DSNPREFIX**(string)
Optional parameter used to change the data set name prefix for the partition data sets contained in a HALDB. Specified as a value, up to 37 characters long, that is a valid JCL data set name.

**RANDOMZR**(name)
Optional parameter used to change the name of the randomizing module for HALDB PHDAM databases only. A randomizing module controls root segment placement in, or retrieval from, the PHDAM HALDB.

**ANCHOR**(value)
Optional parameter used to change the number of RAPs (root anchor points) desired in each control interval or block in the root addressable area of a PHDAM HALDB. The value specified must be between 1 and 255. Typical values are from 1 to 5.

**HIBLOCK**(value)
Optional parameter used to change the maximum relative block number value that the user wishes to allow a randomizing module to produce for this

Reference: Database and Transaction Manager, SC18-7833

Chapter 9. CHANGE Commands 163
HALDB. This value determines the number of control intervals or blocks in the
root addressable area of an PHDAM HALDB. The value may range between 1
and 16,777,215 (2**24 - 1). This parameter is valid only for a PHDAM HALDB.

BYTES(value)
Optional parameter used to change the maximum number of bytes of a
HALDB record that can be stored into the root addressable area in a series of
inserts unbroken by a call to another HALDB record. The value has a range of
0 to 16,777,215 (2**24 - 1). This parameter is valid for PHDAM HALDBs only.
A value of 0 (zero) means that all bytes are addressable. Setting BYTES to 0 is
equivalent to omitting BYTES on the DBD definition

FBFF(value)
Optional parameter used to change the free block frequency factor (fbff) which
specifies that every nth control interval or block in this data set group is left as
free space during database load or reorganization (where FBFF=n). The range
of FBFF includes all integer values from 0 to 100 except 1.

FSPF(value)
Optional parameter used to change the free space percentage factor. It specifies
the minimum percentage of each control interval or block that is to be left as
free space in this data set group. value may be any number between 0 and 99.

BLOCKSZEmmmm
Optional parameter used to change the block size for OSAM data sets. Specify
an even numeric value no greater than 32766. The block size value is used by
OSAM only. You may specify up to 10 values, one for each data set group
defined in the DBD. Omitted values remain unchanged.

GENMAX(value)
Optional parameter used to change the maximum number of image copies that
DBRC is to maintain for the partition DBDSs. If you identify a partition DBDS
with the NOREUSE parameter, the oldest image copy beyond the recovery
period is deleted when the number of image copies exceeds the GENMAX
value. If you identify it with the REUSE parameter, the oldest image copy
beyond the recovery period is reused. Specified as a numeric value from 2 to
255. All partition DBDSs will be changed to use this GENMAX value. The
CHANGE.DBDS command can be used to change this for individual partition
DBDSs.

DEFLTJCL(member)
Optional parameter used to change the implicit skeletal JCL default member
for a HALDB Partition DBDS. The specified member is used by the
GENJCL.IC, GENJCL.OIC, and GENJCL.RECOV commands in order to resolve
keywords you have defined. All partition DBDSs will be changed to use this
DEFLTJCL member. The CHANGE.DBDS command can be used to change this
for individual partition DBDSs.

ICJCL(member)
Optional parameter used to change the name of a member of a partitioned
data set that contains skeletal JCL. When you issue a GENJCL.IC command,
DBRC uses this member to generate the JCL to run the Database Image Copy
utility (or the Database Image Copy 2 utility) for the partition DBDS specified
on the GENJCL command. All partition DBDSs will be changed to use this
ICJCL member. The CHANGE.DBDS command can be used to change this for
individual partition DBDSs.

NOREUSE | REUSE
 Mutually exclusive, optional parameters used to change whether the supported
image copy utilities are to reuse previously used image copy data sets. REUSE allows the GENJCL.IC command or the GENJCL.OIC command to generate a job that causes the supported image copy utilities to reuse the oldest image copy data set (for the DBDS specified on the GENJCL command) when the GENMAX value for it is exceeded. REUSE requires that you create empty image copy data sets for future use by the supported image copy utilities. In addition, you must use an INIT.IC command to record their existence in RECON. The NOREUSE parameter prohibits such actions. All partition DBDSs will be changed to use the parameter specified. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

OICJCL (member)
Optional parameter used to change the name of a member of a partitioned data set that contains skeletal JCL. When you issue a GENJCL.OIC command, DBRC uses this member to generate the JCL to run the Online Image Copy utility for the partition DBDS specified on the GENJCL command. Appropriate partition DBDSs will be changed to use this OICJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

RECOVJCL(member)
Optional parameter used to change the name of a member of a partitioned data set that contains skeletal JCL. When you issue the GENJCL.RECOV command, DBRC uses this member to generate the JCL to run the Database Recovery utility for the partition DBDS specified on the GENJCL command. All partition DBDSs will be changed to use this RECOVJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

RECOVPD(value)
Optional parameter used to change the recovery period for a specified partition DBDS. Specify a numeric value from 0 to 999 that represents the number of days you want DBRC to maintain sufficient recovery-generation information in the RECON data set. All partition DBDSs will be changed to use this RECOVPD value. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

RECVJCL(member)
Optional parameter used to change the name of the skeletal JCL member to be used by the GENJCL.RECEIVE command. RECVJCL can be specified for both RSR-covered and non-covered HALDB DBDSs. All partition DBDSs will be changed to use this RECVJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

DISABLE | ENABLE
Mutually exclusive parameters used to change the state of the partition.
DISABLE marks the partition as unavailable for use by IMS and most utilities.
In most situations, a partition marked DISABLE will not be known by IMS (it is treated as if it is not registered to DBRC). DBRC LIST commands will list information about partitions marked DISABLE and the HALDB Partition Definition utility (PDU) displays partitions which are marked DISABLE. The partition still counts towards the maximum number of partitions which may be defined for a HALDB.

Although a partition that is marked DISABLE is not removed from DBRC groups (a CAGROUP, DBDSGRP, DBGROUP and RECOVGRP), it is generally not processed as part of the group when the group is used in a DBRC command. This also applies to implied groups used in a command. The one exception to this is the GENJCL.CA command and the execution of the IMS
Database Change Accumulation utility. Changes will be accumulated for partitions that are members of a CA Group, even when marked DISABLE. GENJCL.IC and GENJCL.RECOV commands fail for partitions that are marked DISABLE. If the commands specify a group of any kind, partitions that are marked DISABLE are skipped, that is, JCL is not generated and no message is issued.

ENABLE makes a partition which had been made DISABLE available for use by IMS again. The partition is set to a Recovery Needed status and must be recovered before being used by IMS. Setting a partition to ENABLE has the same effect as adding a new partition to the HALDB. If the HALDB uses a partition selection exit, all the other partitions are set to a Partition Initialization Needed status. If the HALDB uses high key values, the partition with the next high key value is set to a Partition Initialization Needed status.

**Requirement:** When you issue a CHANGE.PART ENABLE command, the IMS DBDLIB data set must be identified in the job stream for the Database Recovery Control utility with a ddname of IMS.

### Examples of Changing a Partition

This example specifies changes to be made to the partition that is identified by the DBD and NAME parameters. The high key for the partition is changed, as well as the name of the randomizer routine. The maximum number of images that DBRC maintains is 12. In addition, the image copy data sets for the specified DBDS are kept in RECON for at least 15 days. The partition and the next one will be marked as needing partition initialization as a result of the change.

```plaintext
//CHGPART JOB

//SYSIN DD * CHANGE.PART DBD(DB3) PART(PART3) KEYPSTRING(X'D7C1D9E3D2C5E8') - RANDOMZR(RNDNAME2) GENMAX(12) RECOVPD(15)
/*

The following example changes the partition identified by the DBD and PART parameters to a DISABLE status.

```
//CHGPART JOB

... SYSIN DD * CHANGE.PART DBD(DB3) PART(PART3) DISABLE
/*
```

### CHANGE.PRILOG (for OLDS)

Use a CHANGE.PRILOG (for OLDS) command to change information in the RECON about a PRIOLDS. You cannot change the archive status after an OLDS has been archived.

#### CHANGE.PRILOG (for OLDS) Syntax

```
CHANGE.PRILOG—OLDS(ddname) — ARNEEDED — ARSCHED — ARSTART — ARCHIVED — DSN(name)
AVAIL — UNAVAIL
```

**Examples:**

1. **CHANGE.PRILOG—OLDS(ddname)**

   - **ARNEEDED**:
     - Change Archive Needed
   - **ARSCHED**:
     - Archive Scheduled
   - **ARSTART**:
     - Archive Started
   - **ARCHIVED**:
     - Archive Completed
   - **DSN(name)**
     - Data Set Name
   - **AVAIL**:
     - Available
   - **UNAVAIL**:
     - Unavailable

2. **CHANGE.PRILOG—OLDS(ddname)**

   - Change Archive Needed
   - Archive Scheduled
   - Archive Started
   - Archive Completed
   - Data Set Name
   - Available
   - Unavailable

3. **CHANGE.PRILOG—OLDS(ddname)**

   - Change Archive Needed
   - Archive Scheduled
   - Archive Started
   - Archive Completed
   - Data Set Name
   - Available
   - Unavailable
Parameters for CHANGE.PRILOG (for OLDS)

OLDS(ddname)
Required parameter you use to specify the OLDS for which the RECON record is to be changed.

ARNEEDED | ARSCHED | ARSTART | ARCHIVED
Mutually exclusive, optional parameters you use to change the archive status of an OLDS.

ARNEEDED
Indicates that the OLDS was closed by IMS and needs to be archived.

ARSCHED
Indicates that the GENJCL.ARCHIVE command has been issued for the OLDS.

ARSTART
Indicates that the Log Archive utility is currently archiving the OLDS.

ARCHIVED
Indicates that the OLDS has been archived and is available for reuse.

AVAIL | UNAVAIL
Mutually exclusive, optional parameters you use to change the PRIOLDS to indicate its availability.

AVAIL
Indicates that the OLDS contains valid data and can be used as input to the Log Archive utility.

UNAVAIL
Indicates that the OLDS contains invalid data and should not be used as input to the Log Archive utility.

DSN(name)
Optional parameter you use to change the name of a primary OLDS. name can be up to 44 characters long.

ERROR | NORMAL
Mutually exclusive, optional parameters you use to change the specified PRIOLDS to indicate whether it contains errors.

ERROR
Changes the RECON record to indicate that a specified OLDS contains errors, so IMS is unable to close the OLDS properly. Close the OLDS before it is used as input to the Log Archive utility.

If you use dual logging, the subsystem uses the data in the error-free OLDS (in other words, the SECOLDS) to close the OLDS marked in error.

If you do not use dual logging, the subsystem uses the next-OLDS to close the OLDS that is marked in error.

NORMAL
Changes the record of the PRIOLDS, which was previously marked as containing errors, to indicate that the data set is now available for use as input to any log utility. When you specify NORMAL for an OLDS, the record immediately indicates that neither the secondary OLDS nor the next-OLDS is needed in order to close the specified OLDS.
CHANGE.PRILOG (for OLDS)

DBRC selects the required log data sets from the PRILOG (or SECLOG) records. These can contain RLDS entries, SLDS entries, or both. If you issue a CHANGE.PRILOG RLDS ERROR command, DBRC automatically uses the corresponding SECLOG entry, if one exists. If a SECLOG entry does not exist, or if it is marked in error, the GENJCL commands that require log data for this time frame fail.

SSID(name)
Optional parameter you use to specify the name of the IMS subsystem that created the OLDS for which the RECON record is to be changed. The SSID is an eight-character string consisting of any alphanumeric characters that represent a valid IMS subsystem identification name.

If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. Use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

Example of Renaming an OLDS
In this example, the IMS online subsystem, IMSA, that creates the PRIOLDS, renames an OLDS.

//CHGPRLG JOB
:
//SYSIN DD *
   /CHG PRILOG OLDS(DFSOLP02) -
   DSN(IMS.NEWLOG) SSID(IMSA)
/*

CHANGE.PRILOG (for RLDS)
You can use the CHANGE.PRILOG (for RLDS) command to change information in the RECON about a primary RLDS (or an SLDS that a batch subsystem created). Use the NOTIFY.PRILOG (for RLDS) command to add a PRILOG record or to add data set entries to an existing PRILOG record.

With the exception of the GSG name and the gap information, all the information you can change resides in a data set entry of the PRILOG record. Each CHANGE.PRILOG command you issue changes only one data set entry. If the log has multiple data sets, you must use the DSSTART parameter to identify the data set entry to be changed. (Note that if you are only changing the GSG or the gap information, you must still specify DSSTART if the log has more than one data set.)

If the PRILOG record represents log data which was received by an RSR tracking site from an active IMS subsystem, none of the keywords FILESEQ, NEWTIME, NEWVOL, OLDVOL, RUNTIMES, CHKPTID, UNIT, or VOLLIST can be specified. Log data sets received at a tracking site must be cataloged.

CHANGE.PRILOG (for RLDS) Syntax

```
CHANGE.PRILOG-RLDS-STARTIME(time_stamp)

<table>
<thead>
<tr>
<th>DSN(name)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CHKPTCT(value)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CHKPTID(chkptid)</th>
</tr>
</thead>
</table>
```
CHANGE.PRILOG (for RLDS)

Parameters for CHANGE.PRILOG (for RLDS)

RLDS
Optional parameter you use to specify that a PRILOG record is to be changed.

STARTIME(time_stamp)
Required parameter you use to specify the starting time stamp of the PRILOG record that is to be changed. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

DSN(name)
Optional parameter you use to change data set name. name can be up to 44 characters.

CHKPTCT(value)
Optional parameter you use to change the number of checkpoints completed on each volume of the data set. Specify a value for each volume designated in the OLDVOL or NEWVOL parameters. If OLDVOL is specified without NEWVOL, the number of values for CHKPTCT equals the number of volume serial numbers that appear with OLDVOL. If NEWVOL is specified, the number of values for CHKPTCT equals the number of volume serial numbers that appear in NEWVOL.

The values for CHKPTCT are:
0 No checkpoints on the volume
1 A single checkpoint on the volume
2 More than one checkpoint on the volume

CHKPTID(chkptid)
Optional parameter you use to change the oldest checkpoint ID for any active PST on each volume of the data set. Specify one checkpoint ID for each volume listed in OLDVOL or NEWVOL. If OLDVOL is specified without NEWVOL, the number of checkpoint IDs equals the number of volumes listed in OLDVOL. If NEWVOL is specified, the number of checkpoint IDs equals the number of volumes listed in NEWVOL.
CHANGE.PRILOG (for RLDS)

The checkpoint ID must be in standard form for a time stamp (see “Standard Time Stamp Format” on page 124). You can specify a zero time value.

**DSSTART(time_stamp)**

is a parameter you use to specify the starting time of the data set entry to be changed. The DSSTART parameter is required if the PRILOG has multiple data set entries; it is optional if the PRILOG has only one data set entry. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

**ERROR | NORMAL**

Mutually exclusive, optional parameters you use to change the data set entry to indicate whether it contains errors.

**ERROR**

changes the data set to indicate that it contains errors and should not be used as input to any DBRC-controlled run of a recovery utility.

**NORMAL**

changes a data set which was previously marked as containing errors to indicate that it is now available for use as input to any recovery utility.

**FILESEQ(value)**

Optional parameter you use to specify the file sequence number on the volume. Specify this parameter only if you specify a VOLLIST parameter. The value you substitute in the variable field must be a decimal number from 1 to 9999.

**GAP(ON | OFF)**

Optional parameter you use to set (ON) or reset (OFF) the GAP flag in a tracking PRILOG record.

**GSG(gsgname)**

Optional parameter you use to change the global service group (GSG) name in the PRILOG record.

**NEWTIME(time_stamp)**

Optional parameter you use to change the stop times of any but the last volume of the data set. If you specify NEWTIME, you must also specify OLDVOL and NEWVOL. The parameter sets NEWTIME OLDVOL NEWVOL and RUNTIMES VOLLIST are mutually exclusive.

If you specify NEWTIME, you must specify one less time stamp than the number of volume serial numbers specified in NEWVOL. This is because the stop time of the last volume specified in NEWVOL cannot be changed with this command. (See “NOTIFY.PRILOG (for RLDS)” on page 317 to see how to specify the stop time of the final volume.) Each time stamp is used as the volume stop time of the corresponding volume serial number specified by NEWVOL. If not specified, the stop time of the new volume is the same as the stop time of the last-specified old volume.

Each time stamp you specify must be greater than the previous time stamp. The first time stamp in NEWTIME must be greater than or equal to the stop time of the volume immediately preceding the changed volumes. Each time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

**NEWVOL(volser)**

Optional parameter you use to change the volume serial number of one or more volumes of the data set. If you specify NEWVOL, you must also specify OLDVOL.
The volume serial numbers that you specify in NEWVOL replaces the corresponding volume serial numbers specified in the OLDVOL parameter. You do not need to specify the same number of volume serial numbers in NEWVOL and OLDVOL. You cannot specify a volume serial number in NEWVOL that is the same as one that already exists in the PRILOG record.

You can specify from 1 to 255 volume serial numbers.

Use the NEWTIME parameter to change the time stamps as well as the serial numbers of the volumes.

**OLDVOL**(volser)
Optional parameter you use to change the volume serial number of one or more volumes of the data set. If you specify OLDVOL, you must also specify NEWVOL, CHKPTCT, or CHKPTID.

The volume serial numbers you specify are those of the volumes being changed. Each volume serial number specified in OLDVOL must match a volume serial number in the PRILOG record.

You can specify from 1 to 255 volume serial numbers.

**PREVGAP**(ON | OFF)
Optional parameter you use to set (ON) or reset (OFF) the PREV-GAP flag in a tracking PRILOG record.

**RUNTIMES**(time_stamp)
Optional parameter you use to change the stop times of any but the last volume of the data set. This parameter is provided for compatibility with previous releases of DBRC. Use NEWTIME OLDVOL NEWVOL to change the stop times of log volumes. If you do specify RUNTIMES, you must also specify VOLLIST. The following parameter sets are mutually exclusive:
- NEWTIME
- OLDVOL
- NEWVOL
- RUNTIMES
- VOLLIST

You can specify up to 255 time stamps on the RUNTIMES parameter. Each time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

Each time stamp in the variable field must correspond to a volume in the variable field of the VOLLIST parameter. The variable fields of the RUNTIMES and VOLLIST keywords must each contain the same number of entries. Each time stamp in the variable field of the RUNTIMES parameter must be greater than the previous time stamp.

The first time stamp in the variable field of the RUNTIMES parameter must be greater than the time stamp specified for the STARTIME parameter. The last time stamp in the variable field of the RUNTIMES parameter must be equal to the stop time of the corresponding primary RLDS as specified in the record being changed. You cannot use this command to change the stop time of the primary RLDS. For information about closing open recovery logs, see “NOTIFY.PRILOG (for RLDS)” on page 317.

**UNIT**(3400 | unittype)
Optional parameter you use to change the unit type of the device on which the data set resides. The unit type can be up to eight alphanumeric characters long.

**VOLLIST**(volser)
Optional parameter you use to change the record of the volume serial numbers
of the volumes that contain the data set. This parameter is provided for compatibility with previous releases of DBRC. Use the new parameter set, NEWTIME, OLDVOL, and NEWVOL to change the volume serial numbers of volumes in the data set.

If you specify the VOLLIST parameter, you must also specify the RUNTIMES parameter. See the description of the RUNTIMES parameter under "Parameters for CHANGE.PRILOG (for RLDS)" on page 169 for an explanation of how the two parameters interact. The following parameter sets are mutually exclusive:

- NEWTIME
- OLDVOL
- NEWVOL
- RUNTIMES
- VOLLIST

Examples of Using the CHANGE.PRILOG (for RLDS) Command

Here are some examples of using the CHANGE.PRILOG (for RLDS) command.

Example of Changing Volume Serial Numbers: In this example, some volume serial numbers are changed for a log which contains a single data set. The example PRILOG record in RECON has six volumes (VOL001, VOL002, VOL003, VOL004, VOL005, and VOL006) and a start time of 842331243299. The serial numbers of the third and fourth volumes can be changed with the following command:

```
//CHGPRILG JOB
:
//SYSIN DD *
  CHANGE.PRILOG RLDS STARTIME(842331243299) -
    OLDVOL(VOL003, VOL004) -
    NEWVOL(VOL007, VOL008, VOL009)
/*
```

Example of Marking Primary RLDS for Errors: In this example, one data set of a log is being marked as containing errors.

```
//CHGPRILG JOB
:
//SYSIN DD *
  CHANGE.PRILOG RLDS STARTIME(840541212120) -
    DSSTART(840541212120) ERROR
/*
```

CHANGE.PRILOG (for SLDS and TSLDS)

You can use the CHANGE.PRILOG (for SLDS) command to change information in the RECON about a primary SLDS for an online system. You can use the CHANGE.PRILOG (for TSLDS) command to change information in the RECON about a primary SLDS for an RSR tracking subsystem. Use CHANGE.PRILOG (for RLDS) to change information about an SLDS that a batch subsystem created, because DBRC considers such data to be an RLDS. Use the NOTIFY.PRILOG (for SLDS) command to add a PRISLD record or to add data set entries to an existing PRISLD record.

With the exception of the GSG name, all the information you can change resides in a data set entry of the PRISLD record. Each CHANGE.PRILOG command you issue changes only one data set entry. If the log has multiple data sets, you must use the DSSTART parameter to identify the data set entry to be changed. (Note that if you are only changing the GSG, you must still specify DSSTART if the log has more than one data set.)
If the PRISLD record represents log data that was received by an RSR tracking site from an active IMS subsystem, none of the keywords FILESEQ, NEWTIME, NEWVOL, OLDVOL, RUNTIMES, CHKPTID, UNIT, or VOLLIST can be specified. Log data sets received at a tracking site must be cataloged.

**CHANGE.PRILOG (for SLDS and TSLDS) Syntax**

```
CHANGE.PRILOG SLDS STARTIME(time_stamp) DSN(name)

CHKPTCT(value)

CHKPTID(chkptid)

DSSTART(time_stamp) ERROR NORMAL FILESEQ(value) GSG(gsgname)

NEWTIME(time_stamp)

NEWVOL(volser)

OLDVOL(volser)

RUNTIMES(time_stamp)

SSID(name)

UNIT(unittype)

VOLLIST(volser)
```

**Parameters for CHANGE.PRILOG (for SLDS and TSLDS)**

**SLDS**
- Required parameter you use to specify that an PRISLD record is to be changed.

**TSLDS**
- Required parameter you use to specify that a PRITSLDS record is to be changed at an RSR tracking subsystem. If you do not specify SLDS or TSLDS, the default is RLDS.

**STARTIME(time_stamp)**
- Required parameter you use to specify the starting time stamp of the PRISLD record that is to be changed. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

**DSN(name)**
- Optional parameter you use to change data set name. `name` can be up to 44 characters.

**CHKPTCT(value)**
- Optional parameter you use to change the number of checkpoints completed on each volume of the data set. Specify a value for each volume designated in the OLDVOL or NEWVOL parameters. If OLDVOL is specified without NEWVOL, the number of values for CHKPTCT equals the number of volume
serial numbers that appear with OLDVOL. If NEWVOL is specified, the number of values for CHKPTCT equals the number of volume serial numbers that appear in NEWVOL.

The values for CHKPTCT are:

0  No checkpoints on the volume
1  A single checkpoint on the volume
2  More than one checkpoint on the volume

**CHKPTID(chkptid)**

Optional parameter you use to change the oldest checkpoint ID for any active PST on each volume of the data set. Specify one checkpoint ID for each volume listed in OLDVOL or NEWVOL. If OLDVOL is specified without NEWVOL, the number of checkpoint IDs equals the number of volumes listed in OLDVOL. If NEWVOL is specified, the number of checkpoint IDs equals the number of volumes listed in NEWVOL.

The checkpoint ID must be in standard form for a time stamp (see “Standard Time Stamp Format” on page 124). You can specify a zero time value.

**DSSTART(time_stamp)**

is a parameter you use to specify the starting time of the data set entry to be changed. The DSSTART parameter is required if the PRISLD or PRITSLDS has multiple data set entries. The parameter is optional if the PRISLD or PRITSLDS has only one data set entry. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

**ERROR | NORMAL**

Mutually exclusive, optional parameters you use to change the data set entry to indicate whether it contains errors.

**ERROR**

is used to change the data set entry to indicate that it contains errors.

**NORMAL**

is used to change a data set entry which was previously marked as containing errors to indicate that it is normal.

**FILESEQ(value)**

Optional parameter you use to specify the file sequence number on the volume. Specify this parameter only if you specify a VOLLIST parameter. The value you substitute in the variable field must be a decimal number from 1 to 9999.

**GSG(gsgname)**

Optional parameter you use to change the global service group (GSG) name in the PRISLD record. GSG cannot be specified for PRITSLDS records.

**NEWTIME(time_stamp)**

Optional parameter you use to change the stop times of any but the last volume of the data set. When you specify NEWTIME, you must also specify OLDVOL and NEWVOL. The parameter sets, NEWTIME OLDVOL NEWVOL and RUNTIMES VOLLIST, are mutually exclusive.

When you specify NEWTIME, you must specify one less time stamp than the number of volume serial numbers specified in NEWVOL. This is because the stop time of the last volume specified in NEWVOL cannot be changed with this command. (See “NOTIFY.PRILOG (for SLDS and TSLDS)” on page 322 to learn how to specify the stop time of the final volume.) Each time stamp is used as the volume stop time of the corresponding volume serial number.
specified by NEWVOL. If not specified, the stop time of the new volume is the same as the stop time of the last—specified old volume.

Each time stamp you specify must be greater than the previous time stamp. The first time stamp in NEWTIME must be greater than or equal to the stop time of the volume prior to the changed volumes. Each time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

NEWVOL(volser)
Optional parameter you use to change the volume serial number of one or more volumes of the data set. When you specify NEWVOL, you must also specify OLDVOL.

The volume serial numbers you specify in NEWVOL replace the corresponding volume serial numbers specified in the OLDVOL parameter. You do not need to specify the same number of volume serial numbers in NEWVOL and OLDVOL. You cannot specify a volume serial number in NEWVOL that is the same as one that already exists in the PRISLD or PRITSLDS record.

You can specify from 1 to 255 volume serial numbers.

Use the NEWTIME parameter if you want to change the time stamps as well as the serial numbers of the volumes.

OLDVOL(volser)
Optional parameter you use to change the volume serial number of one or more volumes of the primary SLDS or TSLDS. When you specify OLDVOL, you must also specify NEWVOL, CHKPTCT, or CHKPTID (all described above).

The volume serial numbers you specify are those of the volumes to be changed. Each volume serial number specified must match a volume serial number in the PRISLD or PRITSLDS record.

You can specify from 1 to 255 volume serial numbers.

RUNTIMES(time_stamp)
Optional parameter you use to change the stop times of any but the last volume of the data set. This parameter is provided for compatibility with previous releases of DBRC. Use the NEWTIME, OLDVOL, and NEWVOL parameters to change the stop times of log volumes. If you do specify RUNTIMES, you must also specify VOLLIST. The following parameter sets are mutually exclusive:

• NEWTIME
• OLDVOL
• NEWVOL
• RUNTIMES
• VOLLIST

You can specify up to 255 time stamps on the RUNTIMES parameter. Each time stamp must be in standard form. (See “Standard Time Stamp Format” on page 124).

Each time stamp in the variable field must correspond to a volume in the variable field of the VOLLIST parameter. The variable fields of the RUNTIMES and VOLLIST keywords must each contain the same number of entries. Each time stamp in the variable field of the RUNTIMES parameter must be greater than the previous time stamp.

The first time stamp in the variable field of the RUNTIMES parameter must be greater than the time stamp specified for the STARTTIME parameter. The last time stamp in the variable field of the RUNTIMES parameter must be equal to
the stop time of the corresponding primary SLDS or TSLDS as specified in the
record being changed. You cannot use this command to change the stop time
of the primary SLDS or TSLDS. For information about closing open system
logs, see “NOTIFY.PRILOG (for SLDS and TSLDS)” on page 322.

SSID(name)
Optional parameter you use to specify the name of the IMS subsystem that
created the SLDS or TSLDS for which the RECON record is to be changed.

The SSID is an eight-character string consisting of any alphanumeric characters
that describe a valid IMS subsystem identification name. If you do not specify
SSID, DBRC uses the default subsystem identifier in the RECON header
record. Use the INIT.PRILOG or CHANGE.PRILOG command to set the default
subsystem identifier in the RECON header record. If you have not specified a
default in the RECON header record, you must specify SSID.

UNIT(3400 | unittype)
Optional parameter you use to change the unit type of the device on which the
data set resides. The unit type can be up to eight alphanumeric characters long.

VOLLIST(volser)
Optional parameter you use to change the record of the volume serial numbers
of the volumes that contain the data set. This parameter is provided for
compatibility with previous releases of DBRC. Use the NEWTIME, OLDVOL,
and NEWVOL parameters to change the volume serial numbers of volumes in
the data set.

If you specify the VOLLIST parameter, you must also specify the RUNTIMES
parameter. See the above description of the RUNTIMES parameter for an
explanation of how the two parameters interact. The following parameter sets
are mutually exclusive:
• NEWTIME
• OLDVOL
• NEWVOL
• RUNTIMES
• VOLLIST

Examples of Using CHANGE.PRILOG (for SLDS and TSLDS)
Here are some examples of using the CHANGE.PRILOG (for SLDS and TSLDS)
command.

Example of Changing Volume Serial Numbers and Stop Time: In this example,
some volume serial numbers and a volume stop time for a log which contains a
single data set. The example PRISLD record in RECON has a start time of
832331243299 and six volumes (VOL001, VOL002, VOL003, VOL004, VOL005, and
VOL006). The fourth volume has been copied to new volumes VOL007 and
VOL008, with the new volume stop time 832331248325 for VOL007. The PRISLD
record could be updated with the following command:

```
//CHGPRILG JOB
:
:
//SYSIN DD *
   CHANGE.PRILOG SLDS STARTTIME(832331243299) -
   OLDVOL(VOL004) -
   NEWVOL(VOL007, VOL008) -
   NEWTIME(832331248325)
/*
```

Example of Marking Primary SLDS as Normal: In this example, the first data set
of a primary SLDS is marked as normal.
CHANGE.PRILOG (for SLDS and TSLDS)

//CHGPRILG JOB

://SYSIN DD *
   CHGPRILG JOB
   SLDS STARTIME(820541212120) -
   DSSTART(820541212120) NORMAL
/*

CHANGE.RECON
Use a CHANGE.RECON command to update options in the RECON status record.

CHANGE.RECON Syntax

Notes:
1. The offset subparameter of the TIMEZONE parameter must be omitted in order to delete an entry.
Parameters for CHANGE.RECON

CATDS | NOCATDS
Mutually exclusive, optional parameters you use to modify the status of whether image copy, change accumulation, and log data sets are cataloged.

CATDS
Specifies that these data sets are cataloged. If the data set is allocated by the catalog and the CATDS option is used, DBRC bypasses volume serial and file sequence verification for the data set.

For the CATDS option to be effective, the data set must be cataloged, and VOLSER information for the data set must be omitted from the JCL. If the data set is cataloged, CATDS is specified, and VOLSER information is included in the JCL, DBRC ignores CATDS and allocates the data set by the JCL. Normal VOLSER and file sequence checking occurs.

If the data set is not cataloged, CATDS is not effective, and DBRC allocates the data set by the JCL, with VOLSER and file sequence checking.

Attention: The CATDS option affects restart of IMS from SLDS data sets. Since the CATDS option indicates the SLDS are under the control of a catalog management system, the VOLSER is not passed back to IMS for data set allocation. If the SLDS data sets are not cataloged, IMS restart fails.

NOCATDS
Specifies that these data sets, regardless of their cataloged status, are not to be treated as cataloged. In addition to data set name checking, DBRC verifies that the volume serial and file sequence numbers specified in the JCL are the same as the information recorded in the RECON.

CDSLID(value)
CDSLID sets the cross-DBRC service level ID in the RECON data set. The service level ID indicates that the maintenance corresponding to the given value has been applied to all systems that will be accessing this RECON data set.

Occasionally, maintenance must be applied to all systems that access the same RECON data set before you enable that maintenance. The ++HOLD card associated with the maintenance will indicate the need to issue the CHANGE.RECON CDSLID() command.

This command is used to enable the function provided by the maintenance after the maintenance has been applied to all the systems. The value to specify on the command will be indicated in the ++HOLD card.

After the cross-DBRC service level ID is set, all maintenance related to any service level ID less than or equal to the value set will be activated. If, after enabling some maintenance, you find that a system does not yet have the maintenance applied, to deactivate the enabled function specify a value less than the enabling service level ID. For example if service level ID 5 was set in the RECON data set, issuing a CHANGE.RECON CDSLID(3) command would deactivate functions associated with IDs 4 and 5.

The following table describes the maintenance that is associated with a particular service level ID:
Table 12. Service Level IDs and Associated Required Maintenance

<table>
<thead>
<tr>
<th>Service level ID</th>
<th>Associated maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>• PQ98438 (for IMS V7)</td>
</tr>
<tr>
<td></td>
<td>• PQ98654 (for IMS V8)</td>
</tr>
<tr>
<td></td>
<td>• PQ98655 (for IMS V9)</td>
</tr>
</tbody>
</table>

CMDAUTH(SAF,safhlq | EXIT,safhlq | BOTH,safhlq | NONE)

Optional parameter you use to specify whether DBRC command authorization is to be performed.

Specification of this parameter will result in command authorization being performed using the specified command authorization setting. However, in order to turn off command authorization (specifying NONE), you must be authorized under the current command authorization setting.

SAF | EXIT | BOTH | NONE

Mutually exclusive subparameters you use to specify your DBRC authorization option.

SAF
Specifies that a security product is to be called for command authorization.

EXIT
Specifies that the DBRC Command Authorization Exit (DSPDCAX0) is to be called for command authorization.

BOTH
Specifies that a security product and the DBRC Command Authorization Exit (DSPDCAX0) are to be called for command authorization. The security product is called first, then DSPDCAX0.

safhlq
High level qualifier of the resource name used to determine command authorization. safhlq must be specified with SAF, EXIT, or BOTH. safhlq is a name up to eight alphanumeric characters in length.

NOTE: If you simply want to change your high level qualifier, specify your current command authorization option with the new safhlq. The profiles with the new qualifier must exist before entering the command. Create the new profiles or rename the old ones.

NONE
Specifies that command authorization is to be skipped.

DASDUNIT(unittype)
Optional parameter you use to change the unit type of the DASD device that holds the records for log data sets. The unit type can be up to eight alphanumeric characters long.

DUAL | REPLACE(RECONn)

Mutually exclusive, optional parameters you use to reestablish dual mode or to replace an active RECON data set with the spare RECON:

DUAL
Causes DBRC to enter dual-RECON mode. If DBRC is already using two RECONs, the dual parameter is ignored. If DBRC is using one RECON, it attempts to use a spare RECON. If no spare RECON is available, dual mode is not entered; however, any other optional parameters are processed.
You are not required to use the DUAL parameter to cause DBRC to enter dual-RECON mode. If as a result of a permanent I/O error on a RECON, for example, DBRC is reduced to the use of a single RECON, it automatically reenters dual-RECON mode as soon as it becomes aware of the existence of a spare RECON. However, in installations that use DBRC for log control only, it can be some time before DBRC becomes aware of a recently created spare RECON. Use the CHANGE.RECON command with the DUAL parameter to cause DBRC to enter dual-RECON mode immediately.

REPLACE
Causes DBRC to replace an active RECON with a spare RECON. When you specify this parameter, you can reorganize the RECON data sets online.

Related Reading: See Chapter 4, “Initializing and Maintaining the RECON Data Set,” on page 55 for more information about using REPLACE.

For RECONn, specify the DD statement of the RECON you want replaced. For n, you can specify 1, 2, or 3. If you specify a RECON that is not active or if no spare RECON is available, the replace does not take place; however, any other optional parameters that are specified on the command are executed.

FORCER | NOFORCER
Mutually exclusive, optional parameters you use to specify whether all databases must be registered in RECON.

FORCER
Specifies that all databases must be registered in RECON. If a job tries to access an unregistered database, the database authorization call from IMS to DBRC fails.

NOFORCER
Specifies that databases do not have to be registered in RECON.

DBRC checks this parameter during initialization and it remains in effect for as long as the subsystem runs. If you change this parameter while the control region is active, the change does not take effect until restart or initialization, although the change appears in a listing of the RECON.

NOCHECK | CHECK17 | CHECK44
Mutually exclusive, optional parameters you use to change the type of comparison of log data set names that is done by DBRC.

NOCHECK
Specifies that the data set name you specify as input to DBRC has a new high-level qualifier and is longer than 17 characters. With NOCHECK, DBRC does not compare the log data set name that is recorded in RECON with the name on the appropriate DD statement.

CHECK17
Verifies that the last 17 characters of a log data set name are consistent with RECON. If the name in RECON does not match the name on the appropriate DD statement, the utility stops.

CHECK44
Verifies that the 44-character log data set name is consistent with RECON. If the name in RECON does not match the name on the appropriate log DD statement, the utility stops.
**CHANGE.RECON**

**IMSPLEX(imsplex_name) | NOPLEX**

Mutually exclusive, optional parameters you use to change the IMSplex name associated with the RECON. CHANGE.RECON cannot be used to set an initial IMSplex name in the RECON.

There can be no other DBRC instances that are registered to SCI accessing the RECON when this command is processed. The DBRC instance processing the command must be using the IMSplex name that is currently saved in the RECON.

The IMSplex name in the RECON should only be changed when your SCI registration exit, DSPSCIX0, is changed to return the new IMSplex name.

**Restriction:** CHANGE.RECON IMSPLEX and CHANGE.RECON NOPLEX can only be issued from the batch command utility. You cannot issue these commands from an online IMS.

**IMSPLEX(imsplex_name)**

Specifies the new IMSplex name. The IMSplex name can be up to 5 alphanumeric characters long.

**NOPLEX**

Specifies that no IMSplex name is to be associated with the RECON.

There is no default for this parameter. If neither is specified, the current setting is unchanged.

**Note:** If DSPSCIX0 is not found, DBRC will behave as though the sample exit were used. So, if the IMSPLEX parameter is not used, DBRC will assume that RECON Automatic Loss Notification is inactive. When the IMSPLEX parameter is used, DBRC will use its value to register with SCI.

**LISTDL | NOLISTDL**

Mutually exclusive, optional parameters you use to specify whether data set names that are deleted from the RECON (by the DELETE.L0G command or by an archive job log compression) are listed in the job output. The setting specified on this command can be overridden by the DELETE.L0G command. There is no way to override the setting for log compression during an archive job.

**LISTDL**

Specifies that deleted data set names are to be listed in the job output.

**NOLISTDL**

Specifies that deleted data set names are not to be listed in the job output.

There is no default for this parameter. If neither is specified, the current setting is unchanged.

**LOGALERT(dsnum,volnum)**

Optional parameter you use to define the threshold that triggers the DSP0287W message. Message DSP0287W displays when you just have time to shut down an online IMS subsystem before it terminates abnormally because the PRILOG record size exceeds the 16M record size maximum.

**dsnum,volnum**

These values apply only to PRILOG-family records. The message is issued when both of the following conditions are true:

- A new OLDS data set opens.
CHANGE.RECON

- When there will no longer be room in the PRILOG record to successfully archive all OLDSs currently needing to be archived (including the new one) plus dsnum more, assuming each OLDS uses volnum volumes.

The values that you enter, based on your knowledge of the rate at which the subsystem normally fills OLDSs, should be calculated to give you sufficient time to effect a normal shutdown of the online IMS subsystem.

All values must be supplied. A zero (0) in any position means that the existing value in the RECON record is not to be changed.

The default values in a new RECON or one that has been upgraded from an earlier release are (3,16), and are set during INIT.RECON command processing.

LOGRET(time interval)
Optional parameter you use to change the retention period for log data sets.

Definitions:
- The retention period is the minimum amount of time in which a log becomes inactive after it is opened. (It is then eligible to be deleted.)
- The time interval is a partial, punctuated time stamp representing a time interval (days, hours, minutes, seconds and tenths of a second) instead of date and time. The time stamp for this command follows the format described in “Standard Time Stamp Format” on page 124 except that the year subparameter element is omitted. Valid intervals range from a tenth of a second to 365 days.

Because the time interval is treated as a time stamp, message DSP0106I can be issued for incorrect values. Some examples of valid time intervals include:

- LOGRET(365)
- LOGRET('030 12.00')
- LOGRET('000 00:00:08.0')
- LOGRET('000 00,00,00,1')

The following shows two different formats for equivalent time stamp specifications. Both are valid.

| LOGRET('030') | LOGRET('030') | = 30 days |
| LOGRET('010 12,30') | LOGRET('010 12:30') | = 10 days, 12 hours, 30 minutes |

Related Reading: See “DELETE.LOG (for RLDS and SLDS)” on page 208 for more information on deleting inactive logs.

MINVERS('7.1' | '8.1' | '9.1')
Optional parameter that you use to specify the lowest version of IMS allowed to sign on. This keyword cannot be set to a higher IMS release if any subsystems with a lower version of IMS are currently signed on to DBRC.

When the UPGRADE keyword is specified on a CHANGE.RECON command and the MINVERS value is less than '7.1', the MINVERS value is set to a default of '7.1' in the new RECON.

Note: The IMS Version 9 format for the MINVERS keyword includes a period. DBRC considers this a special character and requires it to be enclosed in single quotes. The format from earlier versions (71 | 81 | 91) is also accepted for compatibility.

REORGV | NOREORGV
Mutually exclusive, optional parameters you use to modify the HALDB
partition reorganization number verification. A CHANGE command with these parameters is allowed while databases are currently authorized to other subsystems.

**REORGV**
Specifies that HALDB partition reorganization number verification is enabled. The RECON will store the reorganization number for each partition to prevent the reorganization number from being regressed, or incremented by more than one. After a CHANGE.RECON REORGV command is issued, the partition reorganization number in each partition record is set to zeros.

Note: The MINVERS value must be a minimum of 8.1 in order to set REORGV.

**NOREORGV**
Specifies that HALDB partition reorganization number verification is disabled. Any requests to store, update or verify the reorganization number will be ignored.

**SSID**(name)
Optional parameter you use to change the name of the IMS subsystem to be used as the subsystem ID for the following commands:
- CHANGE.PRILOG
- CHANGE.SECLOG
- DELETE.LOG
- GENJCL.ARCHIVE
- GENJCL.CLOSE
- NOTIFY.PRILOG
- NOTIFY.SECLOG
The SSID is an eight-character string of any alphanumeric characters that comprise a valid IMS subsystem identification name.

**SIZALERT**(dsnum,volnum,percent)
Optional parameter you use to define thresholds that trigger messages to warn you that a record has grown unusually large. The decimal threshold values that you supply for SIZALERT are:

**dsnum,volnum**
These values apply only to PRILOG-family records. The message DSP0387W is issued when both of the following conditions are true:
- When a new OLDS data set opens.
- All currently open OLDSs, including the new one, have been archived, there will no longer be room in the record for *dsnum* data set entries of *volnum* volumes each, or the record size will exceed *percent* percent of the 16M record size maximum.

**percent**
This value applies to all records. The threshold is reached when a record exceeding *percent* percentage of the 16MB record size maximum.

**STARTNEW | NONEW**
Mutually exclusive, optional parameters you use to specify whether new jobs are to be started when only one RECON is available.

**STARTNEW**
Specifies that new jobs are to be started.
NONEW
Specifies that new jobs are not to be started.

TAPEUNIT(unittype)
Optional parameter you use to specify the unit type of the tape device that holds the records for log data sets. The unit type can be up to eight alphanumeric characters long.

TRACEON | TRACEOFF
Mutually exclusive, optional parameters you use to specify whether to start or stop external tracing.

Note: If the CHANGE.RECON command is issued in a batch job, the RECON header record will be changed to show the requested status (TRACEON or TRACEOFF), however, the actual TRACE status will not be changed for any active IMS system.

If the CHANGE.RECON command is issued as a command in a control region (/RMCHANGE DBRC='RECON TRACEON | TRACEOFF'), the trace status and the RECON header record will be changed to show the requested status (TRACEON or TRACEOFF) for this control region. However, the actual TRACE status will not be changed for any other active IMS system (batch or online).

TRACEON
Starts external tracing. If you specify this parameter, the specified Generalized Trace Facility (GTF) must be active for USR-type records.

TRACEOFF
Stops external tracing. If you specify this parameter, DBRC only does internal tracing.

TIMEZONE((label,offset),(label,offset))
Optional parameter that alters the time zone label table. This parameter is used to define one or more symbolic time zone labels. Because most people do not readily associate a numeric offset with a time zone, TIMEZONE allows you to define symbolic labels, like PST (Pacific Standard Time), for numeric offsets, such as -8.

The time zone label table can contain up to 32 entries, each of which is composed of a label and an offset.

Related Reading: See “Suggestions for Time Zone Label Table Management” on page 186 for more information about the TIMEZONE parameter.

label
An alphanumeric value of up to five characters, the first of which must be alphabetic. Lowercase characters are translated to upper case.

offset
A signed-decimal value in the form of ± [h][mm] that meets the requirements of a valid time stamp offset. See “Standard Time Stamp Format” on page 124 for a description of valid offset formats. The offset is the value that, when added to UTC, gives local time. For example, the value to use for PST (Pacific Standard Time) is -8. The value for JST (Japan Standard Time) is +9.

Adding, replacing, and deleting entries from the stored list is supported as follows:
• **Adding** an entry to the stored table is accomplished when an input list entry contains both a label that does not exist in the RECON and a valid offset value.

• **Replacing** an entry to the stored table is accomplished when the input entry contains both a label that matches an existing label in the table and a valid offset value.

• **Deleting** an entry to the stored table is accomplished when the input entry is a label that matches an existing label in the table and no offset value was specified. If the offset is omitted, and the label is not found in the table, the table is not altered.

The labels in the table must be unique. Information about the use of labels versus offsets is presented in “Suggestions for Time Zone Label Table Management” on page 186.

**TIMEZIN(offset_rep [,duration])**

Optional parameter you use to define a default time zone value for time stamps that are entered without time zone information on subsequent DBRC commands.

**offset_rep**

The default time zone value. It may be one of the following choices:

**label**

A time zone label that has been previously defined using the TIMEZONE parameter.

**offset**

A numeric offset value in the same form as defined above for the TIMEZONE parameter.

**%SYS**

A keyword used to designate that the offset is to be derived from the current offset found in the z/OS CVT control block. This is the initial default for DBRC.

**duration**

Specifies the duration of the offset_rep choice.

**PERM**

Indicates that the label or offset default is to be in effect for any subsequent DBRC command running with the same RECON.

**TEMP**

Indicates that the label or offset default is in effect only for the job in which the command is entered.

**TIMEFMT(sublist)**

Optional parameter you use to define the form in which time stamps appear from DBRC in messages, displays, and listings. See “Standard Time Stamp Format” on page 124 for examples of the different output forms. The five values are positional. Each is optional and can be omitted by including only the comma.

The format of the TIMEFMT parameter sublist is presented in detail in “TIMEFMT Parameter” on page 125.

**UPGRADE**

Optional parameter that you use to upgrade all IMS Version 7 and IMS Version 8 RECON records to an IMS Version 9 level. You do not need to shut down other IMS subsystems that share the RECON, although they might have to
wait for the command to complete. When UPGRADE is specified, a default of 71 is set for the MINVERS value in the new RECON.

**Restriction:** You can issue the CHANGE.RECON UPGRADE command only in the batch command utility. You cannot issue it from an online IMS.

CHANGE.RECON UPGRADE upgrades the RECON in two stages: first, Copy 1 and then Copy 2.

Message DSP02501 indicates the beginning of the first stage. Any failure during this stage causes DBRC to reconfigure the RECONs, discarding Copy 1. If a failure occurs during the first stage, you need to rerun the CHANGE.RECON UPGRADE command.

After successful completion of the first stage, message DSP02511 indicates that the second stage is beginning. Any failure during this stage causes DBRC to reconfigure the RECONs, discarding Copy 2. However, if there is a failure during the second stage, you do not need to rerun the CHANGE.RECON UPGRADE command.

You can use the CHANGE.RECON UPGRADE command in either nonconcurrent or concurrent mode. The mode is determined by the environment at the time you issue the command.

**Nonconcurrent mode**

No other jobs are currently accessing the RECON.

Before issuing the CHANGE.RECON UPGRADE command, you might want to create a backup copy of the RECON in case the upgrade and the RECON recovery procedures fail, which can happen if you have not allocated two RECONs and a spare.

**Concurrent mode**

Other jobs are currently accessing the RECON. These jobs can be either Version 7 or Version 8 and must have the respective Version 9 SPE applied. Two RECONs plus a spare must be allocated for a concurrent upgrade.

When an upgrade is performed, an existing HALDB is marked (in the RECON) to indicate that it is not capable of being reorganized online with the HALDB Online Reorganization function. To enable the HALDB for online reorganization, use the CHANGE.DB DBD(HALDB_master)OLRCAP command.

**Suggestions for Time Zone Label Table Management**

The same offset should not be designated by more than one label because DBRC always uses the first occurrence in the table when outputting a time stamp.

The practicality of using the label format is affected by the scope of the IMS installation. For those operating solely in a single time zone, use of labels eliminates the need for the operator to know the exact offset to UTC at all times during the year. For multiple time zone operation, the use of offsets rather than labels, is suggested (though not mandatory). The time zone label table may not be practical if offsets are not unique from one zone to the next when daylight saving time adjustments are taken into account. Changing the table when daylight saving time switches are made would add to the confusion, so, in that case, use numeric offset values for cross time zone operation.

**Example of Updating the RECON Header Record**

In this example, you are:

- Forcing all databases to be registered
Changing the default subsystem ID to IMSB
And changing the log retention period to 7 days

//CHGRECON JOB

:
//SYFIN DD *
  CHANGE.RECON SSID(IMSB) FORCER LOGRET('007')
/*

CHANGE.SECLOG (for OLDS)
Use a CHANGE.SECLOG (for OLDS) command to change information in RECON about a secondary OLDS.

CHANGE.SECLOG (for OLDS) Syntax

\[
\text{CHANGE.SECLOG} \text{OLDS}(ddname) \text{AVAIL|UNAVAIL} \text{DSN}(name) \text{ERROR|NORMAL} \text{SSID}(name)
\]

Parameters for CHANGE.SECLOG (for OLDS)

OLDS(ddname)
Required parameter you use to specify the OLDS for which the RECON record is to be changed. Failure to specify this parameter results in an RLDS being changed.

AVAIL | UNAVAIL
Mutually exclusive, optional parameters you use to change the SECOLDS to indicate its availability.

AVAIL
Indicates that the OLDS contains valid data and that it can be used as input to the Log Archive utility.

UNAVAIL
Indicates that the OLDS contains invalid data and it should not be used as input to the Log Archive utility.

DSN(name)
Optional parameter you use to change the name of a secondary OLDS. The name you substitute in the variable field can be up to 44 characters long.

ERROR | NORMAL
Mutually exclusive, optional parameters you use to change the specified SECOLDS record to indicate whether it contains errors.

ERROR
Changes the RECON record to indicate that a specified OLDS contains errors, so IMS is unable to close the OLDS properly. The OLDS must be closed before it can be used as input to the Log Archive utility.

When you use dual logging, you use ERROR to change a specified SECOLDS record to indicate that it contains errors. The subsystem uses the data in the error-free OLDS to close the OLDS that is marked ERROR.

NORMAL
Changes the SECOLDS record, which was previously marked as containing
errors, to indicate that the data set is now available for use as input to any log utility. When you specify NORMAL for a secondary OLDS, the record immediately indicates that the next primary OLDS is no longer needed in order to close the corresponding primary OLDS.

SSID(name)
Optional parameter you use to specify the name of the IMS subsystem that created the OLDS for which the RECON record is to be changed.

The SSID is an eight-character string consisting of any alphanumeric characters that comprise a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. Use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

Example Showing a SECOLDS Error
In this example, a SECOLDS that IMS online subsystem IMSA created is known to be in error.

//CHGSECLG JOB
:
//SYSIN DD *
  CHANGE.SECLOG OLDS(DFSOLS02) -
    SSID(IMSA) ERROR
/*

CHANGE.SECLOG (for RLDS)
You can use the CHANGE.SECLOG (for RLDS) command to change information in the RECON about a primary RLDS (or an SLDS that a batch subsystem created). Use the NOTIFY.SECLOG (for RLDS) command to add a SECLOG record or to add data set entries to an existing SECLOG record.

With the exception of the GSG name, all the information you can change resides in a data set entry of the SECLOG record. Each CHANGE.SECLOG command you issue changes only one data set entry. If the log has multiple data sets, you must use the DSSTART parameter to identify the data set entry to be changed. (Note that if you are only changing the GSG you must still specify DSSTART if the log has more than one data set.)

If the SECLOG record represents log data that was received by an RSR tracking site from an active IMS subsystem, none of the keywords FILESEQ, NEWTIME, NEWVOL, OLDVOL, RUNTIMES, CHKPTID, UNIT, or VOLLIST can be specified. Log data sets received at a tracking site must be cataloged.

CHANGE.SECLOG (for RLDS) Syntax

```
CHANGE.SECLOG(R LDS) STARTIME(time_stamp) DSN(name)
  CHKPTCT(value) CHKPTID(chkptid)
```
### CHANGE.SECLOG (for RLDS)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSSTART</td>
<td>Specifies the starting time stamp of the SECLOG record that is to be changed. The time stamp must be in standard form.</td>
</tr>
<tr>
<td>ERROR</td>
<td>Controls error handling.</td>
</tr>
<tr>
<td>FILESEQ</td>
<td>Specifies the file sequence number.</td>
</tr>
<tr>
<td>GSG</td>
<td>Specifies the global synchronous group.</td>
</tr>
<tr>
<td>NEWTIME</td>
<td>Specifies the new time stamp of the SECLOG record.</td>
</tr>
<tr>
<td>NEWVOL</td>
<td>Specifies the new volume serial number.</td>
</tr>
<tr>
<td>OLDVOL</td>
<td>Specifies the old volume serial number.</td>
</tr>
<tr>
<td>RUNTIMES</td>
<td>Specifies the run times.</td>
</tr>
<tr>
<td>UNIT</td>
<td>Specifies the unit type.</td>
</tr>
<tr>
<td>VOLLIST</td>
<td>Specifies the volume list.</td>
</tr>
</tbody>
</table>

### Parameters for CHANGE.SECLOG (for RLDS)

**RLDS**

Is the parameter you can use to specify that a SECLOG record is to be changed. Since RLDS is the default, if you do not specify a record type as the first parameter for CHANGE.SECLOG, RLDS is assumed.

**STARTIME**(time stamp)

Required parameter you use to specify the starting time stamp of the SECLOG record that is to be changed. The time stamp must be in standard form (see "Standard Time Stamp Format" on page 124).

**DSN**(name)

Optional parameter you use to change data set name. name can be up to 44 characters.

**CHKPTCT**(value)

Optional parameter you use to change the number of checkpoints completed on each volume of the data set. Specify a value for each volume designated in the OLDVOL or NEWVOL parameters. If OLDVOL is specified without NEWVOL, the number of values for CHKPTCT equals the number of volume serial numbers that appear with OLDVOL. If NEWVOL is specified, the number of values for CHKPTCT equals the number of volume serial numbers that appear in NEWVOL.

The values for CHKPTCT are:

- 0: No checkpoints on the volume
- 1: A single checkpoint on the volume
- 2: More than one checkpoint on the volume

**CHKPTID**(chkptid)

Optional parameter you use to change the oldest checkpoint ID for any active PST on each volume of the data set. Specify one checkpoint ID for each volume listed in OLDVOL or NEWVOL. If OLDVOL is specified without...
NEWVOL, the number of checkpoint IDs equals the number of volumes listed in OLDVOL. If NEWVOL is specified, the number of checkpoint IDs equals the number of volumes listed in NEWVOL.

The checkpoint ID must be in standard form for a time stamp (see “Standard Time Stamp Format” on page 124). You can specify a zero time value.

**DSSTART** *(time_stamp)*

Is a parameter you use to specify the starting time of the data set entry to be changed. The DSSTART parameter is required if the SECLOG has multiple data set entries; it is optional if the SECLOG has only one data set entry. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

**ERROR | NORMAL**

Mutually exclusive, optional parameters you use to change the data set entry to indicate whether it contains errors.

**ERROR**

changes the data set to indicate that it contains errors and should not be used as input to any DBRC-controlled run of a recovery utility.

**NORMAL**

changes a data set which was previously marked as containing errors to indicate that it is now available for use as input to any recovery utility.

**FILESEQ** *(value)*

Optional parameter you use to specify the file sequence number on the volume. Specify this parameter only if you specify a VOLLIST parameter. The value you substitute in the variable field must be a decimal number from 1 to 9999.

**GSG** *(gsgname)*

Optional parameter you use to change the global service group (GSG) name in the SECLOG record.

**NEWTIME** *(time_stamp)*

Optional parameter you use to change the stop times of any but the last volume of the data set. If you specify NEWTIME, you must also specify OLDVOL and NEWVOL. The following parameter sets are mutually exclusive:

- NEWTIME
- OLDVOL
- NEWVOL
- RUNTIMES
- VOLLIST

If you specify NEWTIME, you must specify one less time stamp than the number of volume serial numbers specified in NEWVOL. This is because the stop time of the last volume specified in NEWVOL cannot be changed with this command. (See “NOTIFY.PRLOG (for RLDS)” on page 317 to see how to specify the stop time of the final volume.) Each time stamp is used as the volume stop time of the corresponding volume serial number specified by NEWVOL. If not specified, the stop time of the new volume is the same as the stop time of the last-specified old volume.

Each time stamp you specify must be greater than the previous time stamp. The first time stamp in NEWTIME must be greater than or equal to the stop time of the volume immediately preceding the changed volumes. Each time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).
NEWVOL(volser)
Optional parameter you use to change the volume serial number of one or more volumes of the data set. If you specify NEWVOL, you must also specify OLDVOL (described below).

The volume serial numbers you specify in NEWVOL replace the corresponding volume serial numbers specified in the OLDVOL parameter. You do not need to specify the same number of volume serial numbers in NEWVOL and OLDVOL. You cannot specify a volume serial number in NEWVOL that is the same as one which already exists in the SECLOG record.

You can specify from 1 to 255 volume serial numbers.

Use the NEWTIME parameter to change the time stamps as well as the serial numbers of the volumes.

OLDVOL(volser)
Optional parameter you use to change the volume serial number of one or more volumes of the data set. If you specify OLDVOL, you must also specify NEWVOL, CHKPTCT, or CHKPTID.

The volume serial numbers you specify are those of the volumes being changed. Each volume serial number specified in OLDVOL must match a volume serial number in the SECLOG record.

You can specify from 1 to 255 volume serial numbers.

RUNTIMES(time_stamp)
Optional parameter you use to change the stop times of any but the last volume of the data set. This parameter is provided for compatibility with previous releases of DBRC. Use the NEWTIME, OLDVOL, and NEWVOL parameter set to change the stop times of log volumes. If you do specify RUNTIMES, you must also specify VOLLIST. The following parameter sets are mutually exclusive:

- NEWTIME
- OLDVOL
- NEWVOL
- RUNTIMES
- VOLLIST

You can specify up to 255 time stamps on the RUNTIMES parameter. Each time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

Each time stamp in the variable field must correspond to a volume in the variable field of the VOLLIST parameter. The variable fields of the RUNTIMES and VOLLIST keywords must each contain the same number of entries. Each time stamp in the variable field of the RUNTIMES parameter must be greater than the previous time stamp.

The first time stamp in the variable field of the RUNTIMES parameter must be greater than the time stamp specified for the STARTTIME parameter. The last time stamp in the variable field of the RUNTIMES parameter must be equal to the stop time of the corresponding secondary RLDS as specified in the record being changed. You cannot use this command to change the stop time of the secondary RLDS. For information about closing open recovery logs, see “NOTIFY.PRILOG (for RLDS)” on page 317.

UNIT(3400 | unittype)
Optional parameter you use to change the unit type of the device on which the data set resides. The unit type can be up to eight alphanumeric characters long.
CHANGE.SECLOG (for RLDS)

VOLLIST(volser)
Optional parameter you use to change the record of the volume serial numbers of the volumes that contain the data set. This parameter is provided for compatibility with previous releases of DBRC. Use the NEWTIME, OLDVOL, and NEWVOL parameter set to change the volume serial numbers of volumes in the data set.

If you specify the VOLLIST parameter, you must also specify the RUNTIMES parameter. See the description of the RUNTIMES parameter under "Parameters for CHANGE.PRIOLOG (for RLDS)" on page 169 for an explanation of how the two parameters interact. The following parameter sets are mutually exclusive:

- NEWTIME
- OLDVOL
- NEWVOL
- RUNTIMES
- VOLLIST

Examples of Using CHANGE.SECLOG (for RLDS)
Here are some examples of using the CHANGE.SECLOG (for RLDS) command.

Example of Changing Volume Serial Numbers: In this example, some volume serial numbers are changed. The example SECLOG record in RECON has one data set with six volumes (VOL001, VOL002, VOL003, VOL004, VOL005, and VOL006) and a start time of 832331243299. The serial numbers of the third and fourth volumes are replaced with three others by the following command:

```
//CHGSECLG JOB
:
//SYSIN DD *
CHANGE.SECLOG RLDS STARTIME(832331243299) -
OLDVOL(VOL003,VOL004) -
NEWVOL(VOL007,VOL008,VOL009)
/*
```

Example of Changing Volume Stop Times: In this example, STARTIME identifies the SECLOG record and DSSTART identifies its first data set entry, of which the data set name and the stop times of three volumes are to be changed.

```
//CHGSECLG JOB
:
//SYSIN DD *
CHANGE.SECLOG RLDS STARTIME(840541212120) -
DSSTART(840541212120) -
DSN(IMS.SECLOG.SEC001.DSN) -
VOLLIST(VOL001,VOL002,VOL993) -
RUNTIMES(840541212122,840541313133,840541515150)
/*
```

CHANGE.SECLOG (for SLDS and TSLDS)
You can use the CHANGE.SECLOG (for SLDS) command to change information in the RECON about a secondary SLDS for an online system. You can use the CHANGE.SECLOG (for TSLDS) command to change information in the RECON about a secondary SLDS for an RSR tracking subsystem. Use CHANGE.SECLOG (for RLDS) to change information about an SLDS that a batch subsystem created, because DBRC considers such data to be an RLDS. Use the NOTIFY.SECLOG (for SLDS) command to add a SECSLD record or to add data set entries to an existing SECSLD record.
With the exception of the GSG name, all the information you can change resides in a data set entry of the SECSLD record. Each CHANGE.SECLOG command you issue changes only one data set entry. If the log has multiple data sets, you must use the DSSTART parameter to identify the data set entry to be changed. (Note that if you are only changing the GSG, you must still specify DSSTART if the log has more than one data set.)

If the SECSLD record represents log data that was received by an RSR tracking site from an active IMS subsystem, none of the keywords FILESEQ, NEWTIME, NEWVOL, OLDVOL, RUNTIMES, CHKPTID, UNIT, or VOLLIST can be specified. Log data sets received at a tracking site must be cataloged.

**CHANGE.SECLOG (for SLDS and TSLDS) Syntax**

```plaintext
CHANGE.SECLOG SLDS STARTIME(time_stamp) DSN(name)

<table>
<thead>
<tr>
<th>CHKPTCT(value)</th>
<th>CHKPTID(chkptid)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSSTART(time_stamp)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ERROR</th>
<th>FILESEQ(value)</th>
<th>GSG(gsgname)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEWTIME(time_stamp)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OLDVOL(volser)</th>
<th>NEWVOL(volser)</th>
<th>RUNTIMES(time_stamp)</th>
</tr>
</thead>
</table>

SSID(name) 3400 UNIT(unittype) VOLLIST(volser)
```

**Parameters for CHANGE.SECLOG (for SLDS and TSLDS)**

**SLDS**
- Required parameter you use to specify that an SECSLD record is to be changed.

**TSLDS**
- Required parameter you use to specify that a SECTSLDS record is to be changed at an RSR tracking subsystem. If you do not specify SLDS or TSLDS, the default is RLDS.

**STARTIME(time_stamp)**
- Required parameter you use to specify the starting time stamp of the SECSLD record that is to be changed. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

**DSN(name)**
- Optional parameter you use to change data set name. name can be up to 44 characters.

**CHKPTCT(value)**
- Optional parameter you use to change the number of checkpoints completed on each volume of the data set. Specify a value for each volume designated in the OLDVOL or NEWVOL parameters. If OLDVOL is specified without
CHANGE.SECLOG (for SLDS and TSLDS)

NEWVOL, the number of values for CHKPTCT equals the number of volume serial numbers that appear with OLDVOL. If NEWVOL is specified, the number of values for CHKPTCT equals the number of volume serial numbers that appear in NEWVOL.

The values for CHKPTCT are:

0 No checkpoints on the volume
1 A single checkpoint on the volume
2 More than one checkpoint on the volume

CHKPTID(chkptid)
Optional parameter you use to change the oldest checkpoint ID for any active PST on each volume of the data set. Specify one checkpoint ID for each volume listed in OLDVOL or NEWVOL. If OLDVOL is specified without NEWVOL, the number of checkpoint IDs equals the number of volumes listed in OLDVOL. If NEWVOL is specified, the number of checkpoint IDs equals the number of volumes listed in NEWVOL.

The checkpoint ID must be in standard form for a time stamp (see “Standard Time Stamp Format” on page 124). You can specify a zero time value.

DSSTART(time_stamp)
Is a parameter you use to specify the starting time of the data set entry to be changed. The DSSTART parameter is required if the SECSLD or SECTSLDS has multiple data set entries. The parameter is optional if the SECSLD or SECTSLDS has only one data set entry. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

ERROR | NORMAL
Mutually exclusive, optional parameters you use to change the data set entry to indicate whether it contains errors.

ERROR Is used to change the data set entry to indicate that it contains errors.

NORMAL Is used to change a data set entry which was previously marked as containing errors to indicate that it is normal.

FILESEQ(value)
Optional parameter you use to specify the file sequence number on the volume. Specify this parameter only if you specify a VOLLIST parameter. The value you substitute in the variable field must be a decimal number from 1 to 9999.

GSG(gsgname)
Optional parameter you use to change the global service group (GSG) name in the SECSLD record. GSG cannot be specified for SECTSLDS records.

NEWTIME(time_stamp)
Optional parameter you use to change the stop times of any but the last volume of the data set. When you specify NEWTIME, you must also specify OLDVOL and NEWVOL. The following parameter sets are mutually exclusive:
- NEWTIME
- OLDVOL
- NEWVOL
- RUNTIMES
- VOLLIST
When you specify NEWTIME, you must specify one less time stamp than the number of volume serial numbers specified in NEWVOL. This is because the stop time of the last volume specified in NEWVOL cannot be changed with this command. (See "NOTIFY.PRILOG (for SLDS and TSLDS)" on page 322 to learn how to specify the stop time of the final volume.) Each time stamp is used as the volume stop time of the corresponding volume serial number specified by NEWVOL. If not specified, the stop time of the new volume is the same as the stop time of the last—specified old volume.

Each time stamp you specify must be greater than the previous time stamp. The first time stamp in NEWTIME must be greater than or equal to the stop time of the volume prior to the changed volumes. Each time stamp must be in standard form (see "Standard Time Stamp Format" on page 124).

**NEWVOL**(volser)
Optional parameter you use to change the volume serial number of one or more volumes of the data set. When you specify NEWVOL, you must also specify OLDVOL.

The volume serial numbers you specify in NEWVOL replace the corresponding volume serial numbers specified in the OLDVOL parameter. You do not need to specify the same number of volume serial numbers in NEWVOL and OLDVOL. You cannot specify a volume serial number in NEWVOL that is the same as one that already exists in the SECSLD or SECTSLDS record.

You can specify from 1 to 255 volume serial numbers.

Use the NEWTIME parameter if you want to change the time stamps as well as the serial numbers of the volumes.

**OLDVOL**(volser)
Optional parameter you use to change the volume serial number of one or more volumes of the secondary SLDS or TSLDS. When you specify OLDVOL, you must also specify NEWVOL, CHKPTCT, or CHKPTID (all described above).

The volume serial numbers you specify are those of the volumes to be changed. Each volume serial number specified must match a volume serial number in the SECSLD or SECTSLDS record.

You can specify from 1 to 255 volume serial numbers.

**RUNTIMES**(time_stamp)
Optional parameter you use to change the stop times of any but the last volume of the data set. This parameter is provided for compatibility with previous releases of DBRC. Use the NEWTIME, OLDVOL, and NEWVOL parameter set to change the stop times of log volumes. If you do specify RUNTIMES, you must also specify VOLLIST. The following parameter sets are mutually exclusive:

- NEWTIME
- OLDVOL
- NEWVOL
- RUNTIMES
- VOLLIST

You can specify up to 255 time stamps on the RUNTIMES parameter. Each time stamp must be in standard form. (See "Standard Time Stamp Format" on page 124).

Each time stamp in the variable field must correspond to a volume in the variable field of the VOLLIST parameter. The variable fields of the RUNTIMES
CHANGE.SECLOG (for SLDS and TSLDS)

and VOLLIST keywords must each contain the same number of entries. Each time stamp in the variable field of the RUNTIMES parameter must be greater than the previous time stamp.

The first time stamp in the variable field of the RUNTIMES parameter must be greater than the time stamp specified for the STARTIME parameter. The last time stamp in the variable field of the RUNTIMES parameter must be equal to the stop time of the corresponding secondary SLDS or TSLDS as specified in the record being changed. You cannot use this command to change the stop time of the secondary SLDS or TSLDS. For information about closing open system logs, see "NOTIFY.PRILG (for SLDS and TSLDS)" on page 322.

SSID(name)
Optional parameter you use to specify the name of the IMS subsystem that created the SLDS or TSLDS for which the RECON record is to be changed.

The SSID is an eight-character string consisting of any alphanumeric characters that describe a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. Use the INIT.SECLOG or CHANGE.SECLOG command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

UNIT(3400 | unittype)
Optional parameter you use to change the unit type of the device on which the data set resides. The unit type can be up to eight alphanumeric characters long.

VOLLIST(volser)
Optional parameter you use to change the record of the volume serial numbers of the volumes that contain the data set. This parameter is provided for compatibility with previous releases of DBRC. You should use the new parameter set, NEWTIME OLDVOL NEWVOL, to change the volume serial numbers of volumes in the data set.

If you specify the VOLLIST parameter, you must also specify the RUNTIMES parameter. See the above description of the RUNTIMES parameter for an explanation of how the two parameters interact. The parameter sets, NEWTIME OLDVOL NEWVOL and RUNTIMES VOLLIST, are mutually exclusive.

Examples of Using CHANGE.SECLOG (for SLDS and TSLDS)
Here are some examples of using the CHANGE.SECLOG (for SLDS and TSLDS) command.

Example of Changing Volume Serial Numbers and Stop Time: In this example, some volume serial numbers and a volume stop time of an SECSLD are changed. The example SECSLD record in RECON has six volumes (VOL001, VOL002, VOL003, VOL004, VOL005, and VOL006) and a start time of 83233124329. The fourth volume has been copied to new volumes VOL007 and VOL008 with the new volume stop time 832331248325 for VOL007. The SECSLD record is updated with the following command:

```
//CHGPRILG JOB
:
//SYSIN    DD    *
    CHANGE.SECLOG SLDS STARTIME(832331243299)    -
                OLDVOL(VOL004)      -
                NEWVOL(VOL007, VOL008) -
                NEWTIME(832331248325)  */
```
Example of Marking the Secondary SLDS as Normal: In this example, the first and only data set of a secondary SLDS is being marked as normal.

```
//CHGPRILG JOB
:
//SYSIN DD +
   CHANGE.SECLOG SLDS STARTIME(840541212120) -
               DSSTART(840541212120) NORMAL
/*
```

**CHANGE.SG**

Use a CHANGE.SG command to change the role of a service group (SG). The role of a service group cannot be changed while a subsystem is signed on to its global service group.

This command fails if RSRFEAT=NO is specified in the IMSCTRL macro.

If two SG entries are present at the time this command is issued, the other SG is assigned the complementary attributes of the SG that is named in the command.

**CHANGE.SG Syntax**

```
CHANGE.SG—SGNAME(sgname)—GSGNAME(gsgname)
   ACTIVE | TRACKING
   LOCAL | NONLOCAL
   NORTA
```

**Parameters for CHANGE.SG**

- **SGNAME(sgname)**: Required parameter you use to specify the service group name.
- **GSGNAME(gsgname)**: Required parameter you use to specify the global service group name.
- **ACTIVE | TRACKING**: Optional parameter you use to specify the new role of the service group. This parameter is sometimes referred to as the STATUS parameter.
- **LOCAL | NONLOCAL**: Optional parameter you use to specify whether the service group is local or nonlocal for this set of RECONs. This parameter is sometimes referred to as the LOCALE parameter.
- **NORTA**: Optional parameter you use to specify that you do not want to continue a remote takeover that is currently in progress. This parameter turns off takeover indicators in the RECON. This parameter is valid for either the active or tracking subsystem.

When you specify the NORTA parameter, you must specify the appropriate STATUS parameter (either ACTIVE or TRACKING), and you cannot specify the LOCALE parameter (LOCAL | NONLOCAL).

If you use NORTA when no remote takeover is in progress, message DSP0144I is issued. You should wait to receive a takeover-in-progress message before using NORTA.

**Example of Changing the Status of a Service Group**

In this example, the Service Group’s role or status is changed to ACTIVE.
In this example, SG1 is changed to a tracking role:

```
CHANGE.SG SGNAME(SG1) GSGNAME(GSG1) TRACKING
```

In this case, SG2 is automatically changed to an active role.

### CHANGE.SUBSYS

Use a `CHANGE.SUBSYS` command to modify information that is contained in a subsystem record in RECON.

#### CHANGE.SUBSYS Syntax

```
CHANGE.SUBSYS
  SSID(name) | BACKIRLM(name) | IRLMID(name) | NORMAL | ABNORMAL
  BACKIRLM | NOBACKUP | IRLMID
  STARTRCV | TRACKING | ENDRECOV
```

#### Parameters for CHANGE.SUBSYS

**SSID(name) | ALL**

Optional parameters you use to specify the subsystem you are using. The SSID is an eight-character string consisting of any alphanumeric characters that comprise a valid z/OS or IMS subsystem identification name. ALL is the default, which will process this command for every subsystem that communicates with the specified Internal Resource Lock Manager (IRLM).

**BACKIRLM(name) | NOBACKUP**

Mutually exclusive, optional parameters you use to change the specification of the alternate subsystem.

**BACKIRLM**

Adds an alternate subsystem IRLM ID to the active subsystem record. When you specify BACKIRLM, you must also specify IRLMID. DBRC locates the specified subsystem record and adds or changes the IRLM ID of the alternate subsystem.

**NOBACKUP**

Deletes the IRLM ID of the alternate subsystem from the active subsystem record. This also resets the flags in the subsystem record to indicate that the alternate subsystem is signed on. This command might be required prior to restarting the alternate subsystem.

**Restriction:** You cannot use NORMAL or ABNORMAL, or STARTRCV or ENDRECOV with this parameter.

**IRLMID(name)**

Optional parameter you use to specify the name of the IRLM with which the subsystem is communicating. The IRLMID is a five-character string consisting of any alphanumeric characters.
**Restriction:** You cannot change the IRLM ID. Specify the IRLM ID in order to change processing mode of a subsystem.

**NORMAL | ABNORMAL**

Mutually exclusive, optional parameters you use to specify the status of the subsystem.

**NORMAL**

Specifies that normal processing is to continue for the subsystem.

**ABNORMAL**

Indicates that the subsystem has abnormally ended. When ABNORMAL is specified, DBRC does the following:

- Removes authorization for any databases that have not been updated but are authorized for the specified subsystem.
- Flags the identified subsystem entry as having been abnormally ended.
- Turns off the recovery-processing-started flag.
- If the subsystem is batch and no databases were updated, then the subsystem record is deleted.

**Restriction:** If you specify STARTRCV or ENDRECOV, you cannot specify ABNORMAL.

**STARTRCV | ENDRECOV**

Mutually exclusive, optional parameters you use to specify whether a signon recovery has completed successfully.

**STARTRCV**

Indicates a signon recovery start.

**ENDRECOV**

removes authorization for all databases that the specified subsystem authorized.

If you want to delete all database authorizations from a subsystem, you must issue the CHANGE.SUBSYS STARTRCV command and then issue the CHANGE.SUBSYS ENDRECOV command. These two commands simulate the signon recovery start and signon recovery complete calls.

**Recommendation:** Do not use this sequence of commands unless an abnormal end occurred. Otherwise, you remove authorization for the databases that an active subsystem is currently using.

If, after using STARTRCV | ENDRECOV and DELETE.SUBSYS commands, subsystem information is still associated with the database, a CHANGE.DB command with the NOBACK parameter is required in order to clear the remaining subsystem ID from the database record.

**TRACKING**

Specifies that information about the RSR tracking subsystem is to be changed.

**Attention:** If you specify TRACKING, do not specify STARTRCV or ENDRECOV.

**Example of Identifying the IRLM**

In this example, IRLMID identifies the IRLM that is communicating with the subsystem identified in the SSID parameter. In addition, ABNORMAL indicates that this subsystem abnormally ended.
CHANGE.UIC

Use a CHANGE.UIC command to modify information in the image copy record in RECON that corresponds to a nonstandard image copy data set.

CHANGE.UIC Syntax

```
CHANGE.UIC  DBD(name)  DDN(name)  AREA(name)  RECTIME(time_stamp)  UDATA('string')
```

Parameters for CHANGE.UIC

**DBD(name)**

Required parameter you use to identify the database name of the DBDS for which a nonstandard image copy data set exists.

**Restrictions for HALDBs**: For HALDBs, name specifies the name of a HALDB partition.

**DDN(name) | AREA(name)**

Mutually exclusive, required parameters you use to identify the name of the DBDS or area for which the nonstandard image copy data set exists.

**RECTIME(time_stamp)**

Required parameter you use to identify the specific image copy record of the nonstandard image copy data set that is to be modified. Use the time stamp with an adjacent asterisk (*) in a listing of the IMAGE record. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

**UDATA('string')**

Required parameter you use to change the user data in the identified image copy record. string can be up to 80 characters and must appear in single quotation marks.

Example of Changing the Nonstandard ICDSN in RECON

In this example, information in RECON about the nonstandard image copy data set identified by the RECTIME parameter is to be changed. The UDATA parameter specifies the new information that is to be recorded for the specified image copy data set.

```
//CHGUC  JOB

//SYSIN  DD *
   CHANGE.UIC  DBD(DBDS1)  AREA(AREA003)  
          RECTIME(840651010100)  
          UDATA('DUMP OF VOLUME VOL001 AT 840651010100')
/*
```
Chapter 10. DELETE Commands

Use the DELETE commands to delete information from a RECON.

DELETE Commands In This Chapter

The following is a list of all the DELETE commands:

- "DELETE.ADS"
- "DELETE.ALSQ" on page 202
- "DELETE.BKOUT" on page 203
- "DELETE.CA" on page 203
- "DELETE.CAGRP" on page 204
- "DELETE.DB" on page 204
- "DELETE.DBDS" on page 205
- "DELETE.DBDSGRP" on page 206
- "DELETE.GSG" on page 206
- "DELETE.IC" on page 207
- "DELETE.LOG (for OLDS)" on page 208
- "DELETE.LOG (for RLDS and SLDS)" on page 208
- "DELETE.PART" on page 211
- "DELETE.RECOV" on page 212
- "DELETE.REORG" on page 212
- "DELETE.SG" on page 213
- "DELETE.SUBSYS" on page 214
- "DELETE.UIC" on page 214

DELETE.ADS

Use a DELETE.ADS command to delete an ADS from its associated area in the RECON record structure. An area can consist of a maximum of seven ADSs. The ADS that is to be deleted must have been registered by an INIT.ADS command.

The DELETE.ADS command fails if you issue it while the area is authorized and the ADS is in AVAILABLE status. The command can be used if the ADS is in UNAVAILABLE status, provided that the ADS Create utility is not running.

DELETE.ADS Syntax

```
DELETE.ADS ADDN(name) AREA(name) DBD(name)
```

Parameters for DELETE.ADS

ADDN(name)
Required parameter you use to identify the area name of the ADS to be deleted.

AREA(name)
Required parameter you use to identify the name of the area that contains the ADS to be deleted.
DELETE.ADS

DBD(name)
Required parameter you use to identify the database name of the area that is
to be deleted.

Example of Deleting an ADS Record
In this example an ADS record is deleted from RECON for the DEDB area that is
identified by the DBD, AREA, and ADDN parameters.

```
//DELAADS JOB ('LEOPARD,IMS','LEOPARD',REGION=880K,
  :
//SYSIN DD *
  DELETE.ADS DBD(DBD00001) AREA(AREA0001) -
  ADDN(AREA0002)
/*
```

DELETE.ALOC
Use a DELETE.ALOC command to delete from RECON an allocation record that is
related to a specified DBDS or DEDB area. An allocation record can be deleted
only when it contains a deallocation time or when its associated log has a stop
time. Except for deleting allocation records that precede the oldest image copy data
set for a DBDS or DEDB area, deleting an allocation record should be done with
cautions, and is not normally required. Deleting an allocation record that represents
a period of time during which you changed the specified DBDS or area can cause a
future recovery to be incorrect.

DELETE.ALOC Syntax
```
DELETE.ALOC DBD(name) DDN(name) AREA(name)
```

Parameters for DELETE.ALOC

DBD(name)
Required parameter you use to identify the database name of the DBDS or area
for which the allocation record is to be deleted.

Restriction for HALDBs: For the DELETE.ALOC command, name specifies the
name of a HALDB partition.

DDN(name) | AREA(name)
Mutually exclusive, optional parameters you use to identify the data set
ddname of the DBDS or DEDB area for which the allocation record is to be
deleted.

RECTIME(time_stamp)
Required parameter you use to identify the specific allocation record to be
deleted for a specified DBDS or DEDB area. Use the time stamp with an
adjacent asterisk (*) in a listing of the ALOC record. The time stamp must be
in standard form (see "Standard Time Stamp Format" on page 124).

Example of Deleting an Allocation Record
In this example, an allocation record is deleted from RECON for the DBDS that is
identified by the DBD and DDN parameters. The RECTIME parameter identifies
the specific allocation record that is to be deleted.
DELETE.BKOUT

Use a DELETE.BKOUT command to delete backout records from the RECON.

Use this command, for example, following the successful restore of a recent image copy. The backout information held in RECON at the time of the copy is meaningless, but DBRC is not aware of this fact, and DBRC does not delete the backout records automatically.

**Attention:** Use the DELETE.BKOUT command with extreme caution. It deletes all backout information for a subsystem from the RECON; this is information that DBRC uses to help IMS maintain database integrity.

**DELETE.BKOUT Syntax**

```plaintext
DELETE.BKOUT_SSID(name)
```

**Parameters for DELETE.BKOUT**

**SSID(name)**

Required parameter you use to identify the subsystem for which a backout record is to be deleted. The subsystem name is an eight-character, alphanumeric string that represents any valid subsystem. You can specify one subsystem each time you issue the command.

For each database entry in the record that is marked as backout required, the backout count in its associated database header record for this subsystem (SSID) is reduced by one. If this results in a zero backout count for this SSID, the SSID entry is removed from the database header record.

**Example of Using the DELETE.BKOUT Command**

This example uses the DELETE.BKOUT command to backout subsystem IMS3.

```plaintext
//DELBKOUT JOB
:
//SYSIN DD *
    DELETE.BKOUT SSID(IMS3)
/*
```

DELETE.CA

Use a DELETE.CA command to delete from RECON a change accumulation run record for a specified CA group.

**DELETE.CA Syntax**

```plaintext
DELETE.CA_GRPNAME(name)—RECTIME(time_stamp)
```
Parameters for DELETE.CA

GRPNAME(name)
   Required parameter you use to specify the CA group. The CA run record that is to be deleted is a member of this CA group.

RECTIME(time_stamp)
   Required parameter you use to specify the change accumulation run record that is to be deleted.

Use the RECTIME marked with an asterisk (*) from the listing of the CA record. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

Example of Deleting a Run Record
In this example, a run record is deleted from RECON for the CA group identified by the GRPNAME parameter. The RECTIME parameter identifies the record to be deleted.

//DELCA  JOB

;  
//SYSIN  DD  *
  DELETE.CA  GRPNAME(CAGRP1)  RECTIME(821220909540)
/*

DELETE.CAGRP
Use a DELETE.CAGRP command to delete a CA group record and all associated CA run records from RECON.

DELETE.CAGRP Syntax

DELETE.CAGRP—GRPNAME(name)

Parameters for DELETE.CAGRP

GRPNAME(name)
   Required parameter you use to specify the name of the CA group whose records that are to be deleted.

Example of Deleting CA Group Records
In this example, CA group records are deleted from RECON. The CA group for which the record is being deleted is identified by the GRPNAME parameter.

//DELCAGRP  JOB

;  
//SYSIN  DD  *
  DELETE.CAGRP  GRPNAME(CAGRP2)
/*

DELETE.DB
Use a DELETE.DB command to delete from RECON a database and all information related to the database that has been recorded in the RECON. If the database is a HALDB, the database and all its partitions are deleted. If the named database is a HALDB partition, the command fails. Use DELETE.PART command to delete a HALDB partition. If the database or any partition of a HALDB database is in use, the command fails and none of the RECON records are deleted.
Related Reading: See the IMS V9: Administration Guide: Database Manager for information on designing databases, especially “Procedures for Changing PHDAM, PHIDAM, and PSINDEX Partition Definitions.”

DELETE.DB Syntax

DELETE.DB—DBD(name)

Parameters for DELETE.DB

DBD(name)

Required parameter you use to identify the name of the database to be deleted.

All database, DBDS, allocation, image copy, recovery, and reorganization records that have the same database name as name are deleted. In addition, all CA group and DBDS group records are scanned in order to delete any entries for which the corresponding DBDS records have been deleted. All log allocation records are also scanned in order to delete any entries in the allocation list for which the corresponding DBDS records have been deleted.

Restriction for HALDBs: For the DELETE.DB command, name specifies the name of a HALDB master database.

Example of Deleting Records from RECON

In this example, records are deleted from RECON for the database and its corresponding DBDSs identified by the DBD parameter.

//DELBDB JOB
::
//SYSIN DD *
   DELETE.DB DBD(THISDB)
/*

DELETE.DBDS

Use a DELETE.DBDS command to delete from RECON all records that are related to a specified DBDS or DEDB area. If the DBDS for which the records are to be deleted belongs to a CA group or to a DBDS group, its name is removed from the group record. The DELETE.DBDS command fails if the DL/I database or Fast Path DEDB area is in use.

Restriction for HALDBs: You must use the HALDB Partition Definition utility to update or delete information about HALDBs in the RECON data set. The DELETE.DBDS command cannot be used to delete any DBDSs of a HALDB partition.

DELETE.DBDS Syntax

DELETE.DBDS—DBD(name)—DDN(name)—AREA(name)

Parameters for DELETE.DBDS

DBD(name)

Required parameter you use to specify the database name of the DBDS or DEDB area for which all records to be deleted.

DDN(name) | AREA(name)

Mutually exclusive, required parameters you use to specify the ddname of the DBDS or area for which all records are to be deleted from RECON.
Example of Deleting Records for the DBDS

In this example, records are deleted from RECON for the DBDS identified by the DBD and DDN parameters.

```
//DELDBDS JOB
:
//SYSIN DD *
   DELETE.DBDS DBD(DBDESDSA) DDN(DDNESDSA)
/*
```

DELETE.DBDSGRP

Use a DELETE.DBDSGRP command to delete the record of a specified DBDS group from RECON.

DELETE.DBDSGRP Syntax

```
*>DELETE.DBDSGRP—GRPNAME(name)
```

Parameters for DELETE.DBDSGRP

**GRPNAME(name)**

Required parameter you use to specify the name of the DBDS group that is being deleted. The specified name must be that of a group that is identified in RECON.

Example of Deleting a DBDS Group Record

In this example, a DBDS group record is deleted from RECON.

```
//DELODGP JOB
:
//SYSIN DD *
   DELETE.DBDSGRP GRPNAME(DBDSGRP1)
/*
```

DELETE.GSG

Use a DELETE.GSG command to delete a global service group record from the RECON. The GSG must not have any subsystem assigned to it.

Databases assigned to this GSG are reset to uncovered status as part of the processing of the DELETE.GSG command. The GSG names and log tokens of all RECON log records that are associated with this GSG are reset.

This command fails if RSRFEAT=NO is specified in the IMSCTRL macro.

DELETE.GSG Syntax

```
*>DELETE.GSG—GSGNAME(gsgname)
```

Parameters for DELETE.GSG

**GSGNAME(gsgname)**

Required parameter you use to specify the name of the global service group to be deleted.
Example of Deleting a Global Service Group Record
In this example, a global service group record is deleted from RECON.

```
//DELDGBGP JOB
:
//SYSIN DD *
DELETE.GSG GSGNAME(GSGNM1)
/*
```

DELETE.IC

Use a DELETE.IC command to delete an image copy record or the information about a second image copy data set. If you specify the ICDSN2 parameter, only the information about a second image copy data set is deleted; otherwise, both the entire image copy record and the information about the second image copy data set are deleted.

DELETE.IC Syntax

```
DELETE.IC DBD(name) DDN(name) AREA(name) RECTIME(time_stamp) ICDSN2(name)
```

Parameters for DELETE.IC

DBD(name)

Required parameter you use to identify the image copy record to be deleted. name is the database name of the DBDS or DEDB area to which it is related.

Restriction for HALDBs: For the DELETE.IC command, name specifies the name of a HALDB partition.

DDN(name) | AREA(name)

Mutually exclusive, required parameters you use to identify the image copy record to be deleted. name is the name of the DBDS or DEDB area to which it is related.

RECTIME(time_stamp)

Required parameter you use to identify the specific image copy record that is to be deleted. Use the RUN time marked with an asterisk (*) in a listing of the IMAGE record. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

ICDSN2(name)

Optional parameter you use to specify the name of a duplicate image copy data set for which information is to be deleted from an image copy record. (The record of the first image copy data set remains in RECON.)

Example of Deleting Information from an Image Copy Record

In this example, information about a duplicate image copy data set is deleted from an image copy record in RECON. The parameters DBD, AREA, ICDSN2, and RECTIME identify the information to be deleted. The asterisk (*) in the ICDSN2 parameter is to be expanded by DBRC according to the default-naming convention for image copy data sets.

```
//DELIC JOB
:
//SYSIN DD *
DELETE.IC DBD(DBDKSDS1) AREA(AREA006) -
   RECTIME(841231223221) ICDSN2(IMS.*.ICDSN5)
/*
```
DELETE.LOG (for OLDS)

DELETE.LOG (for OLDS)
Use this command to delete a data set entry from an OLDS record.

DELETE.LOG (for OLDS) Syntax

\[ \text{DELETE.LOG} \text{-} \text{OLDS(}\text{ddname}\text{)} - \text{INTERIM} - \text{LASTCLOS} - \text{SSID(}\text{name}\text{)} \]

Parameters for DELETE.LOG (for OLDS)

OLDS(\text{ddname})
Required parameter you use to specify the ddname of the primary OLDS. DBRC deletes the RECON records of the primary and secondary OLDS for the specified subsystem with the specified \text{ddname}. You can delete the record of an OLDS only if the OLDS has been archived.

INTERIM
Optional parameter you use to specify that an interim OLDS record is to be deleted.

LASTCLOS
Optional parameter you use to specify that the OLDS specified in the OLDS parameter is the last OLDS in the PRIOLDS record and should be deleted. Use this parameter with caution. Normally, the last OLDS in the PRIOLDS record is the last OLDS that was closed, and you should not delete it. Close the first OLDS in a subsequent restart if the first OLDS is empty because of an error.

SSID(\text{name})
Optional parameter you use to specify the name of the IMS subsystem that created the log data set for which the RECON record is to be deleted.

The SSID is an eight-character string of any alphanumeric characters that comprise a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. Use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not set a default in the RECON header record, you must specify SSID.

Example of Deleting an Interim OLDS Record

In this example, an interim OLDS record is deleted from RECON.

```plaintext
//DELOG JOB

//SYSIN DD *
DELETE.LOG SSID(IMSA) OLDS(DFSOLP03) - INTERIM
/*
```

DELETE.LOG (for RLDS and SLDS)

Use this command to:

- Delete the following records:
  - A PRILOG family of records
  - One data set entry from a PRILOG family
  - Some or all inactive PRILOG families
  - SLDSs produced by RSR tracking subsystems (TPSLD, TSSLD, TIPSL TISSL)
- Delete records of SLDSs created by RSR tracking IMS subsystems.

208 DBRC Guide and Reference
• Delete single data set entries from RLDS and SLDS records.
• Prevent the PRILOG records from exceeding the maximum RECON record size.

DELETE.LOG (for RLDS and SLDS) Syntax

```
>>DELETE.LOG
  INACTIVE
  STARTIME(time_stamp)
  TOTIME(time_stamp)
  OSN(name) — TRACKING–SSID(name)
  INTERIM
  TRACKING–SSID(name)

/LISTDL — NOLISTDL
```

Parameters for DELETE.LOG (for RLDS and SLDS)

<table>
<thead>
<tr>
<th>INACTIVE</th>
<th>STARTIME(time_stamp)</th>
<th>TOTIME(time_stamp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutualy exclusive, optional parameters you use to specify the records that are to be deleted.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For a log to be considered inactive, all of the following conditions must be met:

- The log does not contain any DBDS change records more recent than the oldest image copy data set that is known to DBRC (empty LOGALL record).
- The log is older than the log retention period that was specified in the INIT.RECON or CHANGE.RECON command.
- The log has either been terminated (nonzero stop time) or has the ERROR flag in the PRILOG or SECLOG record set on.

**INACTIVE**

Delete active a PRILOG and its associated log records. INACTIVE is the default.

INACTIVE does not delete PRILOG and SECLOG records for logs that do not satisfy all three conditions. Most logs satisfy the conditions. Use DELETE.LOG STARTIME(time_stamp) command for those logs that do not satisfy all three conditions.

When the DELETE.LOG INACTIVE command is issued, DBRC also examines active PRILOG records to determine if they should be compressed (meaning that any inactive data set entries in the records will be deleted). A data set entry is defined as inactive if all of the following three conditions are met:

- It is older than the log retention period that was specified in the INIT.RECON or CHANGE.RECON command.
- It is older than the earliest log volume required for recovery of any database that is registered in the RECON.
- It is older than the earliest checkpoint that is required for system restart.

See “Deleting Log Records” on page 110 and “Getting PRILOG Compression to Work” on page 108 for more information about deleting log records and compression.
DELETE.LOG (for RLDS and SLDS)

STARTIME(time_stamp)
Specifies the START time of the log records to be deleted. Use the time
time stamp with an adjacent asterisk (*) in a list of the PRILOG or SECLOG
record.

DSN(name)
Optional parameter (used with STARTIME) to specify the data set
name of a particular log data set (located at the tracking site) whose
entry in the RLDS or SLDS record is to be deleted. The specified data
set name can exist in one or more of the primary and secondary RLDS
and SLDS records. All entries having the same log sequence number
range as the specified data set are deleted.

Only tracking log data at the tracking site (the data that is a partial
copy of the log data at the active site) is eligible for deletion.

If the data set to be deleted is the last in the log record and if it is
closed, the log stop time is set to zeros to indicate that a data set gap
exists at the end of the record.

If DSN is not specified, the entire RLDS or SLDS record is deleted.

Restrictions: STARTIME is required with DSN. You cannot specify
TOTIME, INTERIM, or TRACKING with DSN.

INTERIM
Optional parameter you use to specify the interim RLDS record and
interim SLDS record to be deleted.

Restrictions: You cannot specify INTERIM if DSN is specified.
STARTIME must be specified in conjunction with INTERIM.

TOTIME
Specifies that the records of all inactive RLDSs and SLDSs that have a start
time older than the time specified with the TOTIME parameter are to be
deleted. You must specify a time that is older than the current time, minus
the log retention period. The time stamp must be in standard form (see
“Standard Time Stamp Format” on page 124). If the PRILOG is older than
TOTIME specified, but it is still active, the PRILOG will be examined and
all inactive data set entries in the record will be deleted.

Restriction: You cannot specify TOTIME if DSN is specified.

LISTDL | NOLISTDL
Mutually exclusive, optional parameters you use to specify whether the names
of the data sets deleted from the RECON are to be listed in the job output.
These parameters override the default. The default is the value specified by the
INIT.RECON or CHANGE.RECON command.

LISTDL
Specifies that the names of deleted data sets are to be listed in the job
output.

NOLISTDL
Specifies that the names of deleted data sets are not to be listed in the job
output.

SSID(name)
Required parameter (when you specify TRACKING) that specifies the
subsystem name of the RSR tracking IMS subsystem that created the SLDSs to
be deleted.
The SSID is an eight-character string of any alphanumeric characters that comprise a valid IMS subsystem identification name.

**Restriction:** The default SSID specified on the INIT.RECON or CHANGE.RECON command is not used on this form of the DELETE.LOG command.

**TRACKING**
Optional parameter you use to specify that only records of SLDSs created by RSR tracking subsystems are to be deleted. These are the TPSLD, TSSL, TIPSL, and TSSL records.

**Requirement:** This parameter must be specified with SSID and either STARTIME or TOTIME.

**Restrictions:**
- If TRACKING is specified, neither DSN nor INACTIVE can be specified.
- If INTERIM is specified, only the TIPSL and TSSL records are deleted.

**Example of Deleting the Record of an RLDS and SLDS**
In this example, the interim RLDS and interim SLDS records that are identified by the STARTIME parameter are deleted from RECON.

```
//DELLOG  JOB
::
//SYSIN  DD  *
   DELETE.LOG  STARTIME(840541212120)-
      INTERIM
/*
```

**DELETE.PART**
Use a DELETE.PART command to delete from RECON a HALDB partition and all information related to the partition that has been recorded in the RECON. If the partition, or any other partition affected by the deletion, is in use, the command fails and none of the RECON records are deleted.

**Restriction:** You cannot delete individual DBDSs belonging to HALDB partitions with the DELETE.PART command. When a DELETE.PART command is issued, it removes all the DBDSs associated with this partition as well as all of their recovery related records (for example, ALLOC, IMAGE, RECOV, and REORG records).

**Related Reading:** See the *IMS V9: Administration Guide: Database Manager* for information on designing databases, especially "Procedures for Changing PHDAM, PHIDAM, and PSINDEX Partition Definitions."

**DELETE.PART Syntax**

```
DELETE.PART—DBD(name)—PART(name)
```

**Parameters for DELETE.PART**

**DBD(name)**
Required parameter you use to identify the HALDB for which the partition is to be deleted.

**PART(name)**
Required parameter you use to identify the HALDB partition name to be deleted.
Example of Deleting a Partition from RECON
In this example, records are deleted from RECON for the partition and its corresponding DB and DBDSs identified by the DBD and NAME parameter.

```
//DELPART JOB
//SYSIN DD *
DELECPART DBD(DB3) PART(PART3)
/*
```

DELETE.RECOV
Use a DELETE.RECOV command to delete a specified recovery run record from RECON. Specifying DELETE.RECOV for the recovery run record of a time-stamp recovery implies that the DBDS or DEDB area that is related to the record has been restored. It has been restored to the state it was in just before the time-stamp recovery that created the recovery run record that you are deleting. Such a deletion also implies that no allocations of the DBDS or DEDB area took place that generated change records on IMS log data sets after the time-stamp recovery occurred.

DELETE.RECOV Syntax

```
DELETE.RECOV DBD(name) DDN(name) AREA(name) RECTIME(time_stamp)
```

Parameters for DELETE.RECOV

**DBD(name)**
Required parameter you use to identify the recovery record to be deleted; name is the database name of the related DBDS or DEDB area.

**Restriction for HALDBs:** For the DELETE.RECOV command, name specifies the name of a HALDB partition.

**DDN(name) | AREA(name)**
Mutually exclusive, required parameters you use to identify the recovery record to be deleted; name is the name of the related DBDS or DEDB area.

**RECTIME(time_stamp)**
Required parameter you use to specify the time stamp of the recovery run record to be deleted. Use the time stamp with an adjacent asterisk (*) in a list of the RECOV record. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

Example of Deleting a Recovery Record of the DBDS
This example shows the deletion from RECON of the record of a recovery of the DBDS identified by the DBD and DDN parameters. The record to be deleted is identified by the RECTIME parameter.

```
//DELRECOV JOB
:
//SYSIN DD *
DELETE.RECOV DBD(DBDSDSB) DDN(DDNDSDSB) -
  RECTIME(930891919190)
/*
```

DELETE.REORG
Use a DELETE.REORG command to delete a database reorganization record for a specified DBDS from RECON. When you specify the DELETE.REORG command, you are implying that the DBDS and the IMS DBD library have been restored to the
state they were in before the reorganization that created the database reorganization record. By using the DELETE.REORG command, you are also implying that no allocations of the reorganized database that generated records in IMS log data sets took place.

**DELETE.REORG Syntax**

```
//DELREORG DBD(name) DDN(name) RECTIME(time_stamp)
```

**Parameters for DELETE.REORG**

**DBD(name)**
- Required parameter you use to identify the reorganization record to be deleted; name is the database name of the related DBDS.
- **Restriction for HALDBs:** For the DELETE.REORG command, name specifies the name of a HALDB partition.

**DDN(name)**
- Required parameter you use to identify the reorganization record to be deleted; name is the data set ddname of the related DBDS.

**RECTIME(time_stamp)**
- Required parameter you use to identify the specific database reorganization record to be deleted. Use the time stamp with an adjacent asterisk (*) in a list of the REORG record. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

**Example of Deleting a Reorganization Record of a DBDS**

In this example, a record of the reorganization of a DBDS is deleted from RECON.

```
//DELREORG JOB
...
//SYSIN DD *
   DELETE.REORG DBD(DBDESDSB) DDN(DDNESDSB) -
      RECTIME(840231102234)
/*
```

**DELETE.SG**

Use a DELETE.SG command to delete a service group entry within a global service group record in the RECON. The service group cannot be deleted while a subsystem is signed on to the global service group.

This command fails if RSRFEAT=NO is specified in the IMSCTRL macro.

**DELETE.SG Syntax**

```
//DELSG GSGNAME(gsgname) SGNAME(sgname)
```

**Parameters for DELETE.SG**

**GSGNAME(gsgname)**
- Required parameter you use to specify the name of the global service group to which the service group belongs.

**SGNAME(sgname)**
- Required parameter you use to specify the name of the service group to be deleted.
Example of Deleting a Global Service Group Record

In this example, a service group entry within a global service group record is deleted from RECON.

```plaintext
//DELOBDDGP  JOB
:
//SYSIN  DD  *
    DELETE.SG  GSGNAME(GSGNM1)  SGNAME(SGNM1)
/*
```

DELETE.SUBSYS

Use a DELETE.SUBSYS command to delete the subsystem entry in RECON after it is verified that the specified subsystem is not authorized to use any database.

For more information about deleting subsystem entries, see "CHANGE.SUBSYS" on page 198.

To close the subsystem log, issue the NOTIFY.PRILOG command, and then issue the DELETE.SUBSYS command.

DELETE.SUBSYS Syntax

```
DELETE.SUBSYS SSID(name)
```

Parameters for DELETE.SUBSYS

SSID(name)

- Required parameter you use to identify the subsystem for which the entry is deleted from RECON if no database is authorized by the subsystem.
- When you issue this command online, the IMS control region under which the command was issued cannot be the subsystem being deleted.

Example of Deleting a Specified SUBSYS Record

In this example, the specified SUBSYS record is deleted if no database is authorized by the subsystem.

```plaintext
//DELSBSYS  JOB
:
:
//SYSIN  DD  *
    DELETE.SUBSYS  SSID(IMS34)
/*
```

DELETE.UIC

Use a DELETE.UIC command to delete the record of a nonstandard image copy data set from RECON.

DELETE.UIC Syntax

```
DELETE.UIC DBD(name) DDN(name) RECTIME(time_stamp) AREA(name)
```

Parameters for DELETE.UIC

DBD(name)

- Required parameter you use to identify the nonstandard image copy record to be deleted; name is the database name of the related DBDS or area.
Restriction for HALDBs: For the DELETE.UIC command, name specifies the name of a HALDB partition.

\texttt{DDN(name) | AREA(name)}

Mutually exclusive, required parameters you use to identify the nonstandard image copy record to be deleted; name is the name of the related DBDS or DEDB area.

\texttt{RECTIME(time_stamp)}

Required parameter you use to specify the time stamp of the nonstandard image copy record to be deleted. Use the time stamp with an adjacent asterisk (*) in a list of the IMAGE record. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

Example of Deleting a Nonstandard Image Copy Data Set Record

This example shows the deletion of a record of a nonstandard image copy data set from RECON.

```
//DELUIC JOB

//SYSIN DD *
DELETE.UIC DBD(DBDESDSB) AREA(AREAESD2) -
   RECTIME(840871212120)
/*
```
Chapter 11. GENJCL Commands

Use the GENJCL commands to generate JCL and utility control statements that run the various IMS recovery utilities.

GENJCL Commands In This Chapter

The following is a list of all the GENJCL commands:

- “GENJCL.ARCHIVE” on page 217
- “GENJCL.CA” on page 221
- “GENJCL.CLOSE” on page 225
- “GENJCL.IC” on page 229
- “GENJCL.OIC” on page 238
- “GENJCL.RECEIVE” on page 243
- “GENJCL.RECOV” on page 247
- “GENJCL.USER” on page 253

GENJCL.ARCHIVE

Use the GENJCL.ARCHIVE command to generate the JCL and utility control statements that run the Log Archive utility. For the IBM-supplied skeletal JCL execution member that is used by GENJCL.ARCHIVE, see “Generating JCL and User-Defined Output” on page 83.

GENJCL.ARCHIVE Syntax

Parameters for GENJCL.ARCHIVE

ALL | OLDS(ddname) | SLDS

Mutually exclusive, optional parameters that you use to specify which OLDS is being archived or to request the archive of tracking SLDSs.
Attention: Ensure that the RSR tracking IMS subsystem has completed the processing of the SLDS before you run a batch archive. If a tracking IMS process (such as online forward recovery (OFR), log truncation, or catch up) needs to read from an SLDS that is being processed by batch archive, allocation of the SLDS by the tracking IMS fails, and the tracking IMS might terminate abnormally.

ALL
Generates JCL to archive all OLDSs that have not been archived. A multiple-step job can be produced if either of the following conditions exist:
- The specified subsystem has non-contiguous OLDSs.
- A force-EOV condition occurred after you entered /DBRECOVERY.

OLDS
Specifies the ddname of the primary OLDS you are archiving.

SLDS
Generates JCL to archive all tracking SLDSs which are associated with the specified subsystem that have not been archived. A multiple-step job can be produced if the PRISLD or SECSLD (or both) have non-contiguous data set entries that need to be archived, or if they have more unarchived DSNs than the specified MAXOLDS value.

TOTIME(time_stamp)
Specifies that only tracking log data sets with start times older than or equal to time_stamp are to be archived. This parameter is optional and is valid only when SLDS is also specified. Otherwise it is ignored. The time stamp must be in standard format.

DEFAULTS(member)
Optional parameter you use to specify up to 10 skeletal JCL default members to be used when generating JCL. Default members are searched to resolve keywords in the order in which the members are specified on this parameter.

If a keyword is assigned a value in both the DEFAULTS and USERKEYS parameters, the value specified in USERKEYS is used.

JCLOUT(JCLOUT | dname)
Optional parameter you use to specify the output data set for the generated JCL. The data set is specified by dname. A JCL DD statement with this dname must be included in the job step containing the GENJCL command. The specified data set can be a member of a partitioned data set (PDS) as long as it is not the same data set used for the default JCLOUT.

JCLPDS(JCLPDS | dname)
Optional parameter you use to specify the skeletal JCL data set that is to be used for input when generating JCL. The data set that is specified by dname. A JCL DD statement with this dname must be included in the job step containing the GENJCL command.

JOB | JOB(member) | NOJOB
Mutually exclusive, optional parameters that you use to specify whether to produce the job statement in the generated JCL.

JOB
 Specifies that the job statement is to be produced. When JOB is specified without a member name, the IBM-supplied execution member JOBJCL produces the job statement. When JOB(member) is specified, the specified execution member produces the job statement.
NOJOB
   Specifies that the job statement is not produced in the generated JCL.

LIST | NOLIST
   Mutually exclusive, optional parameters that you use to specify whether to print
   the generated JCL using the SYSPRINT data set.
   
   LIST
      Prints the generated JCL.
   
   NOLIST
      Suppresses printing of the generated JCL.

MAXOLDS(n)
   Optional parameter you use to specify the maximum number of OLDSs or
   SLDSs to be archived in a single job. n can be any decimal number from 1 to
   100.
   
   If MAXOLDS is specified and more OLDSs need archiving than are specified
   in n, multiple jobs are generated. Each generated job archives no more than n
   OLDSs.
   
   This parameter functions somewhat differently for SLDSs than for OLDSs. If
   MAXOLDS is specified and more SLDSs need archiving than are specified in n,
   multiple job steps are generated. Each generated job step archives no more
   than n SLDSs.
   
   MAXOLDS applies only to the primary data sets. If dual logging is in effect,
   each job can have DD statements for the secondary and primary data sets (that
   is, DD statements for 2 x n data sets).
   
   If you do not specify MAXOLDS, a single job is generated for all OLDSs or
   SLDSs.

MEMBER(member)
   Optional parameter you use to specify the name of the skeletal JCL execution
   member to be used. If this parameter is not specified, the IBM-supplied
   execution member for the GENJCL.ARCHIVE command is used. For a description
   of this execution member, see "Generating JCL and User-Defined Output" on
   page 83.

SSID(name)
   Optional parameter you use to specify the name of the IMS subsystem that
   created the OLDSs or SLDSs that are to be archived.
   
   The SSID is an eight-character string of any alphanumeric characters that
   comprise a valid IMS subsystem identification name. If you do not specify
   SSID, DBRC uses the default subsystem identifier in the RECON header
   record. Use the INIT.RECON or CHANGE.RECON command to set the default system
   identifier in the RECON header record. If you have not set a default in the
   RECON header record, you must specify SSID.

TIMEFMT(sublist)
   Optional parameter you use to define the form in which time stamps appear in
   user-defined output. This specification overrides both the GENJCL default values
   and any values set on %SET statements in the input skeletal JCL.
   
   The default for the GENJCL output time format is compressed, with a two-digit
   year, and the offset in numeric form: 960021315001 +0700. If you want the
   output time stamps to appear without offsets, for example, you can override
   the default with TIMEFMT(N).
   
   The override is good only for the duration of a single GENJCL command.
Related Reading:
- The TIMEFMT parameter sublist format is described in "TIMEFMT Parameter" on page 125.
- See "TIMEFMT Subparameter Order of Precedence" on page 127 for information on the precedence of the subparameters.
- See "Standard Time Stamp Format" on page 124 for examples of the different output forms.

USERKEYS(%key1,'value' | %key2)
Optional parameter you use to set the value of keywords you have defined. Up to 32 keywords can be specified.

%key1
User-defined keyword being assigned a value. The maximum length of the keyword is eight characters, including the percent sign. The first character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

'value'
Value assigned to the user-defined keyword when it is encountered. value can be any character string enclosed in single quotes. The maximum length of value is 132 characters (excluding the quotes). If value contains a quote, use two single quotes. value can be a null string (''). If value is a time stamp, it can be zero.

%key2
Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.

Any keyword can be assigned a value with the USERKEYS parameter. However, if you assign a value to DBRC-defined keywords, the value is ignored. DBRC-defined keywords are shown in "Symbolic Keywords" on page 86.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS parameter, the value specified in USERKEYS is used.

Examples of Running the Log Archive Utility
Here are some examples of using the GENJCL.ARCHIVE command.

Example with Primary OLDS Defined by the OLDS Parameter: In this example, a GENJCL.ARCHIVE command generates the JCL and control statements required to run the Log Archive utility for the primary OLDSs that are defined by the OLDS parameter. When this command is issued, the PRIOLDS record in RECON is updated to indicate that an archive has been scheduled for the OLDS. Default skeletal member ARCHJCL is taken from the data set that is identified in the JCLPDS DD statement. The generated JCL goes to the data set identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

```
/*GENJAR  JOB   */
//JCLOUT DD . . .
//JCLPDS DD . . .
::
//SYSIN DD *
   GENJCL.ARCHIVE SSID(IMSA) -
      OLDS(DFSOLP01,DFSOLP02)
/*
```
As part of the archive process, the PRIOLDS record in RECON is updated to indicate that the archive has completed. RECON is updated with the PRISLD and SECSLD records that identify the created SLDSs. In addition, RECON is updated with the PRILOG and SECLOG records that identify the created RLDSs.

Example of the SSID IMSB OLDS Parameter Defining the Primary OLDS: In this example, a GENJCL.ARCHIVE command generates JCL to archive the primary OLDS that is defined in the OLDS parameter for SSID IMSB. JCL execution member ARCHJCLA is taken from the JCLPDS data set that is identified in the PDSJCL DD statement. The generated JCL goes to SYSOUT=A, which is identified in the OUTJCL DD statement. Skeletal member JOBJCL produces a job statement.

```plaintext
//GENJAR1 JOB
//OUTJCL DD SYSOUT=A
//PDSJCL DD DSN=dsname
//SYSIN DD *
GENJCL.ARCHIVE SSID(IMSB) OLDS(DFSOLP01) MEMBER(ARCHJCLA) -
JCLPDS(PDSJCL) JCLOUT(OUTJCL)
```

Example for Unarchived Default Subsystem OLDS: In this example, the GENJCL.ARCHIVE command generates JCL and control statements to archive all unarchived OLDSs for the default subsystem ID.

JCL execution member ARCHJCLB is taken from the JCLPDS data set that is identified by the JCLPDS DD statement. Member DEFARC01 from the JCLPDS data set (identified in the JCLPDS DD statement) contains values to resolve user-defined keywords in ARCHJCLB. %SSPACE is a user-defined keyword in member ARCHJCLB which is assigned a value of ‘CYL,1’. %RSPACE is a user-defined keyword in member ARCHJCLB, which is assigned a value of ‘TRK,4’.

The values specified in the USERKEYS parameter for a keyword overrides the values found in the DEFAULTS member. JOB1 is a member in the JCLPDS that produces a job statement.

```plaintext
//GENJAR2 JOB
//JCLPDS DD . .
//JCLOUT DD . .
//SYSIN DD *
GENJCL.ARCHIVE MEMBER(ARCHJCLB) DEFAULTS(DEFARC01) -
USERKEYS(('%SSPACE,'CYL,1'),('%RSPACE,'TRK,4')) JOB(JOB1)
```

**GENJCL.CA**

Use the GENJCL.CA command to generate the JCL and utility control statements to run the Change Accumulation utility for a specified CA group. For information on the IBM-supplied skeletal JCL execution member used by GENJCL.CA, see “Generating JCL and User-Defined Output” on page 83.

**GENJCL.CA Syntax**
Parameters for GENJCL.CA

GRPNAME(name)
Required parameter you use to specify the name of the CA group for which you are running the Change Accumulation utility.

DEFAULTS (member)
Optional parameter you use to specify the names of up to 10 skeletal JCL default members that are used when generating JCL. Default members are searched to resolve keywords in the order in which the members are specified on this parameter.

If a keyword is assigned a value in both the DEFAULTS and USERKEYS parameters, the value specified in USERKEYS is used.

JCLOUT (JCLOUT | ddname)
Optional parameter you use to specify the output data set for the generated JCL. The data set is specified by ddname. A JCL DD statement with this ddname must be included in the job step containing the GENJCL command. The specified data set can be a member of a partitioned data set as, but only if it is not the same data set that is used for the default, JCLOUT.

JCLPDS(JCLPDS | ddname)
Optional parameter you use to specify the skeletal JCL data set that is to be used for input when generating JCL. The data set is specified by ddname. A JCL DD statement with this ddname must be included in the job step containing the GENJCL command.

JOB | JOB(member) | NOJOB
Mutually exclusive, optional parameters that you use to specify whether to produce the first job statement in the generated JCL.

JOB
Specifies that the job statement is to be produced. When JOB is specified without a member name, the IBM-supplied execution member JOBJCL produces the job statement.

JOB(member)
Specified execution member produces the job statement.

NOJOB
Specifies that the job statement is not produced in the generated JCL.

LIST | NOLIST
Mutually exclusive, optional parameters that you use to specify whether you want the generated JCL to be written to the SYSPRINT data set.
LIST
Prints the generated JCL.

NOLIST
Suppresses the printing of the generated JCL.

MEMBER(member)
Optional parameter you use to specify the name of the skeletal JCL execution member that is to be used. If this parameter is not specified, the default specified for the CA group is used. For a description of the IBM-supplied execution member, see “Generating JCL and User-Defined Output” on page 83.

NODEFLT
Optional parameter you use to specify that the implicit skeletal JCL default member, if any, for the CA group is not to be used.

TIMEFMT(sublist)
Optional parameter you use to define the form in which time stamps appear in user-defined output. This specification overrides both the GENJCL default values and any values set on %SET statements in the input skeletal JCL.

The default for the GENJCL output time format is compressed, with a two-digit year, and the offset in numeric form: 960021315001 +0700. If you want the output time stamps to appear without offsets, for example, you can override the default with TIMEFMT,(N).

The override is good only for the duration of a single GENJCL command.

Related Reading:
- The TIMEFMT parameter sublist format is described in “TIMEFMT Parameter” on page 125.
- See “TIMEFMT Subparameter Order of Precedence” on page 127 for information on the precedence of the subparameters.
- See “Standard Time Stamp Format” on page 124 for examples of the different output forms.

UNIT(3400  |  unittype)
Optional parameter you use to specify the unit type of the output change accumulation data set. This parameter is valid only when both of the following conditions are true:
- The VOLLIST parameter is specified.
- The CA group for which the JCL is being generated is defined with the NOREUSE parameter.

USERKEYS(%key1,’value’  |  %key2)
Optional parameter you use to set the value of keywords you have defined.
Up to 32 keywords can be specified.

%key1
User-defined keyword being assigned a value. The maximum length of the keyword is eight characters, including the percent sign. The first character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

‘value’
Value assigned to the user-defined keyword when it is encountered. value can be any character string enclosed in single quotes. The maximum length of value is 132 characters (excluding the quotes). If value contains a quote, use two single quotes. value can be a null string (“”). If value is a time stamp, it can be zero.
%key2
Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.

Any keyword can be assigned a value with the USERKEYS parameter. However, if you assign a value to DBRC-defined keywords, the value is ignored. DBRC-defined keywords are shown in "Symbolic Keywords" on page 86.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS parameter, the value specified in USERKEYS is used.

VOLLIST(volser)
Optional parameter you use to specify the volumes that are to contain the change accumulation data set. This parameter is valid only if the CA group for which the JCL is being generated was defined with the NOREUSE parameter.

VOLNUM(value) | CATIME(time)
Mutually exclusive, optional parameters that you use to specify the log volumes that are used by the Change Accumulation utility.

VOLNUM
Specifies the number of log volumes that are to be used in each change accumulation job step. DBRC generates a multiple-step job that invokes the Change Accumulation utility in each step (unless you specify VOLLIST), and limits the number of log volumes in each step to the specified number. If another volume is needed to complete subset processing, VOLNUM may be overridden by DBRC. DBRC may also override VOLNUM for the following reasons:
• CATDS is specified and a data set entry spans multiple volumes.
• Log volumes have identical start times.
• Log volumes have identical start times and stop times.

For value, specify the number of log volumes. You can specify a decimal number from 1 to 255.

Each job step except the first one uses the change accumulation data set (that was generated in the previous step) as the beginning point of the accumulation in that step.

CATIME
Specifies the time after which no log volumes for the specified CA group are to be included. The time stamp does not need to be the stop time of any log volume. DBRC uses the time stamp as the ending delimiter for the log volume subset. Therefore, all log volumes that have start times less than or equal to the specified time stamp are included in the subset of volumes. The time stamp must be in standard form (see "Standard Time Stamp Format" on page 124).

Examples of Running the Change Accumulation Utility
Here are some examples of using the GENJCL.CA command.

Example for the GRPNAME CA Group: In this example, a GENJCL.CA command generates the JCL and control statements required to run the Change Accumulation utility for the CA group identified in the GRPNAME parameter. CAGRP1 is defined as REUSE. If the INIT.CAGRP command for CAGRP1 is specified without a CAJCL(member) parameter, default skeletal member CAJCL from the data set identified in the JCLPDS DD statement is used. If INIT.CAGRP is specified with the
CAJCL(member) parameter, that member is used. The generated JCL goes to the data set that is identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

```
//GENJCA JOB
//JCLPDS DD . . .
//JCLOUT DD . . .
;
//SYSIN DD *
  GENJCL.CA GRPNAME(CAGRP1)
/*

Example of CAJCL.A Generated Skeletal JCL: In this example, the GENJCL.CA command is generated with skeletal JCL execution member CAJCLA, which was taken from the JCLPDS data set identified by the PDSJCL DD statement. Output from the generated JCL goes to SYSOUT=A, identified in the OUTJCL DD statement. CAGRP2 is defined with the NOREUSE parameter. Skeletal member JOBJCL produces a job statement.

```
//GENJCA1 JOB
//OUTJCL DD SYSOUT=A
//PDSJCL DD DSN=dsname
//SYSIN DD *
  GENJCL.CA GRPNAME(CAGRP2) VOLLIST(VOL001) MEMBER(CAJCLA) -
  JCLPDS(PDSJCL) JCLOUT(OUTJCL)
/*

Example of CAJCL.B Generated Skeletal JCL: In this example, the GENJCL.CA command generates JCL and control statements to run the Change Accumulation utility for CAGRP3, which is defined as REUSE. JCL execution member CAJCLB is taken from the JCLPDS data set identified by the JCLPDS DD statement.

DEFAULTS(DEFCA01) is a member in the JCLPDS data set which contains values to resolve user defined keywords in member CAJCLB. The default member for the CAGR3, if initialized in the INIT.CAGRP.DEFLTJCL(MEMBER) command, is also used to resolve keywords. %DISP is a user-defined keyword in member CAJCLB which is assigned a value of 'SHR'. %OUTCLS is a user-defined keyword in member CAJCLB which is assigned a value of 'B'.

The values in the explicitly defined DEFAULTS member overrides values in the predefined DEFLTJCL member. The values specified in the USERKEYS parameter for a keyword overrides the values found in the DEFAULTS member. JCL is generated with no job statement. All volumes that have stop times less than or equal to the specified time stamp are included in the subset of volumes that is used as input to the Change Accumulation utility. Generated JCL is listed.

```
//GENJCA3 JOB
//JCLPDS DD
//JCLOUT DD
//SYSIN DD *
  GENJCL.CA GRPNAME(CAGRP3) MEMBER(CAJCLB) DEFAULTS(DEFCA01) -
  USERKEYS(('%DISP','SHR'),('%OUTCLS','B')) NOJOB LIST -
  CATIME(861020202111)
/*

GENJCL.CLOSE

Use the GENJCL.CLOSE command to generate the JCL and utility control statements that run the Log Recovery utility to close an OLDS using the WADS. For information about the IBM-supplied skeletal JCL execution member used by GENJCL.CLOSE, see "Generating JCL and User-Defined Output" on page 83
GENJCL.CLOSE

**GENJCL.CLOSE Syntax**

```
GENJCL.CLOSE
  DEFAULTS(member)
  JCLPDS(ddname)
  OLDS(ddname)
  SSID(name)
  USERKEYS(\%key1, 'value')
  JOB
  NOLIST
  List
  MEMBER(member)
  JCLPDS(ddname)
  OLDS(ddname)
  SSID(name)

GENJCL.CLOSE
```

**Parameters for GENJCL.CLOSE**

**DEFAULTS(*member*)**
Optional parameter you use to specify up to 10 names of skeletal JCL default members used when generating JCL. Default members are searched to resolve keywords in the order in which the members are specified on this parameter.

If a keyword is assigned a value in both the DEFAULTS and USERKEYS parameters, the value specified in USERKEYS is used.

**JCLPDS(*ddname*)**
Optional parameter you use to specify the skeletal JCL data set to be used for input when generating JCL. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the GENJCL command. The specified data set can be a member of a partitioned data set, but only if it is not the same data set that is used for the default, JCLPDS.

**OLDS(*ddname*)**

**SSID(*name*)**

**USERKEYS(*key1*, 'value')**

**JOB | JOB(*member*) | NOJOB**
Mutually exclusive, optional parameters that you use to specify whether to produce the job statement in the generated JCL.

**JOB**
Specifies that the job statement is produced. When JOB is specified without a member name, the IBM-supplied execution member JOBJCL produces the job statement.

**JOB(*member*)**
Specified execution member produces the job statement.

**NOJOB**
Specifies that no job statement is to be produced in the generated JCL.
LIST | NOLIST
Mutually exclusive, optional parameters that you use to specify whether you want the generated JCL to be written to the SYSPRINT data set.

LIST
Prints the generated JCL.

NOLIST
Suppresses the printing of the generated JCL.

MEMBER(member)
Optional parameter you use to specify the name of the skeletal JCL execution member to be used. If this parameter is not specified, the IBM-supplied execution member for the GENJCL.CLOSE command is used. For a description of this execution member, see “Generating JCL and User-Defined Output” on page 83.

OLDS(ddname)
Optional parameter you use to specify which OLDS is to be closed. You specify the name of the DD statement that was used when the online IMS subsystem created the log data. The ddname of the primary OLDS must be specified. If you do not specify the OLDS, DBRC generates JCL to close the OLDS that was most recently opened.

SSID(name)
Optional parameter you use to specify the name of the IMS subsystem that created the OLDSs being closed.

The SSID is an eight-character alphanumeric string that comprises a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. Use the INIT.RECON or CHANGE.RECON command to set the default system identifier in the RECON header record. If you have not set a default in the RECON header record, you must specify SSID.

TIMEFMT(sublist)
Optional parameter you use to define the form in which time stamps appear in user-defined output. This specification overrides both the GENJCL default values and any values set on %SET statements in the input skeletal JCL.

The default for the GENJCL output time format is compressed, with a two-digit year, and the offset in numeric form: 960021315001 +0700. If you want the output time stamps to appear without offsets, for example, you can override the default with TIMEFMT(N).

The override is good only for the duration of a single GENJCL command.

Related Reading:
• The TIMEFMT parameter sublist format is described in “TIMEFMT Parameter” on page 125.
• See “TIMEFMT Subparameter Order of Precedence” on page 127 for information on the precedence of the subparameters.
• See “Standard Time Stamp Format” on page 124 for examples of the different output forms.

USERKEYS(%key1,'value' 1 %key2)
Optional parameter you use to set the value of keywords you have defined. Up to 32 keywords can be specified.

%key1
User-defined keyword that is being assigned a value. The maximum length
GENJCL.CLOSE

of the keyword is eight characters, including the percent sign. The first character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

'value'
Value assigned to the user-defined keyword when it is encountered. value can be any character string enclosed in single quotes. The maximum length of value is 132 characters (excluding the quotes). If value contains a quote, use two single quotes. value can be a null string ("'). If value is a timestamp, it can be zero.

%key2
Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.

Any keyword can be assigned a value with the USERKEYS parameter. However, if you assign a value to DBRC-defined keywords, the value is ignored. DBRC-defined keywords are shown in "Symbolic Keywords" on page 86.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS parameter, the value specified in USERKEYS is used.

Examples of Running the Log Recovery Utility
Here are some examples of using the GENJCL.CLOSE command.

Example When a Host Operating System Failed and /ERE Is Not Possible: In this example, a GENJCL.CLOSE command generates the JCL and control statements that are required to run the Log Recovery utility for the IMS online subsystem with subsystem ID IMSA, which was using a primary OLDS when a host operating system failed and /ERE could not be performed. Default skeletal member LOGCLJCL is taken from the data set identified in the JCLPDS DD statement. Output from the generated JCL goes to the data set identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

//GENJCL JOB
//JCLOUT DD
//JCLPDS DD
:
//SYSIN DD *
GENJCL.CLOSE SSID(IMSA)
/*

After the close job runs, the PRIOLDS record in RECON that corresponds to the OLDS is updated to indicate the successful close.

Example Using the CLOSE1 JCLPDS Member: In this example, the GENJCL.CLOSE command is generated with a skeletal JCL execution member CLOSE1, which is taken from the JCLPDS data set identified in the PDS DD statement. Output from the generated JCL goes to the data set identified in the OUT DD statement. MEMBER DEFCL1 from the JCLPDS DD statement contains values to resolve user-defined keywords in member CLOSE1. Skeletal member JOBJCL produces a job statement. Generated JCL is listed.

//GENJCL1 JOB
//OUT DD . .
//PDS DD . .
//SYSIN DD *
GENJCL.CLOSE MEMBER(CLOSE1) OLDS(DFSOLP01) -
JCLPDS(PDS) JCLOUT(OUT) DEFAULTS(DEFCL1) LIST
GENJCL.IC

Use the GENJCL.IC command to generate the JCL and utility control statements needed to run the Database Image Copy utility or the Database Image Copy 2 utility. For information about the IBM-supplied skeletal JCL execution member used by GENJCL.IC, see “Generating JCL and User-Defined Output” on page 83.

Important:

- For HALDB partitions, the GENJCL.IC command treats ILDS and index data sets differently than data DBDSs. The GENJCL.IC command skips these data sets in groups, regardless of whether the groups are explicit or implicit. If you explicitly specify one of these data sets, the GENJCL.IC command fails.
- For HALDB DBDSs, the GENJCL.IC command does not generate DD statements for the DBDSs to be copied. The SYSIN control statement identifies which active DBDSs to copy. These can be either the A-through-J data set or the M-through-V data sets.


GENJCL.IC Syntax

\[
\text{GENJCL.IC } \text{DBD}((\text{name})) \text{ GROUP}((\text{name})) \text{ COPIES}((1)) \text{ DDN}((\text{ddname})) \text{ AREA}((\text{name})) \text{}\
\]

CIC

\[
\text{NOCIC} \text{ SMSCIC} (\text{Sameds}) (\text{Compress}) (n) (\text{Sameds,Compress}) (\text{Sameds,Compress},n) (\text{Compress},n) \text{}\
\]

SMSNOCIC

\[
\text{DBREL}(L) (\text{Sameds}) (\text{Compress}) (n) (\text{Sameds,Compress}) (\text{Sameds,Compress},n) (\text{Compress},n) \text{}\
\]

\[
\text{DEFAULTS}((\text{member})) \text{}\
\]

\[
\text{DBREL}(P) \text{}\
\]

\[
\text{JCLPDS} \text{}\
\]

\[
\text{JCLOUT}((\text{ddname})) \text{}\
\]

\[
\text{JCLPDS}((\text{ddname})) \text{}\
\]

\[
\text{JOB}((\text{member})) \text{}\
\]

\[
\text{JOB}((\text{member})) \text{}\
\]

\[
\text{LIST} \text{}\
\]

\[
\text{MULTIJOB} \text{}\
\]

\[
\text{NODEFLT} \text{}\
\]

\[
\text{TIMEFMT}((\text{sublist})) \text{}\
\]

\[
\text{NOLIST} \text{}\
\]

\[
\text{NOJOB} \text{}\
\]
Parameters for GENJCL.IC

**DBD(name)** | **GROUP(name)**
---|---
Mutually exclusive, required parameters that you use to specify the database or DBDS group for which DBDSs are to be copied.

**DBD**
Specifies the name of the database that contains the DBDS or area to be copied.

If DDN is omitted and ONEJOB is specified for a SMSCIC or SMSNOCIC image copy, a utility control statement specifying the DBD name as the group name is generated in the JCL.

**Restriction for HALDBs:** For the GENJCL.IC command, *name* specifies the name of either a HALDB master database name or a HALDB partition name.

**GROUP**
Specifies that all DBDSs of a DBDS group are to be copied. If GROUP is specified, the GENJCL.IC command executes repeatedly for each DBDS of the named DBDS group.

If GROUP and ONEJOB are specified for an SMSCIC or SMSNOCIC image copy, a utility control statement specifying the group name is generated in the JCL.

**COPIES(1 | 2 | 3 | 4)**
Optional parameter you use to specify how many two image copy data sets are to be produced for the specified DBDS.

If the specified DBDS is identified in RECON with the NOREUSE attribute, you can specify the COPIES parameter if you want two or more image copy data sets; otherwise, one image copy data set is produced. COPIES(3 | 4) can only be specified if either SMSCIC or SMSNOCIC is also specified. The third and fourth copies are not recorded in RECON.
If the specified DBDS is identified in RECON with the REUSE attribute, the COPIES parameter is not allowed. The number of copies is determined by the number of pre-defined image copy data sets specified on the INIT.IC command.

**DDN**(name) | **AREA**(name)

Mutually exclusive, optional parameters that you use to identify the DBDS ddname or DEDB area name that is to be copied.

Specify the DDN or AREA parameter only if you specify the DBD parameter. If you do not specify DDN or AREA, the GENJCL.IC command executes repeatedly, once for each DBDS or area of the specified database. If you specify a HALDB master name, the GENJCL.IC command is performed for all data DBDSs for each partition in the HALDB master database. If you specify a HALDB partition name, the GENJCL.IC command is performed for all data DBDSs of the identified partition.

For HALDBs, you must specify a partition database name with the DBD parameter in order to use the DDN parameter. The DDN parameter value is the partition DDN. The GENJCL.IC command is performed for the identified DBDS of the partition. The GENJCL.IC command fails if DDN does not identify a data DBDS in the partition.

**CIC | NOCIC | SMSCIC(Sameds) | Compress | n | any combination |**

**SMSNOCIC(Sameds) | Compress | n | any combination | DBREL(L | P)**

Optional parameters that you use to indicate how the image copy is to be taken.

**Restriction:** CIC and SMSCIC cannot be used for nonrecoverable or user-recoverable databases.

**CIC**

Specifies that the Database Image Copy (DFSUDMP0) utility is to be used to take an image copy. If CIC is specified, the copy is concurrent with update processing.

**NOCIC**

Specifies that the Database Image Copy (DFSUDMP0) utility is to be used to take an image copy. If NOCIC is specified, the copy is made while the database is unavailable for update processing.

**SMSCIC(Sameds) | Compress | n | any combination**

Indicates that the Database Image Copy 2 utility is to be used to take an image copy concurrent with update processing. The Database Image Copy 2 utility invokes DFSMSdss to dump the database data set or sets using the Concurrent Copy option.

**Sameds**

An optional parameter you use to specify that all the image copies generated by the job are to be written to a single output data set. This option is not valid if any of the DBDs or areas have the REUSE attribute, if ONEJOB is not specified, or if more than 255 data sets and/or areas are to be copied. You can abbreviate this option when specifying it. When a DBDS group (explicit or implicit) is specified, this option applies to all members of the group.

**Compress**

An optional parameter you use to indicate that the utility is to specify the DFSMSdss compress option. The compress option enables you to reduce the storage space required to hold the image copy; however, using the compress option increases the CPU time that is required to
perform the copy operation. You can abbreviate this option when
specifying it. When a DBDS group (explicit or implicit) is specified, this
option applies to all members of the group.

\( n \) An optional parameter you use to specify the value to be used for the
DFSMSt parameter. \( n \) is a number from 1 to 4 where:

- If \( n \) is 1, DFSMSt reads one track at a time.
- If \( n \) is 2, DFSMSt reads two tracks at a time.
- If \( n \) is 3, DFSMSt reads five tracks at a time.
- If \( n \) is 4, DFSMSt reads one cylinder at a time.

If \( n \) is not specified, the default is 1 for SMSCIC and 4 for SMSNOCIC.
Note that higher values use more real and virtual storage, and keeps
the channel busy for longer blocks at a time. You can abbreviate this
option when specifying it. When a DBDS group (explicit or implicit) is
specified, this option applies to all members of the group.

**SMSNOCIC(Sameds | Compress | P)**

Indicates that the Database Image Copy 2 utility is to be used to take an
image copy while the database is unavailable for update processing. The
Database Image Copy 2 utility invokes DFSMSt to dump the database
data set(s) using the Concurrent Copy option.

**Sameds**

An optional parameter you use to specify that all the image copies
generated by the job are to be written to a single output data set. This
option is not valid if any of the DBDs or areas have the REUSE
attribute, if ONEJOB is not specified, or if more than 255 data sets
and/or areas are to be copied. You can abbreviate this option when
specifying it. When a DBDS group (explicit or implicit) is specified, this
option applies to all members of the group.

**Compress**

An optional parameter you use to indicate that the utility is to specify
the DFSMSt compress option. The compress option enables you to
reduce the storage space required to hold the image copy; however,
using the compress option increases the CPU time that is required to
perform the copy operation. You can abbreviate this option when
specifying it. When a DBDS group (explicit or implicit) is specified, this
option applies to all members of the group.

\( n \) An optional parameter you use to specify the value to be used for the
DFSMSt parameter. \( n \) is a number from 1 to 4 where:

- If \( n \) is 1, DFSMSt reads one track at a time.
- If \( n \) is 2, DFSMSt reads two tracks at a time.
- If \( n \) is 3, DFSMSt reads five tracks at a time.
- If \( n \) is 4, DFSMSt reads one cylinder at a time.

If \( n \) is not specified, the default is 1 for SMSCIC and 4 for SMSNOCIC.
Note that higher values use more real and virtual storage, and keeps
the channel busy for longer blocks at a time. You can abbreviate this
option when specifying it. When a DBDS group (explicit or implicit) is
specified, this option applies to all members of the group.

**DBREL(L | P)**

Indicates when the database is to be made available for update
processing. DBREL is applicable only when SMSNOCIC is also
specified. IMS ignores DBREL when it is specified with SMSCIC.
L L indicates that updates are to be allowed after the image copy is logically complete (after DFSMS has initialized a concurrent copy session). Update processing can occur (or be resumed) before the image copy is physically complete.

P P indicates that updates are not to be allowed until the image copy is physically complete.

**DEFAULTS(member)**
Optional parameter you use to specify up to 10 names of skeletal JCL default members to be used when generating JCL. Default members are searched to resolve keywords in the order in which the members are specified on this parameter.

If a keyword is assigned a value in both the DEFAULTS and USERKEYS parameter, the value specified in USERKEYS is used.

**JCLOUT(JCLOUT | ddname)**
Optional parameter you use to specify the output data set for the generated JCL. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the `GENJCL` command. The specified data set can be a member of a partitioned data set, but only if it is not the same data set that is used for the default, JCLOUT.

**JCLPDS(JCLPDS | ddname)**
Optional parameter you use to specify the skeletal JCL data set that is to be used for input when generating JCL. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the `GENJCL` command.

**JOB | JOB(member) | NOJOB**
Mutually exclusive, optional parameters that you use to specify whether to produce the job statement in the generated JCL.

- **JOB** Specifies that the job statement is to be produced. When JOB is specified without a member name, the IBM-supplied execution member JOBJCL produces the job statement. When JOB(*member) is specified, the specified execution member produces the job statement.

- **NOJOB** Specifies that the job statement is not to be produced in the generated JCL.

**LIST | NOLIST**
Mutually exclusive, optional parameters that you use to specify whether to write the generated JCL to the SYSPRINT data set.

- **LIST** Prints the generated JCL.

- **NOLIST** Suppresses the printing of the generated JCL.

**MEMBER(member)**
Optional parameter you use to specify the name of the skeletal JCL execution member to be used. For a description of the IBM-supplied execution member, see "Generating JCL and User-Defined Output" on page 83. If this parameter is not specified, the default specified on the INIT.DBDS command is used.
MULTIJOB | ONEJOB
Mutually exclusive, optional parameters that you use to control how many JOB and EXEC statements are to be generated when a DBDS group is specified either explicitly or implicitly.

MULTIJOB
Produces one job for each group member.

ONEJOB
Produces one job with:
- A separate step for each group member with the CIC or NOCIC option.
- A single step to copy all members of the group with the SMSCIC or SMSNOCIC option.

These parameters are invalid if the NOJOB subparameter is specified on the JOB parameter, or if a DBDS group is not specified.

NODEFLT
Optional parameter you use to specify that the implicit skeletal JCL default member, if any, for the DBDS is not to be used.

TIMEFMT(sublist)
Optional parameter you use to define the form in which time stamps appear in user-defined output. This specification overrides both the GENJCL default values and any values set on %SET statements in the input skeletal JCL.

The default for the GENJCL output time format is compressed, with a two-digit year, and the offset in numeric form: 96021315001 +0700. If you want the output time stamps to appear without offsets, for example, you can override the default with TIMEFMT(N).

The override is good only for the duration of a single GENJCL command.

Related Reading:
- The TIMEFMT parameter sublist format is described in “TIMEFMT Parameter” on page 125.
- See “TIMEFMT Subparameter Order of Precedence” on page 127 for information on the precedence of the subparameters.
- See “Standard Time Stamp Format” on page 124 for examples of the different output forms.

UNIT(unittype)
Optional parameter you use to specify the unit type of the first, second, third, or fourth output image copy data sets. These parameters are valid only if both of the following conditions are true:
- The corresponding VOLLIST parameter is specified
- The DBDS for which the JCL is being generated was defined with the NOREUSE option

USERKEYS(%key1, value | %key2)
Optional parameter you use to set the value of keywords you have defined. Up to 32 keywords can be specified.

%key1
User-defined keyword that is being assigned a value. The maximum length of the keyword is eight characters, including the percent sign. The first
character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

'value'

Value assigned to the user-defined keyword when it is encountered. value can be any character string enclosed in single quotes. The maximum length of value is 132 characters (excluding the quotes). If value contains a quote, use two single quotes. value can be a null string (''). If value is a time stamp, it can be zero.

%key2

Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.

Any keyword can be assigned a value with the USERKEYS parameter. However, if you assign a value to DBRC-defined keywords, the value is ignored. DBRC-defined keywords are shown in "Symbolic Keywords" on page 86.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS parameter, the value specified in USERKEYS is used.

VOLLIST(volser)
VOLLIST2(volser)
VOLLIST3(volser)
VOLLIST4(volser)

Optional parameters that you use to specify the volumes on which the image copy data set copies are to reside. These parameters are valid only if:

- The DBDS for which the JCL is being generated is defined with the NOREUSE option.
- DBD is specified with DDN or AREA.

Examples of Running the Database Image Copy Utility

Here are some examples of using the GENJCL.IC command.

Example for DBDS Defined by the DBD and DDN Parameters:  In this example, a GENJCL.IC command generates the JCL and control statements required to run the Database Image Copy utility for the DBDS identified in the DBD and DDN parameters. The default Concurrent Image Copy (CIC) is used. The database is defined as REUSE. If the INIT.DBDS command for the DBDS for which JCL is being generated is specified with ICJCL(member), that skeletal member is used from the data set that is identified in the JCLPDS DD statement. If not, the default skeletal member ICJCL from the JCLPDS data set is used. Output from the generated JCL goes to the data set identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

```
//GENJIC1 JOB
//JCLOUT DD . . .
//JCLPDS DD . . .
::
//SYSIN DD *
   GENJCL.IC DBD(DBDS1) DDN(DDNKS1)
```

The EXEC and SYSIN statements for the generated JCL are shown below:

```
::
//IC1 EXEC PGM=DFSUDMP0,REGION=nnnK,
//     PARM='CIC,GSGNAME='
```
Example for All DBDSs in a Group with NOCIC: In this example, the GENJCL.IC command generates JCL and control statements to run the Image Copy utility for all DBDSs of GROUP1 and batch image copies (NOCIC) are also taken. The skeletal member used is ICJCL1 from the data set identified in PDS4. The keyword, %DEFIC, is a user-defined value in ICJCL1 that is resolved to '1ST USERKEYS PARM'. The default member for the database initialized in INIT.DBDS DEFLTJCL(MEMBER) is not used to resolve keywords. Skeletal member JOBJCL produces a job statement.

Example of Running the Database Image Copy 2 Utility with SMSCIC: In this example, a GENJCL.IC command generates the JCL and control statements required to run the Database Image Copy 2 utility in shared database mode (SMSCIC) for the DBDS that is identified in the DBD and DDN parameters. The database is defined as NOREUSE and four copies are requested. The default skeletal member ICJCL from the JCLPDS data set is used. Output from the generated JCL goes to the data set identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

The following statements are examples of one of the EXEC statements and one of the SYSIN statements for the generated JCL:

Example of Running the Database Image Copy 2 Utility with SMSCIC: In this example, a GENJCL.IC command generates the JCL and control statements required to run the Database Image Copy 2 utility in shared database mode (SMSCIC) for the DBDS that is identified in the DBD and DDN parameters. The database is defined as NOREUSE and four copies are requested. The default skeletal member ICJCL from the JCLPDS data set is used. Output from the generated JCL goes to the data set identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

The EXEC and SYSIN statements for the generated JCL are shown below:
Example of Running the Database Image Copy 2 Utility with SMSNOCIC: In this example, the GENJCL.IC command generates JCL and control statements to run the Database Image Copy 2 utility with exclusive database usage (SMSNOCIC) for the DBDS that is identified in the DBD and DDN parameters. The database is defined as REUSE and the default, one copy, is requested. The global service group name for DBDVSAM2 is GSGN4IC2. The default skeletal member ICJCL from the JCLPDS data set is used. Output from the generated JCL goes to the data set identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement. The database is to be unlocked after the physical copy completes.

The EXEC and SYSIN statements for the generated JCL are shown below:

Example of Running the Database Image Copy 2 utility with an implied database group: All data sets for the database will be copied in a single step to a single output data set, with the Compress option.

The following are the EXEC and SYSIN cards in the generated JCL:
GENJCL.OIC

Use the GENJCL.OIC command to generate the JCL and utility control statements needed to run the Online Database Image Copy utility. For information about the IBM-supplied skeletal JCL execution member used by GENJCL.OIC, see “Generating JCL and User-Defined Output” on page 83.

Important:
• For HALDB partitions, the GENJCL.OIC command treats ILDS and index data sets differently than data DBDSs. The GENJCL.OIC command skips these data sets in groups, regardless of whether the groups are explicit or implicit. If you explicitly specify one of these data sets, the GENJCL.OIC command fails.
• For HALDB DBDSs, the GENJCL.IC command does not generate DD statements for the DBDSs to be copied. The SYSIN control statement identifies which active DBDSs to copy. These can be either the A-through-J data set or the M-through-V data sets.

Restriction: The GENJCL.OIC command and online image copy cannot be used on databases at an RSR-tracking site.

GENJCL.OIC Syntax

Parameters for GENJCL.OIC

DBD(name) | GROUP(name)
Mutually exclusive, required parameters that you use to specify the DBDS or DEDB area that is to be copied.
**DBD**
Specifies the name of the DBDS or DEDB area that is to be copied.

**Restriction for HALDBs:** For the GENJCL.01C command, name specifies the name of either a HALDB master database name or a HALDB partition name.

**GROUP**
Specifies that all DBDSs of a DBDS group are to be copied. If GROUP is specified, the GENJCL.01C command executes repeatedly, once for each DBDS of the DBDS group.

**PSB(name)**
Required parameter you use to specify the name of the PSB that is required for a run of the Online Database Image Copy utility.

If you specify GROUP, the same PSB name is used for all members of the group.

**CHKINT(value)**
Optional parameter you use to specify the checkpoint interval for the Online Database Image Copy utility. value must be a decimal number from 1 to 9999. If this keyword is omitted, the Online Database Image Copy utility uses its own default value for the checkpoint interval.

**COPIES(1 | 2)**
Optional parameter you use to request that the Online Database Image Copy utility in order to produce one or two image copy data sets for the specified DBDS.

If the specified DBDS is identified in RECON with the NOREUSE attribute, you must specify the COPIES parameter in order to produce two image copy data sets; otherwise, one image copy data set is produced.

If the specified DBDS is identified in RECON with the REUSE attribute, you cannot specify a COPIES parameter; the number of image copy data sets that are produced for this DBDS is determined by parameters in the INIT.1C command.

**DDN(name)**
Optional parameter you use to identify the DBDS that is to be copied.

The DDN parameter can be specified only if the DBD parameter is specified. If DDN is not specified, the GENJCL.01C command executes repeatedly, once for each DBDS of the specified database. If you specify a HALDB master name, the GENJCL.01C command is performed for all data DBDSs for each partition in the HALDB master. If you specify a HALDB partition name, the GENJCL.01C command is performed for all data DBDSs of the identified HALDB partition.

For HALDBs, you must specify a partition database name with the DBD parameter in order to use the DDN parameter. The DDN parameter value is the partition DDN. The GENJCL.01C command is performed for the identified DBDS of the partition. The GENJCL.01C command fails if DDN does not identify a data DBDS in the partition.

**DEFAULTS (nmember)**
Optional parameter you use to specify up to 10 names of skeletal JCL default members to be used when generating JCL. Default members are searched in order to resolve keywords in the order in which the members are specified on this parameter.
If a keyword is assigned a value in both the DEFAULTS and USERKEYS parameters, the value specified in USERKEYS is used.

**JCLOUT (JCLOUT | ddname)**
Optional parameter you use to specify the output data set for the generated JCL. The data set is specified by *ddname*. A JCL DD statement with this *ddname* must be included in the job step containing the GENJCL command. The specified data set can be a member of a partitioned data set, but only if it is not the same data set used for the default, JCLOUT.

**JCLPDS (JCLPDS | ddname)**
Optional parameter you use to specify the skeletal JCL data set that is to be used for input when generating JCL. The data set is specified by *ddname*. A JCL DD statement with this *ddname* must be included in the job step that contains the GENJCL command.

**JOB | JOB(member) | NOJOB**
Mutually exclusive, optional parameters that you use to specify whether to produce the job statement in the generated JCL.

**JOB**
Specifies that the job statement is to be produced. When JOB is specified without a member name, the IBM-supplied execution member JOBJCL produces the job statement. When JOB(*member*) is specified, the specified execution member produces the job statement.

**NOJOB**
Specifies that the job statement is not to be produced in the generated JCL.

**LIST | NOLIST**
Mutually exclusive, optional parameters that you use to specify whether to write the generated JCL to the SYSPRINT data set.

**LIST**
Prints the generated JCL.

**NOLIST**
Suppresses the printing of the generated JCL.

**MEMBER(*member*)**
Optional parameter you use to specify the name of the skeletal JCL execution member that is to be used. For a description of the IBM-supplied execution member, see "Generating JCL and User-Defined Output" on page 83. If this parameter is not specified, the default specified on the INIT.DBDS command is used.

**MULTIJOB | ONEJOB**
Mutually exclusive, optional parameters that you use to control how many JOB statements are to be generated when a DBDS group is specified either explicitly or implicitly.

**MULTIJOB**
Processes the skeletal JCL JOB member for each group member (multiple JOB statements are to be produced).

**ONEJOB**
Processes the skeletal JCL JOB member only for the first group member.

These parameters are invalid if NOJOB is specified or if a DBDS group is not specified.
NODEFLT
Optional parameter you use to specify that the implicit skeletal JCL default member, if any, for the DBDS is not to be used.

TIMEFMT(sublist)
Optional parameter you use to define the form in which time stamps appear in user-defined output. This specification overrides both the GENJCL default values and any values set on $\%$ET statements in the input skeletal JCL.

The default for the GENJCL output time format is compressed, with a two-digit year, and the offset in numeric form: 960021315001 +0700. If you want the output time stamps to appear without offsets, for example, you can override the default with TIMEFMT(,N).

The override is good only for the duration of a single GENJCL command.

Related Reading:
- The TIMEFMT parameter sublist format is described in "TIMEFMT Parameter" on page 125.
- See "TIMEFMT Subparameter Order of Precedence" on page 127 for information on the precedence of the subparameters.
- See "Standard Time Stamp Format" on page 124 for examples of the different output forms.

UNIT(3400 | unittype)
Optional parameter you use to specify the unit type of the primary output data set. This parameter is valid only if:
- The VOLLIST parameter was specified.
- The DBDS for which the JCL is being generated was defined with the NOREUSE option.

UNIT2(3400 | unittype)
Optional parameter you use to specify the unit type of the secondary output data set. This parameter is valid only if:
- The VOLLIST2 parameter was specified.
- The DBDS for which the JCL is being generated was defined with the NOREUSE option.

USERKEYS(%key1,’value’ | %key2)
Optional parameter you use to set the value of keywords you have defined. Up to 32 keywords can be specified.

%key1
User-defined keyword that is being assigned a value. The maximum length of the keyword is eight characters, including the percent sign. The first character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

'value'
Value assigned to the user-defined keyword when it is encountered. value can be any character string enclosed in single quotes. The maximum length of value is 132 characters (excluding the quotes). If value contains a quote, use two single quotes. value can be a null string (’’). If value is a time stamp, it can be zero.

%key2
Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.
Any keyword can be assigned a value with the USERKEYS parameter. However, if you assign a value to DBRC-defined keywords, the value is ignored. DBRC-defined keywords are shown in "Symbolic Keywords" on page 86.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS parameter, the value specified in USERKEYS is used.

**VOLLIST**(volser)
Optional parameter you use to specify the volumes on which the image copy data set is to reside. This parameter is valid only if the DBDS for which the JCL is being generated is defined with the NOREUSE option, and if the DBDS is being used with DBDs, not with groups.

**VOLLIST2**(volser)
Optional parameter you use to specify the volumes on which the duplicate image copy data set is to reside. This parameter is valid only if the DBDS for which the JCL is being generated is defined with the NOREUSE option, and if the DBDS is being used with DBDs, not with groups.

**Examples of Running the Online Database Image Copy Utility**
Here are some examples of using the GENJCL.OIC command.

**Example Using JCLPDS Member OICJCL:** In this example, a GENJCL.OIC command generates the JCL and control statements required to run the Online Database Image Copy utility for the DBDS identified in the DBD and DDN parameters. The database is defined as REUSE. If the INIT.DBDS command for the DBDS for which the JCL is being generated is specified with OICJCL(member), that member is used and is found in the data set identified in the JCLPDS DD statement. If not, default skeletal member OICJCL from the JCLPDS data set is used. Output from the generated JCL goes to the data set defined in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

```plaintext
//GENJCL.OIC JOB
//JCLPDS DD . . .
//JCLOUT DD . . .

;
//SYSIN DD *
  GENJCL.OIC DBD(DB0KS0D1) DDN(DDNKS0D1) - PSB(MYJOB)
/*
```

**Example Using JCLPDS Member OICJCL2:** In this example, the GENJCL.OIC command generates JCL and control statements to run the Online Image Copy utility for all DBDSs of GROUP1. The skeletal member used is OICJCL2 from the data set identified in the OICPDS DD statement. One job statement for each group member is generated from the JOBCARD member found in the data set identified in the OICPDS DD statement. %DEF0DBDS is a user-defined value in OICJCL2 and is resolved with 'DATABASE DEFINED HERE'. Members DEF1, DEF2, and DEF3 are used to resolve user-defined keywords in OICJCL2. The default member for the database initialized in INIT.DBDS DEFLTJCL(MEMBER) is not used to resolve keywords. The values specified in the USERKEYS parameter for a keyword override the values found in the DEFAULTS member.

```plaintext
//GENJCL.OIC JOB
//OICOUT DD . . .
//OICPDS DD . . .
//SYSIN DD *
```
GENJCL.RECEIVE

Use the GENJCL.RECEIVE command to apply an image copy from an RSR active site to a database data set or area at an RSR tracking site. This command generates the JCL and utility control statements required to run the database recovery utility for image copy receive. If more than one image copy data set is registered in the RECON for a given DBDS or area, the most recent usable image copy data set is received. A usable image copy is one that meets all of the following requirements:

- Is not flagged as being in error
- Was created by the IMS Batch Image Copy utility or the Database Image Copy 2 utility while the database was unavailable for update processing
- Was created after any updates were received at the tracking site

The GENJCL.RECEIVE command can only be used for RSR-covered databases. Also, the local service group of the covering global service group must be a tracking service group.

**Important:** For HALDB partitions, the GENJCL.RECEIVE command treats ILDS and index data sets differently than data DBDSs. The GENJCL.RECEIVE command skips these data sets in groups, regardless of whether the groups are explicit or implicit. If you explicitly specify one of these data sets, the GENJCL.RECEIVE command fails.

For information about the IBM-supplied skeletal JCL execution member that is used by GENJCL.RECEIVE, see "Generating JCL and User-Defined Output" on page 83.

**GENJCL.RECEIVE Syntax**

```
GENJCL.RECEIVE DBD(name) GROUP(name)Defaults(member) DDN(ddname) AREA(name)
```

```
JCLOUT(ddname) JCLPDS(ddname) JOB(member) NOJOB LIST
```

```
MEMBER(member) TIMEFMT(sublist)
```

```
USERKEYS((%key1, 'value') %key2)
```

**Parameters for GENJCL.RECEIVE**

**DBD(name) | GROUP(name)**

Mutually exclusive, required parameters that you use to specify the database that is to be received.
DBD
   Specifies the name of the database to be received. The database must be RSR-covered.

Restriction for HALDBs: For the GENJCL.RECEIVE command, name specifies the name of either a HALDB master database name or a HALDB partition name.

GROUP
   Specifies that image copies for all DBDSs of a DBDS or CA group are to be received. If GROUP is specified, the GENJCL.RECEIVE command executes repeatedly, once for each DBDS of the DBDS or CA group. If you attempt an implicit or explicit group execution with recoverable and nonrecoverable DBDSs, JCL is not generated for the nonrecoverable DBDSs.

   If GROUP is specified, all DBDS areas of the group must be covered by the same global service group.

DEFAULTS (member)
   Optional parameter you use to specify up to 10 names of skeletal JCL default members that are to be used when generating JCL. Default members are searched to resolve keywords in the order in which the members are specified on this parameter.

   If a keyword is assigned a value in both the DEFAULTS and the USERKEYS parameters, the value specified in USERKEYS is used.

DDN(name) | AREA(name)
   Mutually exclusive, optional parameters that you use to identify the DBDS ddbname or DEDB area to be received.

   The DDN or AREA parameter is specified only if the DBD parameter is specified.

   For HALDBs, you must specify a partition database name with the DBD parameter in order to use the DDN parameter. The DDN parameter value is the partition DDN. The GENJCL.RECEIVE command is performed for the identified DBDS of the partition. The GENJCL.RECEIVE command fails if DDN does not identify a data DBDS in the partition.

   If DDN or AREA is not specified, the GENJCL.RECEIVE command executes repeatedly, once for each DBDS or area of the specified database. If you specify a HALDB master name, the GENJCL.RECEIVE command is performed for all data DBDSs for each HALDB partition in the HALDB master. If you specify a HALDB partition name, the GENJCL.RECEIVE command is performed for all data DBDSs of the identified partition.

JCLOUT (JCLOUT | ddname)
   Optional parameter you use to specify the output data set for the generated JCL. The data set is specified by ddname. A JCL DD statement with this ddbname must be included in the job step containing the GENJCL command. The specified data set can be a member of a partitioned data set, but only if it is not the same data set that is used for the default (JCLOUT).

JCLPDS (JCLPDS | ddname)
   Optional parameter you use to specify the skeletal JCL data set that is to be used for input when generating JCL. The data set is specified by ddname. A JCL DD statement with this ddbname must be included in the job step containing the GENJCL command.
MEMBER

LIST

USERKEYS

TIMEFMT

JOB

NOJOB

Specifies that the job statement is to be produced. When JOB is specified without a member name, the IBM-supplied execution member ICRCVJCL produces the job statement. When JOB(member) is specified, the specified execution member produces the job statement.

NOJOB

Specifies that the job statement is not to be produced in the generated JCL.

LIST

NOLIST

Mutually exclusive, optional parameters that you use to specify whether to write the generated JCL to the SYSPRINT data set.

LIST

Prints the generated JCL.

NOLIST

Suppresses printing of the generated JCL.

MEMBER(member)

Optional parameter you use to specify the name of the skeletal JCL execution member that is to be used. For a description of the IBM-supplied execution member, see “Generating JCL and User-Defined Output” on page 83. If this parameter is not specified, the default specified on the INIT.DBDS command is used.

TIMEFMT(sublist)

Optional parameter you use to define the form in which time stamps appear in user-defined output. This specification overrides both the GENJCL default values and any values set on %SET statements in the input skeletal JCL.

The default for the GENJCL output time format is compressed, with a two-digit year, and the offset in numeric form: 960021315001 +0700. If you want the output time stamps to appear without offsets, for example, you can override the default with TIMEFMT(N).

The override is good only for the duration of a single GENJCL command.

Related Reading:

- The TIMEFMT parameter sublist format is described in “TIMEFMT Parameter” on page 125.
- See “TIMEFMT Subparameter Order of Precedence” on page 127 for information on the precedence of the subparameters.
- See “Standard Time Stamp Format” on page 124 for examples of the different output forms.

USERKEYS(%key1,’value’ | %key2)

Optional parameter you use to set the value of keywords you have defined. Up to 32 keywords can be specified.

%key1

User-defined keyword that is being assigned a value. The maximum length of the keyword is eight characters, including the percent sign. The first character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

'value'

Value assigned to the user-defined keyword when it is encountered. value
can be any character string enclosed in single quotes. The maximum length of value is 132 characters (excluding the quotes). If value contains a quote, use two single quotes. value can be a null string ("'). If value is a time stamp, it can be zero.

%key2
Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.

Any keyword can be assigned a value with the USERKEYS parameter. However, if you assign a value to DBRC-defined keywords, the value is ignored. DBRC-defined keywords are shown in "Symbolic Keywords" on page 86.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS parameter, the value specified in USERKEYS is used.

Examples of Running the Database Recovery Utility to Receive an Image Copy
Here are some examples of using the GENJCL.RECEIVE command.

Example for the DBDS Identified by the DBD and DDN Parameters: In this example, a GENJCL.RECEIVE command generates the JCL and control statements that are required to run the database recovery utility to receive an image copy for the DBDS that is identified in the DBD and DDN parameters.

If the INIT.DBDS command for the DBDS, for which the JCL is being generated, is specified with RECVJCL(member), that member is used and is found in the data set that is identified in the JCLPDS DD statement. If not, default skeletal member RECVJCL, from the JCLPDS data set is used. Output from the generated JCL goes to the data set that is identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

//GENJRCVE JOB
//JCLPDS DD . . .
//JCLOUT DD . . .

; SYSIN DD *
GENJCL.RECEIVE DBD(DBESDSA) DDN(DDESDSA)
/*

Example for All DBDSs in a Group: In this example, the GENJCL.RECEIVE command generates JCL and control statements to run the database recovery utility to receive image copies for all DBDSs of GROUP1.

The skeletal member used is RCVJCL2 from the data set identified in the PDS DD statement. Skeletal member JOBJCL produces a job statement for each member of the group. %DEFDBD1 and %DEFDBD2 are user-defined values in member RCVJCL2, which resolve to 'DEFINE DB1' and 'DEFINE DB2'. Default members DEF1, DEF2, and DEF3 are used to resolve user-defined keywords in RECVJCL2. The default member for the DBDS, if initialized in the INIT.DBDS DEFLTJCL(MEMBER) command, is also used to resolve keywords. The values in the explicitly defined DEFAULTS members override values in the predefined DEFLTJCL member. The values specified in the USERKEYS parameter for a keyword override the values found in the DEFAULTS members.

//GENJRCVE1 JOB
//OUT DD . . .
//PDS DD . . .
GENJCL.RECEIVE

Use the GENJCL.RECEIVE command to generate the JCL and utility control statements required to run the database recovery utility. You can request the JCL and utility control statements for a full recovery or a time-stamp recovery of a specified DBDS or area. All log data must be archived; otherwise the GENJCL.RECEIVE command fails.

Restrictions:

- A nonstandard image copy data set cannot be used as input to the database recovery utility. The procedure for recovering a database with a nonstandard image copy is slightly different depending on whether the IMS system is an active or tracking subsystem.
- A GENJCL.RECEIVE command cannot be issued for a user-recoverable database.

Active subsystem
1. Restore the DBDS from the nonstandard image copy.
2. Record this restoration by entering a NOTIFY.RECEIVE command with the image copy run time as the RCVTIME parameter.
3. Complete the recovery, applying changes made since the image copy, by entering a GENJCL.RECEIVE command with the USEDBDS parameter.

Tracking subsystem
1. Restore the DBDS from the nonstandard image copy. This must be the latest recorded image copy.
2. Record this restoration by entering a NOTIFY.RECEIVE command with the image copy run time as the RUNTIME parameter and the USID from the image copy as the RUNUSID parameter.
3. Issue a /START command for the database. Online forward recovery (OFR) automatically completes, applying changes made since the image copy.

For information about the IBM-supplied skeletal JCL execution member used by GENJCL.RECEIVE, see “Generating JCL and User-Defined Output” on page 83.

Use the RESTORE parameter to recover DBDSs that were designated as nonrecoverable.

Important: For HALDB partitions, the GENJCL.RECEIVE command treats ILDS and index data sets differently than data DBDSs. The GENJCL.RECEIVE command skips these data sets in groups, regardless of whether the groups are explicit or implicit. If you explicitly specify one of these data sets, the GENJCL.RECEIVE command fails.

Restriction: The GENJCL.RECEIVE command does not support ILDS and index data sets. To generate JCL for the HALDB Index/ILDS Rebuild Utility (DFSPREC0), use the GENJCL.USER command. See “Sample JCL for HALDB INDEX/ILDS Rebuild Utility (DSPUPJCL)” on page 466 for information about IBM-supplied sample JCL that you can use.
GENJCL.RECOV

GENJCL.RECOV Syntax

```
GENJCL.RECOV
  DBD(name)
  GROUP(name)
  DEFAULTS(member)
  DDN(ddname)
  AREA(name)
  JCLPDS(ddname)
  JCLOUT(ddname)
  JCPUT
  JOB(member)
  NODEFLT
  RCVTIME(time_stamp)
  USEIC
  USEDBDS
  USEAREA
  USEOLRRG
  USERKEYS(('%key1', 'value'), '%key2')

```

Parameters for GENJCL.RECOV

**DBD(name)** | **GROUP(name)**
---|---
Mutually exclusive, required parameters that you use to specify the DBDSs that are to be recovered.

**DBD**

Specify the database name of the DBDSs to be recovered.

Restriction for HALDBs: For the GENJCL.RECOV command, name specifies the name of either a HALDB master database name or a HALDB partition name.

**GROUP**

Specify that all DBDSs of a DBDS or CA group are to be recovered. If GROUP is specified, the GENJCL.RECOV command executes repeatedly for each DBDS of the DBDS or CA group. If you attempt an implicit or explicit group execution with recoverable and nonrecoverable DBDSs (and RESTORE is not specified), JCL is not generated for the nonrecoverable DBDSs.

**DEFAULTS (member)**

Optional parameter you use to specify up to 10 names of skeletal JCL default members to be used when generating JCL. Default members are searched to resolve keywords in the order in which the members are specified on this parameter.

If a keyword is assigned a value in both the DEFAULTS and the USERKEYS parameters, the value specified in USERKEYS is used.

**DDN(name) | AREA(name)**

Mutually exclusive, optional parameters that you use to identify the DBDS ddname or DEEB area to be recovered.

The DDN or AREA parameter is specified only if the DBD parameter is specified.

For HALDBs, you must specify a partition database name with the DBD parameter in order to use the DDN parameter. The DDN parameter value is
the partition DDN. The **GENJCL.RECOV** command is performed for the identified DBDS of the partition. The **GENJCL.RECOV** command fails if DDN does not identify a data DBDS in the partition.

If DDN or AREA is not specified, the **GENJCL.RECOV** command executes repeatedly for each DBDS or area of the specified database. If you specify a HALDB master name, the **GENJCL.RECOV** command is performed for all data DBDSs for each HALDB partition in the HALDB master. If you specify a HALDB partition name, the **GENJCL.RECOV** command is performed for all data DBDSs of the identified partition.

**JCLOUT (JCLOUT | ddname)**
Optional parameter you use to specify the output data set for the generated JCL. The data set is specified by `ddname`. A JCL DD statement with this ddname must be included in the job step containing the **GENJCL** command. The specified data set can be a member of a partitioned data set, but only if it is not the same data set used for the default (JCLOUT).

**JCLPDS (JCLPDS | ddname)**
Optional parameter you use to specify the skeletal JCL data set that is to be used for input when generating JCL. The data set is specified by `ddname`. A JCL DD statement with this ddname must be included in the job step containing the **GENJCL** command.

**JOB | JOB(member) | NOJOB**
Mutually exclusive, optional parameters that you use to specify whether to produce the job statement in the generated JCL.

**JOB**
Specifies that the job statement is to be produced. When JOB is specified without a member name, the IBM-supplied execution member JOBJCL produces the job statement. When JOB(`member`) is specified, the specified execution member produces the job statement.

**NOJOB**
Specifies that the job statement is not to be produced in the generated JCL.

**LIST | NOLIST**
Mutually exclusive, optional parameters that you use to specify whether to write the generated JCL to the SYSPRINT data set.

**LIST**
Prints the generated JCL.

**NOLIST**
Suppresses printing of the generated JCL.

**MEMBER(member)**
Optional parameter you use to specify the name of the skeletal JCL execution member to be used. For a description of the IBM-supplied execution member, see “Generating JCL and User-Defined Output” on page 83. If this parameter is not specified, the default specified on the **INIT.DBDS** command is used.

**MULTIJOB | ONEJOB**
Mutually exclusive, optional parameters that you use to control how many JOB statements are generated when a DBDS group is specified either explicitly or implicitly.

**MULTIJOB**
Processes the skeletal JCL JOB member for each group member (multiple JOB statements are produced).
**ONEJOB**

Processes the skeletal JCL JOB member only for the first group member.

These parameters are invalid if NOJOB is specified or a DBDS group is not specified.

**NODEFLT**

Optional parameter you use to specify that the implicit skeletal JCL default member, if any, for the DBDS is not to be used.

**RCVTIME(time_stamp)**

Optional parameter you use to specify a time-stamp recovery, which is a partial recovery of a DBDS or area to a point in time earlier than its most recent state. If you omit this parameter, you are requesting a full recovery to the most recent state.

A valid time stamp for a partial recovery is any point at which there are no allocations of the DBDS or area and there is not a merge of logs needed that cannot be resolved by running the change accumulation utility.

**Attention:** An allocation that has no de-allocation time recorded persists until the stop time of the current log.

**Related Reading:**
- See “Using the %SET TIMEFMT Keyword” on page 93 for more information on full recovery.

**RESTORE**

Optional parameter used to generate JCL for a database data set that is designated as nonrecoverable. If the last image copy was taken before the DBDS was designated as nonrecoverable, normal recovery JCL is generated to recover the DBDS up to the point of the recovery-status change. If the last image copy was taken after the DBDS was designated as nonrecoverable, the generated JCL uses only the image copy for recovery.

If you attempt an implicit or explicit group execution with recoverable and nonrecoverable DBDSs (and RESTORE is specified), JCL is generated only for the nonrecoverable DBDSs.

Do not specify RESTORE for a recoverable DBDS.

**TIMEFMT(sublist)**

Optional parameter you use to define the form in which time stamps appear in user-defined output. This specification overrides both the GENJCL default values and any values set on %SET statements in the input skeletal JCL.

The default for the GENJCL output time format is compressed, with a two-digit year, and the offset in numeric form: 960021315001 +0700. If you want the output time stamps to appear without offsets, for example, you can override the default with TIMEFMT(N).

The override is good only for the duration of a single GENJCL command.

**Related Reading:**
- The TIMEFMT parameter sublist format is described in “TIMEFMT Parameter” on page 125.
- See “TIMEFMT Subparameter Order of Precedence” on page 127 for information on the precedence of the subparameters.
• See “Standard Time Stamp Format” on page 124 for examples of the different output forms.

**USEIC | USEDBDS | USEAREA | USEOLRRG**

Mutually exclusive, optional parameters that you use to specify the starting point of the requested recovery action.

**USEIC**

Starts the recovery with an image copy data set. You can then apply subsequent changes that occurred in the DBDS.

USEIC is the default.

**USEDBDS**

Recovery is performed using only the changes that have occurred to the DBDS in its current state. An image copy data set is not used as input to this recovery. You can specify the USEDBDS parameter only if you also specify the DBDS parameter, and only after performing a time-stamp recovery in which an image copy data set is used as input.

**USEAREA**

Recovery is performed using only the changes that have occurred to the DEDB area in its current state. An image copy data set is not used as input to this recovery. You can specify USEAREA only if you also specify the AREA parameter, and only after performing a time-stamp recovery in which an image copy data set is used as input.

**USEOLRRG**

Starts the recovery with a HALDB Online Reorganization updates.

You can use these parameters to recover a DBDS or area to a specified time stamp using an image copy data set and then apply the changes that have occurred since the image copy by specifying an additional recovery using the USEDBDS or USEAREA parameter.

**Restriction:** If this required time-stamp recovery restored the DBDS or DEDB area to a time that falls within an existing time-stamp recovery’s range (the time between the RECOV TO and RUN times), then the USEDBDS or USEAREA parameter is invalid.

**USERKEYS(%key1,'value' | %key2)**

Optional parameter you use to set the value of keywords you have defined. Up to 32 keywords can be specified.

%key1

User-defined keyword that is being assigned a value. The maximum length of the keyword is eight characters, including the percent sign. The first character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

'value'

Value assigned to the user-defined keyword when it is encountered. value can be any character string enclosed in single quotes. The maximum length of value is 132 characters (excluding the quotes). If value contains a quote, use two single quotes. value can be a null string ("""). If value is a time stamp, it can be zero.

%key2

Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.
Any keyword can be assigned a value with the USERKEYS parameter. However, if you assign a value to DBRC-defined keywords, the value is ignored. DBRC-defined keywords are shown in "Symbolic Keywords" on page 86.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS parameter, the value specified in USERKEYS is used.

Examples of Running the Database Recovery Utility
Here are some examples of using the GENJCL.RECOV command.

Example for the DBDS Identified in the DBD and DDN Parameters: In this example, a GENJCL.RECOV command generates the JCL and control statements required to run the database recovery utility for the DBDS identified in the DBD and DDN parameters. The USEIC parameter indicates that the time-stamp recovery starts with an image copy data set and ends with the log data set that has the stop time stamp specified in the RCVTIME parameter.

If the INIT.DBDS command for the DBDS for which the JCL is being generated is specified with RECOVJCL(member), that member is used and is found in the data set identified in the JCLPDS DD statement. If not, default skeletal member RECOVJCL from the JCLPDS data set is used. Output from the generated JCL goes to the data set identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

```
//GENJRCOV JOB
//JCLPDS DD . . .
//JCLOUT DD . . .

:  //SYSIN DD *
   GENJCL.RECOV DBD(DBESDSA) DDN(DDESDSA) USEIC
      RCVTIME(821001212130)
/*
```

Example for All DBDSs in a Group: In this example, the GENJCL.RECOV command generates JCL and control statements to run the database recovery utility for all DBDSs of GROUP1. The skeletal member that is used is RECJCL2 from the data set identified in the PDS DD statement.

Skeletal member JOBJCL produces a job statement for each member of the group. %DEFDB01 and %DEFDB02 are user-defined values in member RECJCL2 which resolve to 'DEFINE DB1' and 'DEFINE DB2'. Default members DEF1, DEF2, and DEF3 are used to resolve user-defined keywords in RECJCL2. The default member for the DBDS, if initialized in the INIT.DBDS DEFLTJCL(MEMBER) command, is also used to resolve keywords. The values in the explicitly defined DEFAULTS members override values in the predefined DEFLTJCL member. The values specified in the USERKEYS parameter for a keyword override the values found in the DEFAULTS members.

```
//GENJRCV1 JOB
//OUT DD . . .
//PDS DD . . .
//SYSIN DD *
   GENJCL.RECOV GROUP(GROUP1) MEMBER(RECJCL2) -
      JCLPDS(PDS) JCLOUT(OUT) -
      USERKEYS(%DEFDB01,'DEFINE DB1'),(%DEFDB02,'DEFINE DB2') -
      DEFAULTS(DEF1,DEF2,DEF3)
```
GENJCL.USER

Use the GENJCL.USER command to generate JCL or any kind of user output. You must provide the skeletal JCL execution member that is needed for the GENJCL.USER command. For more information, see “Generating JCL and User-Defined Output” on page 83.

GENJCL.USER Syntax

```
GENJCL.USER MEMBER(name) DBD(name) DDN(ddname) 
  JCLOUT(JCLOUT) JCLPOS(JCLPOS) JOB
  NOJOB
  NOLIST
  LIST
  MULTJOB
  ONEJOB
  NODEFLT
  PARTONLY
  PSB(name)
  SSID(name)
  DEFAULTS(member)
  TIMEFMT(sublist)
  USERKEYS((%key1, 'value'), %key2)
```

Parameters for GENJCL.USER

MEMBER(name)

Required parameter you use to specify the name of the skeletal JCL execution member that is used to generate output. You must have already supplied the execution member.

The name can be any valid member name for a partitioned data set. If the specified member does not exist in the skeletal JCL data set, the command fails.

DBD(name) | GROUP(name)

Mutually exclusive, optional parameters that you use to set the value of the %dbname keyword.

DBD

If you specify DBD without the DDN parameter, the GENJCL.USER command executes repeatedly for each DBDS or the specified database.

Restriction for HALDBs: For the GENJCL.USER command, name specifies the name of either a HALDB master database name or a HALDB partition name. You can use this parameter to set the value of the %dbname keyword to be either a HALDB master name or a HALDB partition name. If you use a HALDB master name, the GENJCL.USER command is performed for all data DBDSs for each HALDB partition in the HALDB master. If you use a HALDB partition name, the GENJCL.USER command is performed for all DBDSs of the identified partition.

GROUP

If you specify GROUP, the GENJCL.USER command executes repeatedly, once
for each DBDS of the specified DBDS group. For each repeated execution, the DBD and DDN parameters are set to the corresponding group member.

If you specify neither DBD nor GROUP, the value of the %dbname keyword is null unless a value is assigned in the USERKEYS parameter or a skeletal JCL default member.

**DDN**(name)
Optional parameter you use to set the value of the %ddname keyword. If you do not specify DDN, the value of the %ddname keyword is null unless a value is assigned in the USERKEYS parameter or a skeletal JCL default member.

For HALDBs, you must specify a partition database name with the DBD parameter in order to use the DDN parameter. In this case, the DDN is the partition DDN. The GENJCL.USER command is performed for the identified DBDS of the partition. The GENJCL.USER command fails if DDN does not identify a DBDS in the partition.

You cannot specify DDN if you also specify GROUP.

**DEFAULTS**(member)
Optional parameter you use to specify up to 10 names of skeletal JCL default members to be used when generating JCL or other user-defined output. Default members are searched to resolve keywords in the order in which the members are specified on this parameter.

If a keyword is assigned a value in both the DEFAULTS and USERKEYS parameters, the value specified in USERKEYS is used.

**JCLOUT**(JCLOUT | ddname)
Optional parameter you use to specify the output data set for the generated JCL or other user-defined output. The data set is specified by ddname. A JCL DD statement with this ddname must be included in the job step containing the GENJCL.USER command. The specified data set can be a member of a partitioned data set, but only if it is not the same data set that is used for the default (JCLOUT).

**JCLPDS**(JCLPDS | ddname)
Optional parameter you use to specify the skeletal JCL data set that is to be used for input when generating the JCL or other user-defined output. The data set is specified by ddname. A JCL DD statement with this ddname must be included in the job step containing the GENJCL.USER command.

**JOB | JOB**(member) | NOJOB
Mutually exclusive, optional parameters that you use to specify whether to produce the job statement in the generated JCL.

**JOB**
Specifies that the job statement is to be produced. When JOB is specified without a member name, the IBM-supplied execution member JOBJCL produces the job statement. When JOB(member) is specified, the specified execution member produces the job statement.

**NOJOB**
Specifies that the job statement is not to be produced in the generated JCL.

**LIST | NOLIST**
Mutually exclusive, optional parameters that you use to specify whether to write the generated JCL to the SYSPRINT data set.

**LIST**
Prints the generated JCL.
NOLIST
Suppresses printing of the generated JCL.

MULTIJOB | ONEJOB
Mutually exclusive, optional parameters that you use to control how many JOB statements are generated when a DBDS group is specified either explicitly or implicitly.

MULTIJOB
Processes the skeletal JCL JOB member for each group member (multiple JOB statements are produced).

ONEJOB
Only processes the skeletal JCL JOB member for the first group member.

These parameters are invalid if NOJOB is specified or if a DBDS group is not specified.

NODEFLT
Optional parameter you use to specify that the implicit skeletal JCL default member, if any, for the DBDS is not to be used. If you do not specify GROUP or DBD, this parameter is ignored.

PARTONLY
Optional parameter you use to specify that only the HALDB partitions are to be processed instead of all data DBDSs for each HALDB partition. PARTONLY is only meaningful for HALDB. If the specified DBD is not a HALDB master name or a HALDB partition name then this parameter is ignored. This parameter sets the value of %dbname to the HALDB master name and %dbname to the HALDB partition name.

PSB(name)
Optional parameter you use to set the value of the %PSB keyword.

name can be any character string. It does not need to be an actual PSB ddname. The maximum length of the name is eight characters.

If you do not specify PSB, the value of the %PSB keyword is null unless a value is assigned in the USERKEYS parameter or a skeletal JCL default member.

SSID(name)
Optional parameter used to set the value of the %SSID keyword.

name can be any character string. It does not need to be an actual IMS subsystem ID. The maximum length of the name is eight characters.

If the SSID parameter is not specified, the value of the %SSID keyword is null, unless a value is assigned in the USERKEYS parameter or a skeletal JCL default member.

TIMEFMT(sublist)
Optional parameter you use to define the form in which time stamps appear in user-defined output. This specification overrides both the %GENJCL default values and any values set on %SET statements in the input skeletal JCL.

The default for the %GENJCL output time format is compressed, with a two-digit year, and the offset in numeric form: 960021315001 +0700. If you want the output time stamps to appear without offsets, for example, you can override the default with TIMEFMT(N).

The override is good only for the duration of a single %GENJCL command.

Related Reading:
**GENJCL.USER**

- The `TIMEFMT` parameter sublist format is described in "TIMEFMT Parameter" on page 125.
- See "TIMEFMT Subparameter Order of Precedence" on page 127 for information on the precedence of the subparameters.
- See "Standard Time Stamp Format" on page 124 for examples of the different output forms.

**USERKEYS(%key1,'value' | %key2)**

Optional parameter you use to set the value of keywords you have defined.

Up to 32 keywords can be specified.

%key1

User-defined keyword being assigned a value. The maximum length of the keyword is eight characters, including the percent sign. The first character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

'value'

Value assigned to the user-defined keyword when it is encountered. `value` can be any character string enclosed in single quotes. The maximum length of `value` is 132 characters (excluding the quotes). If `value` contains a quote, use two single quotes. `value` can be a null string (''). If `value` is a time stamp, it can be zero.

%key2

Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.

Any keyword can be assigned a value with the USERKEYS parameter. However, if you assign a value to DBRC-defined keywords, the value is ignored. DBRC-defined keywords are shown in "Symbolic Keywords" on page 86.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS parameter, the value specified in USERKEYS is used.

**Example of Running DBRC**

In this example, the `GENJCL.USER` command generates JCL from member USER4 in the data set that is identified in DD statement MYJCLPDS. Output from the generated JCL goes to the data set identified in DD statement JCLOUT. Substitutions for %SSID, %DBNAME and %DDNAME should be made. Skeletal member JOBJCL produces a job statement.

```plaintext
//GENUSER JOB
//JCLOUT DD
//MYJCLPDS DD
//SYSIN DD *
GENJCL.USER MEMBER(USER4) JCLPDS(MYJCLPDS) DBD(DHONTZ04) -
SSID(IMSA) DDN(HIDAM)
```

The following example shows the member USER4 that is to be executed:

```plaintext
/ ADD LIST=ALL,NAME=USER4,LEVEL=01,SOURCE=0
/ NUMBER NEW1=00000100,INCR=100
/******************************************************************************
//* MEMBER NAME = USER4 *
//* (SSID) SHOULD BE SUBSTITUTED IN LIST.SUBSYS COMMAND *
//* (DBNAME) SHOULD BE SUBSTITUTED IN LIST.DDBS COMMAND *
//* (DDNAME) SHOULD BE SUBSTITUTED IN LIST.DDBS COMMAND *
******************************************************************************
//USER4 EXEC PGM=DSPURX00
//SYSPRINT DD SYSOUT=A
```
//SYSIN DD *
LIST.SUBSYS SSID(%SSID) /* SSID SHOULD BE SUBSTITUTED */
LIST.DBDS DBD(%DBNAME) DDN(%DDNAME) -
   /* DBNAME and DDNAME SHOULD BE SUBSTITUTED */
/*
Chapter 12. INIT Commands

Use the INIT commands to make the following changes to a RECON:

- Create entries
- Create CA groups
- Register DBDSs or DEDB areas
- Define global service groups
- Register HALDB partitions
- Initialize a RECON

INIT Commands In This Chapter

The following is a list of all the INIT commands:

- "INIT.ADS" on page 260
- "INIT.CA" on page 261
- "INIT.CAGRP" on page 261
- "INIT.DB" on page 263
- "INIT.DBDS" on page 266
- "INIT.DBDSGRP" on page 271
- "INIT.GSG" on page 273
- "INIT.IC" on page 274
- "INIT.PART" on page 275
- "INIT.RECON" on page 280
- "INIT.SG" on page 285

INIT.ADS

Use an INIT.ADS command to create an entry in RECON that defines an ADS (area data set) that belongs to an area. An area can consist of a maximum of seven data sets.

Before you issue the INIT.ADS command, you must create the area and database records in RECON. If the ADDN or ADSN names are not unique for this area, the INIT.ADS command fails. In addition, the INIT.ADS command fails if the specified area is not registered in RECON.

When you register the area with an DS command, the area status is set as “recovery needed” to prevent inadvertent use by the IMS online system before you have completed registration of the required ADSs. You can create the ADS records in RECON, but only if you use the INIT.ADS command with the default UNAVAIL parameter. You must first issue a CHANGE.DBDS command for the area, specifying the NORECOV option to make the status of the ADS immediately available (with INIT.ADS with the AVAIL parameter).

INIT.ADS Syntax

```
INIT.ADS ADDN(name) ADSN(name) AREA(name) DBD(name)
```

© Copyright IBM Corp. 1974, 2006
Parameters for INIT.ADS

**ADDN**(name)
- Required parameter you use to identify the ADS that is being identified to DBRC by its ddname.

**ADSN**(name)
- Required parameter you use to identify the ADS that is being identified to DBRC by its data set name.

**AREA**(name)
- Required parameter you use to identify the area name for which an ADS is being identified to DBRC.

**DBD**(name)
- Required parameter you use to identify that area by database name for which an ADS is being identified to DBRC.

**AVAIL | UNAVAL**
- Mutually exclusive, optional parameters that you use to indicate whether the ADS record is available.
  - **AVAIL**
    - Makes the ADS status available. The INIT.ADS AVAIL command fails if you issue it when the area is in use or if the area needs to be recovered.
  - **UNAVAL**
    - Makes the ADS status unavailable.

**Example of Creating a Record That Defines an ADS**

In this example, a record is created in RECON that identifies an ADS.

```plaintext
//INITADS JOB :
:
//SYSIN DD *
   INIT.ADS DBD(DBD03) AREA(AREA03) -
      ADDN(AREADDN1) ADSN(AREADSN2)
/*

**INIT.CA**

Use an INIT.CA command to create a record in RECON that identifies a change accumulation data set that is available for future use by the Change Accumulation utility in processing the specified CA group. You can create such change accumulation records only for those CA groups that have been defined with the REUSE option of the INIT.CAGR command. You can create change accumulation records in RECON up to the number specified in the GRPMAX parameter of the INIT.CAGR command that you used to define the CA group.

**INIT.CA Syntax**

```plaintext
INIT.CA CADSN(name) GRPNAME(name) VOLLIST(volser) FILESEQ(value)
```
Parameters for INIT.CA

**CADSN(name)**
Required parameter you use to specify the name of the change accumulation data set for which you are creating a record in RECON. The name you substitute in the variable field can be up to 44 characters. You can use the default naming convention for change accumulation data sets to assign this name.

**GRPNAME(name)**
Required parameter you use to specify the name of the CA group for which you are creating the record. The GRPNAME keyword must specify the name of a CA group that is already defined in RECON.

**VOLLIST(volser)**
Required parameter you use to specify the volume serial numbers of the volumes on which the change accumulation data set being defined are to reside. You can substitute from 1 to 255 volume serial numbers in the variable field; each can be up to six alphanumeric characters long, and they must follow z/OS JCL conventions for volume serial numbers.

**FILESEQ(1 | value)**
Optional parameter you use to specify the file-sequence number of the change accumulation data set that is being defined.

*value* must be a decimal number from 1 to 9999.

**UNIT(3400 | unittype)**
Optional parameter you use to specify the unit type of the volumes on which the change accumulation data sets are to reside. The unit type can be up to eight alphanumeric characters.

Example of Creating a Record That Defines a CA Data Set

In this example, a record is created in RECON that identifies a change accumulation data set (identified by the CADSN parameter). This change accumulation data set is being created for use by a subsequent run of the Change Accumulation utility for the CA group identified in the GRPNAME parameter. Creation of this record implies that the identified CA group was defined with a REUSE parameter.

```plaintext
//INITCA JOB
:
//SYSIN DD *
INIT.CA GRPNAME(CAGRP1) -
   CADSN(IMS.CAGRP1.CA.CA001) -
   VOLLIST(VOL001) FILESEQ(4)
/*

INIT.CAGRP

Use an INIT.CAGRP command to specify the DBDSs that are to belong to a specified CA group. You must have created a record in RECON with an INIT.DBDS command for each DBDS in the CA group before you assign it to a CA group. Each DBDS can belong to only one CA group.
**Restriction:** Index and ILDS DBDSs are not recoverable, and changes to them are not logged. The INIT.CAGRP command does not support these data sets.

**INIT.CAGRP Syntax**

```
INIT.CAGRP—GRPMAX(value)—GRPMEM(dbdname,ddname)—GRPNAME(name)
```

**Parameters for INIT.CAGRP**

**GRPMAX(value)**

Required parameter you use to specify the maximum number of change accumulation data sets that DBRC is to maintain for the specified CA group. The value must be a decimal number from 2 to 1024.

When the number of times you run the Change Accumulation utility for the specified group exceeds the GRPMAX value, the record of the earliest change accumulation run for the group is deleted if you specify the NOREUSE keyword for this CA group. The record of the earliest change accumulation run is reused if you specify the REUSE keyword for this CA group.

**GRPMEM(dbdname,ddname)**

Required parameter you use to specify the names of the DBDSs that are to be members of the CA group you are defining.

There can be from 1 to 32767 members in a CA group. The names you substitute in the variable field must be pairs of names enclosed in parentheses, where `dbdname` is the database name of the DBDS, and `ddname` is its data set ddname.

**Restrictions:**

- When using the INIT.CAGRP command for HALDBs, `dbdname` specifies the name of a HALDB partition.
- User-recoverable databases cannot be added to a CA group.

**GRPNAME(name)**

Required parameter you use to specify the name of the CA group being created. The name can be up to eight alphanumeric characters, and it must not be the same as the name of a CA group that already exists in RECON.

**CAJCL(CAJCL | member)**

Optional parameter you use to specify the name of a member of a partitioned data set of skeletal JCL. You create this member to be used to generate the JCL required to run the Change Accumulation utility for the CA group being created.

**DEFLTJCL(DEFLTJCL | member)**

Optional parameter you use to specify an implicit skeletal JCL default member for the CA group. The specified member is used by the `GENJCL.CA` command to resolve keywords you have defined.

**NOREUSE | REUSE**

Mutually exclusive, optional parameters that you use to specify whether the change accumulation data sets for the CA group being defined can be reused.
REUSE
Indicates that the Change Accumulation utility is to reuse the oldest change accumulation data set and record (for the group being defined) when the GRPMAX value for the group is exceeded. Reuse means that the Change Accumulation utility uses the same physical space, volumes, data set name, and record in RECON for the new change accumulation data set as were used for the oldest change accumulation data set in the group.

NOREUSE
Indicates that the change accumulation data sets in this group are not to be reused by the Change Accumulation utility.

Example of Creating a CA Group
In this example, a CA group is being created. A maximum of 15 change accumulation data sets are to be maintained for this group. (This is indicated in the GRPMAX parameter.) The NOREUSE parameter indicates that change accumulation data sets for this group are not to be reused by the Change Accumulation utility when the GRPMAX value has been reached. That parameter also implies that empty change accumulation data sets cannot be defined for this group for use in future runs of the Change Accumulation utility.

//INITCAGRP JOB
:
:
//SYSIN DD *
INIT.CAGRP GRPNAME(CAGRP1) GRPMAX(15) NOREUSE -
GRPMEM((DB1,DD1) (DB2,DD2) (DB3,DD3) -
(DB4,DD4) (DB5,DD5) (DB6,DD6) (DB7,DD7) -
(DB8,DD8) (DB9,DD9) (DB10,DD10))
/*

INIT.DB

Use an INIT.DB command to register a database with DBRC and specify the level of database sharing. A database must be registered with DBRC before you can initialize a new DBDS, HALDB partition, or DEDB area with the INIT.DBDS command or INIT.PART.

When you issue an INIT.DB command for a HALDB partition (TYPE=PART), the following fields are set as defaults:

- OLREORG CURSOR ACTIVE =NO
- OLRIMSID=**NULL**
- ACTIVE DBDS=A-J

INIT.DB Syntax

![INIT.DB Syntax Diagram]
Parameters for INIT.DB

**DBD(name)**
Required parameter you use to specify the database name of the database to be registered in RECON.

**Restriction for HALDBs:** For the INIT.DB command, name specifies the name of a HALDB master database.

**DBTRACK | RCVTRACK**
Mutually exclusive, optional parameters that specify the type of tracking (shadowing) for a database that is assigned to a global service group.

**Restrictions:**
- Neither RCVTRACK nor DBTRACK can be specified without GSGNAME.
- Neither RCVTRACK nor DBTRACK can be specified if a TYPEFP is specified.
- Specifying DBTRACK has no effect if the tracking subsystem is a recovery readiness level (RLT) subsystem.

**DBTRACK**
Indicates database readiness tracking.

**RCVTRACK**
Indicates recovery readiness tracking.

**GSGNAME(gsname)**
Optional parameter used to specify to which global service group a database is to be assigned.

GSGNAME cannot be specified if TYPEFP is specified.

**ICREQ | NOICREQ**
Mutually exclusive, optional parameters used to specify whether DBRC should enforce the requirement of taking an image copy of a non-recoverable or user-recoverable database after it has been loaded or reorganized. This option affects all DBDSs within the DB.

**ICREQ**
Default value that turns on IC NEEDED in the DBDS record and increments the IMAGE COPY NEEDED COUNT in the DB record following initial load (PROCOPT=L) or REORG of the DBDS.

**NOICREQ**
Turns off the IC NEEDED requirement. DBRC will not turn on IC NEEDED in the DBDS record or increment the IMAGE COPY NEEDED COUNT in the DB record following initial load or REORG of the DBDS. The IC NEEDED flag can still be set manually with the CHANGE.DBDS ICON command.

**Requirement:** When NOICREQ is specified, NONRECOV or USERRCOV must also be specified.
NONRECOV | RECOVABL | USERRCOV
Mutually exclusive, optional parameters used to specify whether DBRC can record the updates in the RECON for the data sets of the specified databases.

NONRECOV
Specifies that no record of the updates for the data sets of the specified database are to be kept in the RECON.

RECOVABL
Specifies that the update allocations on the database are to be written in the RECON.

USERRCOV
Specifies that update allocations for the Fast Path database are not to be recorded in the RECON. Database updates are logged, however, DBRC will not be able to manage recovery for the database. Recovery is allowed with JCL manually generated by the user. This parameter is only valid for TYPEFP. USERRCOV cannot be specified for an RSR-covered database.

Notes:
1. USERRCOV should only be used if a critical need to reduce database open time exists and recoverability is require.
2. USERRCOV requires the user to provide their own manually-created JCL for change accumulation and recovery.

Restrictions:
• If the database is registered as RECOVABL or USERRCOV, VIO or DD DUMMY data sets cannot be used for the output log (IEFRDER) in any job that updates the database. Temporary log data sets, such as VIO data sets, are deleted at job termination, so they are not usable for recovery.
• NONRECOV or USERRCOV cannot be specified if GSGNAME is specified.
• You cannot make concurrent image copies of non-recoverable or user-recoverable databases.

SHARELVL(0 | 1 | 2 | 3)
Optional parameter you use to specify the level of data sharing for which authorized subsystems can share a database. For a description of the levels of data sharing, see “Assigning a Sharing Level with DBRC” on page 17.

Restriction:
• You must specify a SHARELVL of 1, 2, or 3 for concurrent image copies.
• If you are using IRLM, and have specified SHARELVL 2 or 3, ensure that the VSAM SHAREOPTIONS (3 3) parameter is also specified.

For more information on coordinating VSAM data set definitions with share options, see IMS V9: Administration Guide: System.

TYPEFP | TYPEIMS | TYPHALDB
Mutually exclusive, optional parameters that you use to specify whether the database is a Full Function, Fast Path DEDB, or a HALDB.

TYPEFP
Specifies that the database is Fast Path DEDB.

TYPEIMS
Specifies that the database is a DL/I database (non-HALDB).

TYPHALDB
Specifies that the database is a DL/I database (HALDB). When you issue a
INIT.DB

INIT.DB TYPHALDB command, the IMS DBDLIB data set must be identified in the job stream for the Database Recovery Control utility with a ddname of IMS.

OLRCAP | OLRNOCAP
Mutually exclusive, optional parameters used to indicate that HALDB OLR is allowed to run on that database.

OLRCAP Specifies in the master database record and in the partition database record that HALDB OLR is allowed to run on that database. This is the default (except for a PSINDEX, which is always OLRNOCAP).

OLRNOCAP Specifies in the master database record and in the partition database record that HALDB OLR is not allowed to run on that database.

HIKEY | PARTSEL (pgname)
Mutually exclusive, optional parameters that you use to identify whether the HALDB uses a Partition Selection Exit or high key values. This setting determines whether the KEYSTRING parameter on the INIT.PART command defines a partition selection exit string or a high key value. These parameters are valid only with TYPHALDB.

PARTSEL(pgname)
Specifies the name of the Partition Selection Exit program to be used for the HALDB. The pgname is specified as a value up to 8 characters long that is a valid program name.

HIKEY
Specifies that this HALDB uses high key values.

If both PARTSEL and HIKEY are omitted, the HALDB will use the Partition Selection Exit module name (PSNAME) from the HALDB DBD. If no Partition Selection Exit module name was specified in the DBD, the HALDB will use high key values.

Example of Creating a SHARELVL 1 DB Record
In this example, a new database record is created in RECON. This database has a share level of 1.

//INITDB JOB

; //SYSIN DD * INIT.DB DBD(THISDB) SHARELVL(1) TYPEFP
/*

INIT.DBDS
Use an INIT.DBDS command to register a DBDS or DEDB area. The DBDS must exist for any of the other commands to work for a given DBDS or DEDB area. In order to register the DBDS, DBRC examines the IMS DBDLIB data set to:
• Verify that the DBDS or DEDB area exists.
• Obtain the DBDS’s data set identifier (DSID), its data set organization, and its database organization.

The IMS DBDLIB data set must be identified in the job stream for the Database Recovery Control utility with a ddname of IMS.

The INIT.DBDS command fails if you issue it while the database is in use.
When you issue an INIT.DBDS command, DBRC sets the image copy status in the DBDS record in the RECON to IC RECOMMENDED. After registering the DBDS, you can reset the image copy status in the RECON by performing either of the two following actions:

- Use the Image Copy utility to create a backup copy.
- Use the CHANGE.DBDS command to change the image copy status in the RECON (see the explanations of the ICON and ICOFF parameters under "CHANGE.DBDS" on page 149 for more information).

**Restriction for HALDBs:** You must use the HALDB Partition Definition utility to update or delete information about HALDBs in the RECON data set. You cannot use the INIT.DBDS command to register DBDSs of HALDBs.

**INIT.DBDS Syntax**

```
INIT.DBDS DBD(name) DDN(name) AREA(name) DSN(name) GENMAX(value) DBTRACK RCVTRACK NOREUSE REUSE
```

```
DEFLTJCL(member) GSGNAME(gsgname) ICJCL(member) OICJCL(member) PREOPEN RECOVJCL(member) NOREUSE
```

```
RECOVPD(value) RECVJCL(member) NOVSO VSO CFSTR1(name) CFSTR2(name) LKASID MAS NOPREL
```

**Parameters for INIT.DBDS**

**DBD(name)**
Required parameter you use to specify the database name of the DBDS or DEDB area being identified to DBRC.

**DDN(name) | AREA(name)**
Mutually exclusive, required parameters that you use to specify the ddname of the DBDS or DEDB area being identified to DBRC.

**DSN(name)**
Required parameter you use with the DDN(name) parameter to specify the data set name of the DBDS being identified to DBRC. You cannot use this parameter with a DEDB area.

If the DBDS is an ADS that is registered to DBRC, do not specify this parameter. Instead specify the data set name in the INIT.ADS command.

**GENMAX(value)**
Required parameter you use to specify the maximum number of image copies that DBRC is to maintain for the identified DBDS.
Each time you run the Database Image Copy utility for the DBDS that is being initiated, a new image copy is defined in RECON. If you identified the DBDS with the NOREUSE parameter, the oldest image copy for the DBDS beyond the recovery period is deleted when the number of image copies exceeds the GENMAX value. If you identify the DBDS with the REUSE parameter, the oldest image copy beyond the recovery period is reused.

*value* must be a decimal number from 2 to 255.

**DBTRACK | RCVTRACK**

Mutually exclusive, optional parameters that you use to specify the type of RSR tracking (shadowing) for an area that is assigned to a global service group. Neither RCVTRACK nor DBTRACK can be specified without GSGNAME and AREA.

Specifying DBTRACK has no effect if the tracking subsystem is a recovery-readiness level (RLT) subsystem.

**DBTRACK**

Indicates database-readiness tracking.

**RCVTRACK**

Indicates recovery-readiness tracking.

**DEFLTJCL(member)**

Optional parameter you use to specify an implicit skeletal JCL default member for the DBDS. The specified member is used by the GENJCL.IC, GENJCL.01C, and GENJCL.RECOV commands in order to resolve keywords you have defined.

**GSGNAME(gsname)**

Optional parameter used to specify to which global service group a database is to be assigned.

GSGNAME can only be specified if AREA is specified.

**ICJCL(ICJCL | member)**

Optional parameter you use to specify the name of a member of a partitioned data set that contains skeletal JCL. When you issue a GENJCL.IC command, DBRC uses this member to generate the JCL to run the Database Image Copy utility for the DBDS or DEDB area being identified.

**NOREUSE | REUSE**

Mutually exclusive, optional parameters that you use to specify whether the supported image copy utilities are to reuse previously used image copy data sets.

**REUSE**

Allows the GENJCL.IC command or the GENJCL.01C command to generate a job that causes the supported image copy utilities to reuse the oldest image copy data set (for the DBDS being defined) when the GENMAX value for the DBDS is exceeded. REUSE requires that you create empty image copy data sets for future use by the supported image copy utilities. In addition, you must use INIT.IC commands to record their existence in RECON. The NOREUSE parameter prohibits such actions.

**NOREUSE**

Prevents the automatic reuse of image copy data sets for this DBDS by the supported image copy utilities.

If the NOREUSE option is specified for the HISAM database, the image-copy-needed flag is not turned on at the end of the HISAM Reload
utilities. The input data set that is used while the HISAM database is being reloaded is used as an image copy data set.

If you want HSSP image copy processing, you must specify REUSE. Reuse means that the image copy job uses the same volumes, data set name, and record in RECON for the new image copy data set as those of the oldest DBDS image copy data set.

**OICJCL(RECVJCL | member)**
Optional parameter you use to specify the name of a member of a partitioned data set that contains skeletal JCL. You cannot use this parameter with a DEDB area. When you issue a GENJCL.REC command, DBRC uses this member to generate the JCL to run the Online Database Image Copy utility for the DBDS being identified.

**PREOPEN | NOPREO**
Mutually exclusive, optional parameters that you use to specify whether an area is to be opened after the first checkpoint following the next control region initialization or when the next /START AREA command is processed. NOPREO is the default, except if you specify PRELOAD, in which case PREOPEN is the default.

**PREOPEN**
Indicates that the area is to be opened the next time the control region is started or a /START AREA command is processed. This option is valid for both VSO and non-VSO areas.

**NOPREO**
Indicates that the area is not to be pre-opened the next time the control region is started or a /START AREA command is processed. You cannot specify NOPREO with PRELOAD.

**RECOVPD( | value)**
Optional parameter you use to specify the recovery period for a specified DBDS or DEDB area.

The recovery period is the amount of time before the current date for which DBRC maintains recovery information in the RECON data set. For example, if the recovery period of a DBDS or area is 14 days, DBRC maintains sufficient recovery-generation information for at least 14 days.

The recovery period is calculated as the current date minus the date of the oldest image copy. If the image copies are dated within the days specified in the RECOVPD(value), DBRC keeps them in the RECON.

For value, specify a decimal number from 0 to 999 that represent the number of days the image copies are to be kept in RECON. If you specify 0 (the default), there is no recovery period.

**Related Reading:** See [IMS V9: Operations Guide](#) for more information about the recovery period of image copy data sets.

**RECVJCL(RECVJCL | member)**
Optional parameter you use to specify the name of the skeletal JCL member to be used by the GENJCL.RECEIVE command.
**RECVJCL** can be specified for both RSR-covered and non-covered DL/I DBDSs and Fast Path areas.

**VSO | NOVSO**

Mutually exclusive, optional parameters that you use to specify whether an area is to reside in virtual storage the next time the control region is initialized or when the next `/STA AREA` command is processed.

**VSO**

Indicates that the area is to reside in virtual storage. Areas that are defined with SHARELVL(0 | 1) are read into and written from a z/OS data space. Areas defined with SHARELVL(2 | 3) use the coupling facility to share data between connected subsystems.

**CFSTR1(name)**

Optional parameter you use to specify the name of the first coupling facility structure for the identified area. z/OS coupling facility structure naming conventions must be adhered to. This parameter is valid only for VSO areas of DEDBs that are defined with SHARELVL(2 | 3). The area name is the default if VSO is specified and the DEDB isSHARELVL(2 | 3).

**CFSTR2(name)**

Optional parameter you use to specify the name of the second coupling facility structure for the identified area. z/OS coupling facility structure naming conventions must be adhered to. This parameter is valid only for VSO area of DEDBs defined with SHARELVL(2 | 3). There is no default.

**Restrictions:** The name cannot be the area name if the CFSTR1 keyword is not specified. CFSTR2 cannot be specified if MAS is specified.

**Related Reading:** See [IMS V9: Administration Guide: Database Manager](#) for details on CFSTR (coupling facility structure) naming conventions.

**LKASID | NOLKASID**

Mutually exclusive optional parameters that you use to specify whether local data caching for the specified area is to be used for buffer lookaside on read requests. The LKASID option is valid only for SHARELVL(2 | 3) VSO areas.

**LKASID**

Indicates that buffer lookaside is to be performed on read requests for this area.

**NOLKASID**

Indicates that buffer lookaside is not to be performed on read requests for this area.

**MAS | NOMAS**

Mutually exclusive optional parameters that you use to specify whether a shared VSO is to reside in a multi-area coupling facility structure.

**MAS**

Indicates that the area is to reside in a multi-area coupling facility structure. If MAS is specified, a second coupling facility structure (CFSTR2) cannot be specified.
NOMAS
Indicates that the coupling facility structure (or structures) should contain data only for this area. NOMAS is the default.

PRELOAD | NOPREL
Mutually exclusive, optional parameters that you use to specify whether a VSO area is to be loaded the next time it is opened.

PRELOAD
Indicates that the area is to be loaded into a data space or coupling facility structure the next time that it is opened. Selecting this option also causes the area to be pre-opened.

NOPREL
Indicates that the area is not to be loaded into a data space or coupling facility structure the next time that it is opened. For VSO areas, CIs are copied into a data space when they are read for the first time.

NOVSO
Indicates that this area is not to reside in virtual storage.

Example of Identifying the DBDS to Initiate DBRC’s Control Over Recovery
In this example, a DBDS is registered with DBRC. The IMS DD statement is required to allow access to the IMS DBDLIB data set to obtain the data set identifier, data set organization, and database organization of the DBDS. The DBDS is identified by the DBD, DDN, and DSN parameters and is accessed only by an IMS system. Image copy data sets (of the DBDS) can be reused and DBRC maintains a maximum number of 2 image copies. The ICJCL parameter specifies the member of the partitioned data set of skeletal JCL that is to be used for the generation of JCL for the Database Image Copy utility. The RECOVJCL parameter does the same for the Database Recovery utility.

//INITDBDS JOB
:
   //IMS DD DSN=IMS.DBDLIB,DISP=SHR
   //SYSIN DD *
      INIT.DBDS DBD(DBD002) DDN(DDN003) GENMAX(2) REUSE -ICJCL(ICJCLX) RECOVJCL(RECOVJCX) DSN(DSN003)
   /*

INIT.DBDSGRP
Use an INIT.DBDSGRP command to define a group of these types:

- DBDS group (DBDSs or DEDB areas)
- DB group (DL/I databases or DEDB areas)
- Recovery group (DL/I databases or DEDB areas)

A DBDS group can be used anywhere that a DB group can be used, such as for the /DBR command, but this usage is inefficient. Define a separate DB group for such use.

A recovery group is used with database recoveries. It can also be used anywhere that a DB group can be used.
INIT.DBDSGRP

INIT.DBDSGRP Syntax

```
INIT.DBDSGRP
  GRPNAME(name)
  MEMBERS(dbname,ddname)
  DBGRP(dbname)
  RECOVGRP(dbname,areaname)
```

Parameters for INIT.DBDSGRP

**GRPNAME(name)**

Required parameter you use to identify the DBDSGRP to be created. The name can be from one to eight alphanumeric characters, and must not be the name of an existing DBDSGRP or CAGRP record.

**MEMBERS(dbname,ddname) | DBGRP(dbname) | RECOVGRP(dbname,areaname)**

Mutually exclusive, required parameters that identify the members to be included in the new group. A group can contain up to 32767 members.

**MEMBERS(dbname,ddname)**

Indicates that the group is a DBDS group. This parameter identifies one or more DBDSs or DEDB areas, each by a pair of names enclosed in parentheses, where `dbname` is the database name and `ddname` is the DD statement name or the DEDB area name.

Any member can belong to more than one DBDS group.

**DBGRP(dbname)**

Indicates that the group is a DB group, and identifies one or more databases or area names. Any member can belong to more than one DB group.

**RECOVGRP(dbname,areaname)**

Indicates that the group is a recovery group. A recovery group is a group of full-function databases, HALDB databases, or DEDB areas that you consider to be related. HALDB partitions and Fast Path databases cannot belong to a recovery group. If you perform a time-stamp recovery on one of the members of the group, some database recovery tools require you to recover all members of the group to the same time. A recovery group otherwise can be used like a DB group.

RECOVGRP identifies one or more DBs or DEDB areas. If a DEDB area is to be added to the recovery group, both `dbname` and `areaname` must be specified. Otherwise, `areaname` must not be specified. A database or area can belong to only one recovery group. If any of the members specified by RECOVGRP already belongs to another recovery group, the command fails.

**Restrictions for HALDBs:** For HALDBs, `dbname` specifies the name of a HALDB partition or the HALDB master database. When specifying HALDB database names, use the INIT.DBOSGRP command only as defined in Table 13 on page 273.
Table 13. Supported dbname specifications for HALDBs on INIT.DBDSGRP command parameters

<table>
<thead>
<tr>
<th>INIT.DBDSGRP</th>
<th>dbname = HALDB</th>
<th>dbname = Partition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMBERS</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>DBGROUP</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>RECOVGRP</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Example of Creating a Group of DBDSs
In this example, a group of DBDSs is defined.

//INITDBGROUP JOB
:
//SYSIN DD *
INIT.DBDSGRP GRPNAME(DBDSG1) -
MEMBERS((DB1,DD1),(DB2,DD2),(DB3,DD3))
/*

INIT.GSG
Use an INIT.GSG command to define a global service group (GSG). The GSG must be defined in every RECON which is to be used by any IMS subsystem in the GSG.

This command fails if RSRFEAT=NO is specified in the IMSCTRL macro.

INIT.GSG Syntax

```
INIT.GSG GSGNAME(gsname)
    [SEQNUM(number)]
```

Parameters for INIT.GSG

GSGNAME(gsname)
   Required parameter you use to specify the name of the GSG you want to create.

SEQNUM(number)
   Optional parameter you use to specify the initial DSN sequence number for the GSG you want to create. If you do not specify a SEQNUM parameter, the GSG DSN SEQ NUMBER is set to zero (0). This value is used to create unique tracking log data set names. If you have deleted an old GSG and are now creating a new GSG with the same name, specify a SEQNUM equal to the value of the last DSN SEQ NUMBER of the old GSG. Otherwise, the tracker might create logs that have data set duplicate names of previously created logs.

Example of Creating a Global Service Group
Here is an example of using the INIT.GSG command.

//INITGSG JOB
:
//SYSIN DD *
INIT.GSG GSGNAME(IMSGSG1)
/*
INIT.IC

Use an INIT.IC command to create image copy records in RECON. These image copy records define image copy data sets that are available for use during subsequent runs of the supported image copy utilities.

Each INIT.IC command creates one image copy record. You can define image copy data sets for subsequent use only if you have specified a REUSE parameter for the corresponding DBDS or DEDB area when it was identified in RECON with an INIT.DBDS command. The maximum number of image copy records that are to be used for a given DBDS or DEDB area is determined by the value of GENMAX for the specified DBDS or DEDB area.

INIT.IC Syntax

```
/SM590000/SM590000
INIT.IC DBD(name) DDN(name) AREA(name) ICDSN(name)
FILESEQ(value)

/SM590000
1
FILESEQ2(value)
ICDSN2(name)
3400
UNIT(unittype)

/SM590000/SM630000
3400
UNIT2(unittype)

/SV040000
VOLLIST(volser)

/SV040000
VOLLIST2(volser)
```

Parameters for INIT.IC

**DBD(name)**
Required parameter you use to identify the image copy data set being created by the database name of its related DBDS or DEDB area.

**Restriction for HALDBs**: For the INIT.IC command, *name* specifies the name of a HALDB partition.

**DDN(name) | AREA(name)**
Mutually exclusive, required parameters that you use to identify the image copy data set being created; *name* is the data set ddname of the related DBDS or DEDB area.

**ICDSN(name)**
Required parameter you use to specify the name of the image copy data set for which the image copy record is being created. *name* can be up to 44 characters. You can use the default-naming convention for image copy data sets for this name.

**FILESEQ(1 | value)**
Optional parameter you use to specify the file sequence number of the image copy data set for which the image copy record is being created. You can specify this parameter only if you specify a VOLLIST parameter, and only if the file sequence number is not 1. *value* must be a decimal number from 1 to 9999.

**FILESEQ2(1 | value)**
Optional parameter you use to specify the file-sequence number of the duplicate image copy data set for which the image copy record is being created. You can specify this parameter only if you are creating a duplicate image copy data set, if you specify a VOLLIST2 parameter, and if the
file-sequence number is not 1. The value you substitute in the variable field must be a decimal number from 1 to 9999.

**ICDNS2**(name)

Optional parameter you use to specify the name of the duplicate image copy data set for which the image copy record is being created. name can be up to 44 characters. You can use the default naming convention for duplicate image copy data sets for this name.

**UNIT**(3400 | unittype)

Optional parameter you use to specify the unit type of the image copy data set being defined. The unit type can be up to eight alphanumeric characters long.

If you specify the UNIT parameter, you must also specify the VOLLIST parameter.

**UNIT2**(3400 | unittype)

Optional parameter you use to specify the unit type of the duplicate image copy data set being defined. The unit type can be up to eight alphanumeric characters.

**VOLLIST**(volser)

Optional parameter you use to specify the list of volumes on which the image copy data set resides when it is used by the supported image copy utilities. Each volume serial number you substitute in the variable field can be up to six alphanumeric characters. The volume serial list can contain from 1 to 255 volume serial numbers.

**VOLLIST2**(volser)

Optional parameter you use to specify the list of volumes on which the duplicate image copy data set (identified with the ICDNS2 parameter) resides when it is used by the supported image copy utilities. Each volume serial number you substitute in the variable field can be up to six alphanumeric characters. The volume serial list can contain from 1 to 255 volume serial numbers.

**Example of Creating a Record That Defines the ICDSN**

In this example, a record is created in RECON that defines an image copy data set that is to be used for creating an image copy of the DBDS. The name of the image copy data set is specified in the ICDSN parameter; in this example, the default naming convention is used to generate the fully qualified data set name. The volume on which the image copy data set is to reside is specified by the VOLLIST parameter, and its file-sequence number is specified by the FILESEQ parameter.

```plaintext
//INITIC  JOB
:
//SYSIN  DD  *
INIIT.IC  DBD(DB1) DDN(DD1) -
       ICDSN(IMS.*.ICDNS2) -
       VOLLIST(VOL003) FILESEQ(5)
/*

**INIT.PART**

Use an INIT.PART command to register a HALDB Partition. The INIT.PART command creates the RECON HALDB partition structure (a PART record, the partition DB record and one or more DBDS records according to the DBD specification). The INIT.PART command will fail if the HALDB is being used by the HALDB Partition Definition utility. The IMS DBDLIB data set must be identified in the job stream for the Recovery Control utility with a ddname of IMS.
Some parameters (identified below in the parameter description) apply to all the partition DBDSs created as a result of this command. This differs from the HALDB Partition Definition utility where these parameters may be specified separately for each partition DBDS being created. These parameters can later be changed individually with the CHANGE.DBDS command.

Related Reading: See [IMS V9: Administration Guide: Database Manager, SC18-7806](#) for information on designing databases, especially the following chapters:
- "Designing Databases"
- "Naming Conventions"

**INIT.PART Syntax**

```
| INIT.PART — DBD(name) — PART(name) — KEYSTRING — string — hex_string |
| DSNPREFIX(string) — RANDOMZR(name) — ANCHOR(value) — HIBLOCK(value) |
| BYTES(value) — FBFF — FSPF — |
| BLOCKSZE — 4096 — value — GENMAX(value) |
| DEFLTJCL(member) — ICJCL(member) — ICJCL — NOMVDBDS — MVDBDS — |
| OICJCL(member) — OICJCL — REUSE — RECOVJCL(member) — RECOVPD(value) — |
| RECVJCL(member) — RECVJCL — |
```

**Parameters for INIT.PART**

**DBD** *(name)*
- Required parameter used to identify the HALDB for which the partition is to be defined.

**PART** *(name)*
- Required parameter used to identify a HALDB partition name. Specified as an alphanumeric value, up to 7 characters long, with the first character being alphabetic.
KEYSTRING *(char value or hex value)*
Optional parameter you use to specify a HALDB partition high key value or a selection string for use by a partition selection exit. Specified as a character value up to 256 characters long or a hexadecimal value up to 512 characters long. Character values must be alphanumeric (with no embedded blanks or commas unless the string is enclosed by single quotes). Unless enclosed by single quotes, the character string will be folded to uppercase. Hexadecimal values must be enclosed by single quotes and preceded by the letter X, for example: KEYSTRING(X’D7C1D9E3D2C5E8’).

If no partition selection routine was specified in the HALDB master definition, KEYSTRING defines the Partition high key and is required. The high key length cannot be longer than the root key length. If the high key length is less than the defined root key length, the high key value is padded with X’FF’s up to the defined root key length. The partition high key values must be unique for each partition within a HALDB.

If a partition selection routine was specified in the HALDB master definition, KEYSTRING defines a Partition Selection String which is passed to the partition selection routine. Your installation partition selection routine may or may not require a Partition Selection String. If required, the content of the string is determined by your installation. It can be up to 256 bytes long and consist of simple character information. If it contains non-printable characters, it must be identified using hex notation. A hex character string is enclosed by single quotation marks and prefixed with an X.

DSNPREFIX *(string)*
Required parameter you use to specify the data set name prefix for the partition data sets contained in a HALDB. Specified as a value, up to 37 characters long, that is a valid JCL data set name.

RANDOMZR *(name)*
Optional parameter used to specify the name of the randomizing module for HALDB PHDAM databases only. If RANDOMZR is omitted, the name of the randomizing module is obtained from the DBD. A randomizing module controls root segment placement in, or retrieval from, the PHDAM HALDB.

ANCHOR *(value)*
Optional parameter used to specify the number of root anchor points (RAPs) desired in each control interval or block in the root addressable area of a PHDAM HALDB. The value specified must be between 1 and 255. Typical values are from 1 to 5. If ANCHOR is omitted, the value is obtained from the DBD. This parameter is for PHDAM HALDBs only.

HIBLOCK *(value)*
Optional parameter used to specify the maximum relative block number value that the user wishes to allow a randomizing module to produce for this HALDB. This value determines the number of control intervals or blocks in the root addressable area of an PHDAM HALDB. The value may range between 1 and 16,777,215 (2**24-1). If BYTES is omitted, the value is obtained from the DBD. This parameter is for PHDAM HALDBs only.

BYTES *(value)*
Optional parameter used to specify the maximum number of bytes of a HALDB database record that can be stored into the root addressable area in a series of inserts unbroken by a call to another HALDB database record. The value can range between 0 and 16,777,215 (2**24-1). If BYTES is omitted, the value is obtained from the DBD. This parameter is for PHDAM HALDBs only.
A value of 0 (zero) means that all bytes are addressable. Setting BYTES to 0 is equivalent to omitting BYTES on the DBD definition.

**FBFF (0| value)**
Optional parameter used to specify the free block frequency factor (fbff) which specifies that every nth control interval or block in this data set group is left as free space during database load or reorganization (where FBFF=n). The range of FBFF includes all integer values from 0 to 100 except 1. The default value for FBFF is 0.

**FSPF (0| value)**
Optional parameter used to specify the free space percentage factor. It specifies the minimum percentage of each control interval or block that is to be left as free space in this data set group. Value may be any number between 0 and 99. The default value for FSPF is 0.

**BLOCKSZE (4096| nnnnn)**
Optional parameter you use to specify the block size for OSAM data sets. Specify an even number no greater than 32,766. The block size value is used for OSAM only. The default is 4096. You may specify up to 10 values, one for each data set group defined in the DBD. See the SIZE keyword on the DATASET statement in the chapter on Database Description (DBD) Generation in the [IMS V9: Utilities Reference: Database and Transaction Manager](https://www.ibm.com) for further information on specifying the block size for OSAM data sets (although DBDGEN is not used to define HALDB partitions).

**GENMAX (2| value)**
Optional parameter you use to specify the maximum number of image copies that DBRC is to maintain for the partition DBDSs. If you identify a partition DBDS with the NOREUSE parameter, the oldest image copy beyond the recovery period is deleted when the number of image copies exceeds the GENMAX value. If you identify it with the REUSE parameter, the oldest image copy beyond the recovery period is reused. Specified as a numeric value from 2 to 255. All partition DBDSs will be created with this GENMAX value. The CHANGE.DBDS command can be used to change this for individual partition DBDSs. The default value for GENMAX is 2.

**DEFLTJCL (member)**
Optional parameter you use to specify an implicit skeletal JCL default member for a HALDB Partition DBDS. The specified member is used by the GENJCL.IC, GENJCL.OIC, and GENJCL.RECOV commands in order to resolve keywords you have defined. All partition DBDSs will be created with this DEFLTJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

**ICJCL (ICJCL | member)**
Optional parameter you use to specify the name of a member of a partitioned data set that contains skeletal JCL. When you issue a GENJCL.IC command, DBRC uses this member to generate the JCL to run the Database Image Copy utility (or the Database Image Copy 2 utility) for the partition DBDS specified on the GENJCL command. All partition DBDSs will be created with this ICJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

**NOREUSE | REUSE**
Mutually exclusive, optional parameters that you use to specify whether the supported image copy utilities are to reuse previously used image copy data sets.
NOREUSE

NOREUSE allows the GENJCL.IC command or the GENJCL.OIC command to generate a job that causes the supported image copy utilities to reuse the oldest image copy data set (for the DBDS specified on the GENJCL command) when the GENMAX value for it is exceeded. REUSE requires that you create empty image copy data sets for future use by the supported image copy utilities. In addition you must use an INIT.IC command to record their existence in RECON.

NOREUSE

NOREUSE parameter prohibits such actions. All partition DBDSs will be created with the parameter specified. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

MVDBDS | NOMVDBDS

Optional parameters used to allow the partner DBDSs (M-V,Y) to be created in RECON for HALDBs capable of running HALDB Online Reorganization.

MVSDBDS

Optional parameter used to create the M-V, Y data sets if the HALDB is defined as OLRCAP. If the parameter is not included on an INIT.0DBS command, the M-V, Y data sets are not created until HALDB Online Reorganization is run the first time.

NOMVDBDS

Optional parameter to indicate the M-V, Y data sets are not to be created in RECON for HALDBs defined as OLRCAP. NOMVDBDS is the default for HALDBs that are OLRCAP.

OICJCL (OICJCL { member})

Optional parameter you use to specify the name of a member of a partitioned data set that contains skeletal JCL. When you issue a GENJCL.OIC command, DBRC uses this member to generate the JCL to run the Online Database Image Copy utility for the partition DBDS specified on the GENJCL command. All partition DBDSs will be created with this OICJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

RECOVJCL (RECOVJCL { member})

Optional parameter you use to specify the name of a member of a partitioned data set that contains skeletal JCL. When you issue the GENJCL.RECOV command, DBRC uses this member to generate the JCL to run the Database Recovery utility for the partition DBDS specified on the GENJCL command. All partition DBDSs will be created with this RECOVJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

RECOVPD (01 value)

Optional parameter you use to specify the recovery period for a specified partition DBDS. Specify a numeric value from 0 to 999 that represents the number of days you want DBRC to maintain sufficient recovery-generation information in the RECON data set. The default is 0 which means there is no recovery period. All partition DBDSs will be created with this RECOVPD value. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

RECVJCL(RECVJCL { member})

Optional parameter you use to specify the name of the skeletal JCL member to be used by the GENJCL.RECEIVE command. RECVJCL can be specified for both RSR-covered and non-covered HALDB DBDSs. All partition DBDSs will
be created with this RECVJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

**Example of Creating a Partition**

This example creates a partition for the HALDB identified by the DBD parameter. The partition name is identified by the PART parameter. The HALDB uses a partition selection exit. The KEYSTRING parameter defines the selection string used by the Partition Selection exit. The string is 149 characters long. The character string must be enclosed in quotes because it contains blanks. The ending minus sign ("-"") indicates the line is continued. Note that the selection string also contains minus signs, so the first line of the string ends with two minus signs (the first, being part of the selection string, and the second, being the continuation character). The first DSGROUP (DBDS) is assigned a block size of 32000, the second is defaulted to 4096, the third is 4000, and the remaining DSGROUPs will all have the default of 4096. The partition is marked as needing partition initialization when it is created.

```
//INITPART JOB

//IMS DD DSN=IMS.OBDLIB,DISP=SHR
//SYSIN DD *
INIT.PART DBD(DBHDOK01) PART(PDHDOKE) KEYSTRING('ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789--

ABCDEFH'IJKL--

MNPQ'RSTUVWXYZ0123456789 ABCDEFH'IJKLMNPQ'RSTUVWXYZ0123456789--

ABCD--

EFGHIJKLMNOPQRSTUVWXYZ0123456789--

DSNPREFIX(TESTSYS1.PROJECT) -

BLOCKSZE(32000,,4000)

 INIT.RECON
```

**INIT.RECON**

Use the INIT.RECON command to initialize the RECON for use by DBRC.

The RECON data sets must first be created using the AMS DEFINE CLUSTER command, and must be empty.

**INIT.RECON Syntax**

```
INIT.RECON NOCATDS CMDAUTH(NONE)

CATDS CMDAUTH(SAF,safhlq)

EXIT BOTH

NOFORCER FORCER CHECK17 CHECK44 NOCHECK

LISTDL NOLISTDL

LOGRET(001) LOGRET('time_interval')

MINVERS('7.1') MINVERS('8.1') MINVERS('9.1')

REORGV NOREORGV
```
Parameters for INIT.RECON

**CATDS | NOCATDS**

Mutually exclusive, optional parameters that you use to indicate whether image copy, change accumulation, and log data sets are cataloged.

**CATDS**

Specifies that these data sets are cataloged. If the data set is allocated by the catalog and the CATDS option is used, DBRC bypasses volume serial and file sequence verification for the data set.

In order for the CATDS option to be effective, the data set must be cataloged, and volume serial information for the data set must be omitted from the JCL. If the data set is cataloged, CATDS is specified, and volume serial information is included in the JCL, DBRC ignores CATDS and allocates the data set by the JCL. Normal volume serial and file sequence checking occurs.

If the data set is not cataloged, CATDS is not effective, and DBRC allocates the data set by the JCL, with volume serial and file sequence checking.

**Attention:** The CATDS option affects restart of IMS from SLDS data sets. Because the CATDS option indicates the SLDSs are under the control of a catalog management system, the VOLSER is not passed back to IMS for data set allocation. If the SLDS data sets are not cataloged, IMS restart fails.

**NOCATDS**

Specifies that these data sets, regardless if they are cataloged, are not to be treated as cataloged. DBRC verifies that the volume serial and file sequence numbers appearing in the job file control block are the same as the information recorded in the RECON.

**CMDAUTH(SAF | EXIT | BOTH | NONE,safhlq)**

Optional parameter you use to specify whether DBRC batch command authorization is to be performed.

Specification of this parameter will result in command authorization being performed using the specified setting.

**SAF | EXIT | BOTH | NONE**

Mutually exclusive subparameters that you use to specify your DBRC batch command security option.

**SAF**

Specifies that a security product is to be called for command authorization.

**EXIT**

Specifies that the DBRC Command Authorization Exit (DSPDCAX0) is to be called for command authorization.

**BOTH**

Specifies that a security product and the DBRC Command Authorization Exit (DSPDCAX0) are to be called for command authorization. The security product is called first, then DSPDCAX0.
NONE
Specifies that command authorization is to be skipped. This is the default.

safhlq
High level qualifier of the resource name used to determine command authorization. safhlq must be specified with SAF, EXIT, or BOTH. safhlq is a name up to eight alphanumeric characters in length.

DASDUNIT(unittype)
Optional parameter you use to specify the unit type of the DASD device holding the records for log data sets. The unit type can be up to eight alphanumeric characters long.

If you do not use this parameter to specify a DASD device, then the INIT.RECON command defaults to DASD unit type 3400.

FORCER | NOFORCER
Mutually exclusive, optional parameters that you use to specify whether all IMS databases must be registered in RECON.

FORCER
Specifies that all databases must be registered. If you specify FORCER and a job attempts to access an unregistered database, the database authorization call from IMS to DBRC fails.

NOFORCER
Specifies that databases do not have to be registered.

CHECK17 | CHECK44 | NOCHECK
Mutually exclusive, optional parameters that you use to change the type of log data set name comparison that DBRC does.

CHECK17
Verifies that the last 17 characters of a log data set name are consistent with RECON. If the name in RECON does not match the names on the appropriate ddname, the utility stops.

CHECK44
Verifies that the 44-character log data set name is consistent with RECON. If the name in RECON does not match the name on the appropriate log ddname, the utility stops.

NOCHECK
Used if the data set name specified as input to the Database Recovery utility is longer than 17 characters and has a new high-level qualifier. DBRC does not compare the log data set name recorded in RECON with the name on the appropriate ddname.

LOGRET(time_interval)
Optional parameter you use to specify the retention period for log data sets.

Definitions:
• The retention period is the minimum amount of time in which a log becomes inactive after it is opened. (It is then eligible to be deleted.)
• The time_interval is a partial, punctuated time stamp representing a time interval (days, hours, minutes, and seconds) rather than a date and time. The time stamp has the following format:
  
  ddd|hh|mm|ss|t

  ddd          Number of days (000 to 365)
hh Number of hours (0 to 23)
mm Number of minutes (0 to 59)
ss Number of seconds (0 to 59)
t Tenths of a second (0 to 9)

The punctuation for the time stamp (shown in the above format as a vertical bar (|)) can be any non-numeric character, such as a period (.) or a comma (,). The time stamp must be enclosed in single quotes (’) if it contains any blanks or special characters. The number of days must include any leading zeros, but you can omit trailing zeros. Valid intervals range from a tenth of a second to 365 days. The default value, 001, is 24 hours.

Because the time interval is treated as a time stamp, message DSP0106I can be issued for incorrect values. Some examples of valid time intervals include:

LOGRET(365)
LOGRET(’030 12.00’)  
LOGRET(’000 00:00:08.0’)  
LOGRET(’000 00,00,00,1’)

Two different valid formats for equivalent time stamp specifications are shown.

LOGRET(030) LOGRET(’030’) = 30 days
LOGRET(’010 12,30’) LOGRET(’10 12:30’) = 10 days, 12 hours, 30 minutes

If you do not use this parameter to specify a retention period, then the INIT.RECON command defaults to a period of 001 (24 hours).

Related Reading:
- See “DELETE.LOG (for RLDS and SLDS)” on page 208 for more information on deleting inactive logs.
- See “Standard Time Stamp Format” on page 124 for more information on time stamps.

LISTDL | NLISTDL

Mutually exclusive, optional parameters that you use to specify whether data set names deleted from the RECON (by the DELETE.LOG command or by an archive job log compression) are listed in the job output. The setting specified on this command can be overridden by the DELETE.LOG command. There is no way to override the setting for log compression during an archive job.

LISTDL

Specifies that names of deleted data sets are to be listed in the job output.

NLISTDL

Specifies that names of deleted data sets are not to be listed in the job output.

MINVERS(‘7.1’ | ’8.1’ | ’9.1’)

Optional parameter used to specify the lowest version of IMS allowed to sign on. This keyword cannot be set to a higher IMS release if any subsystems with a lower version of IMS are currently signed on to DBRC. The default is IMS ‘7.1’.

Note: The IMS Version 9 format for the MINVERS keyword includes a period. DBRC considers this a special character and requires it to be enclosed in single quotes. The format from earlier versions (71 | 81 | 91) is also accepted for compatibility.
REORGV | NOREORGV
Mutually exclusive, optional parameters you use to modify the HALDB partition reorganization number verification. A CHANGE command with these parameters is allowed while databases are currently authorized to other subsystems.

REORGV
Specifies that HALDB partition reorganization number verification is enabled. The RECON will store the reorganization number for each partition to prevent the reorganization number from being regressed, or incremented by more than one. After a CHANGE.RECON REORGV command is issued, the partition reorganization number in each partition record is set to zeros.

Note: The MINVERS value must be a minimum of 8.1 in order to set REORGV.

NOREORGV
Specifies that HALDB partition reorganization number verification is disabled. Any requests to store, update or verify the reorganization number will be ignored.

SSID(name)
Optional parameter you use to specify the name of the IMS subsystem that is to be used as the default subsystem ID for the following commands:

CHANGE.PRILOG
CHANGE.SECLOG
DELETE.LOG
GENJCL.ARCHIVE
GENJCL.CLOSE
NOTIFY.PRILOG
NOTIFY.SECLOG

name is an eight-character alphanumeric string that identifies a valid IMS subsystem ID.

STARTNEW | NONEW
Mutually exclusive, optional parameters that you use to specify whether new jobs are to be started when only one RECON is available.

STARTNEW
Specifies that new jobs are to be started.

NONEW
Specifies that new jobs are not to be started.

TAPEUNIT(unittype)
Optional parameter you use to specify the unit type of the tape device that is holding the records for log data sets. The unit type can be up to eight alphanumeric characters long.

If you do not use this parameter to specify a tape device, then the INIT.RECON command defaults to unit type 3400.

Example of Initializing the RECON
In this example, the RECON data sets are identified by the RECON1 and RECON2 DD statements.
INIT.RECON

//INITRCON JOB

: 

//RECON1 DD DSN=RECON7,DISP=SHR
//RECON2 DD DSN=RECON8,DISP=SHR
//SYSIN DD *

    INIT.RECON NOCHECK SSID('IMS8) LOGRET('007 00:00:30.0')

/*

INIT.SG

Use an INIT.SG command to define a service group as a member of a GSG. Every service group in the GSG must be defined in every RECON that is to be used by any IMS subsystem in the GSG. This command also specifies the initial role of the service group.

This command fails if RSRFEAT=NO is specified in the IMSCTRL macro.

INIT.SG Syntax

GSGNAME(gsgname) SGNAME(sgname) ACTIVE TRACKING LOCAL NONLOCAL

Parameters for INIT.SG

ACTIVE | TRACKING
Mutually exclusive, required parameters that you use to specify the initial role of the service group.

ACTIVE
Indicates that the service group is an active subsystem. ACTIVE can only be specified for one service group of a GSG.

TRACKING
Indicates that the service group is a tracking subsystem. TRACKING can only be specified for one service group of a GSG.

GSGNAME(gsgname)
Required parameter you use to specify the name of the GSG to which the service group belongs.

SGNAME(sgname)
Required parameter you use to specify the name of the service group you want to create.

LOCAL | NONLOCAL
Mutually exclusive, optional parameters that you use to specify whether the service group is local or nonlocal.

LOCAL
Indicates that this is the local service group for this set of RECONs.

NONLOCAL
Indicates that this is the nonlocal service group for this set of RECONs.

Examples of Creating Service Groups

In this example, the ACTIVE Service Group named STLSITE1 is added to the GSG IMSGSG1, and a LOCAL TRACKING SG named STLSITE2 is added to the same GSG.
//INITSG  JOB
:
//SYSIN DD *
INIT.SG  GSGNAME(IMSGSG1)  SGENAME(STLSITE1)  ACTIVE
INIT.SG  GSGNAME(IMSGSG1)  SGENAME(STLSITE2)  TRACKING LOCAL
/*
Chapter 13. LIST Commands

Use the LIST commands to produce a formatted printout of all or selected information contained in a RECON.

LIST Commands In This Chapter

The following is a list of all the LIST commands:
- “LIST.BKOUT”
- “LIST.CAGRP” on page 288
- “LIST.DB” on page 289
- “LIST.DBDS” on page 290
- “LIST.DBDSGRP” on page 292
- “LIST.GSG” on page 293
- “LIST.HISTORY” on page 294
- “LIST.LOG (for a PRILOG Family)” on page 296
- “LIST.LOG (for a Category of Records)” on page 298
- “LIST.RECON” on page 300
- “LIST.SUBSYS” on page 302

LIST.BKOUT

Use a LIST.BKOUT command to list information about the backout record for the selected subsystem or to list all backout records in RECON. For the format of the records listed by this command, see the Appendix B, “Sample Listings from the RECON Data Set,” on page 469.

LIST.BKOUT Syntax

```
LIST.BKOUT ALL | SSID(name) [TIMEFMT(sublist)]
```

Parameters for LIST.BKOUT

ALL | SSID(name)

Mutually exclusive, optional parameters that identify the backout records that are to be displayed.

ALL

Specifies that all the backout records in the RECON are to be displayed.

SSID(name)

Specifies that only one backout record is to be displayed. name is an eight-character alphanumeric string that identifies a valid subsystem ID.

TIMEFMT(sublist)

Optional parameter you use to define the form in which time stamps appear in messages, displays, and listings from DBRC.

Related Reading:
- See “TIMEFMT Parameter” on page 125 for a description of the TIMEFMT parameter sublist format.
LIST.BKOUT

- See “Standard Time Stamp Format” on page 124 for examples of the different output forms.

The LIST commands get the TIMEFMT default from what is specified in the RECON header record.

Example of Running LIST.BKOUT
Here is an example of running the LIST.BKOUT command.

```
/*
//LISTBKOUT   JOB
;
//SYSIN       DD *
   LIST.BKOUT SSID(IMS1)
/*
```

LIST.CAGRP
Use a LIST.CAGRP command to list information in the Copy1 RECON about either a specified CA group or all CA groups. For the format of the records listed by this command, see the Appendix B, “Sample Listings from the RECON Data Set,” on page 469.

LIST.CAGRP Syntax

```
LIST.CAGRP  ALL  |  GRPNAME(name)  |  TIMEFMT(sublist)  |  MEMBERS
```

Parameters for LIST.CAGRP

**ALL | GRPNAME(name)**
Mutually exclusive, optional parameters you use to specify the name of the CA group for which information is to be displayed.

**ALL**
Produces a list of the CA group record and corresponding change accumulation run records for each CA group in RECON.

**GRPNAME(name)**
Produces a list of the CA group record and the change accumulation run records for the group that you request in name.

**MEMBERS**
Optional parameter you use to only list information about the CA group definition. CA execution information is not listed.

**TIMEFMT(sublist)**
Optional parameter you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

Related Reading:
- See “TIMEFMT Parameter” on page 125 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 124 for examples of the different output forms.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.
Example of Specifying the CA Group and CA Records via GRPNAME

In this example, the CA group record and the associated CA records are specified by the GRPNAME parameter.

//LISTCAGP JOB

: //SYSIN DD *
  LISTCAGRP GRPNAME (MYGROUP)
/*

LIST.DB

Use a LIST.DB command to receive a list of databases registered in RECON. You can list one or all database records, with or without their associated DBDS records.

For the format of the records listed by this command, see the Appendix B, “Sample Listings from the RECON Data Set,” on page 469.

The LIST.DB command displays the recoverable or nonrecoverable status of the database.

Related Reading:
- See “HALDB Records” on page 68 for detailed information about how HALDBs are represented in the RECON data set.
- See IMS V9: Administration Guide: Database Manager for an overview of HALDBs and information about how to create them.

LIST.DB Syntax

Parameters for LIST.DB

ALL | TYPEIMS | TYPEFP | TYPHALDB | DBD(dbname)

Mutually exclusive, optional parameters you use to specify which databases in RECON are to be displayed.

ALL
- Specifies that all database records in RECON are to be displayed. For HALDBs, the partition database records are listed under the master record.

TYPEIMS
- Specifies that all database records in RECON that describe a DL/I database are to be displayed.

TYPEFP
- Specifies that all database records in RECON that describe a Fast Path DEDB are to be displayed.

TYPHALDB
- Specifies that all database records that represent HALDBs are to be
displayed, including the HALDB master database records (TYPE=HALDB) along with their associated HALDB partition database records (TYPE=PART).

**DBD(dbname)**
Displays a specific database record or recovery group for a database.

**Restriction for HALDBs:** For HALDBs, specifying the HALDB master name lists the HALDB master record and all of its partition database records. Specifying the partition database name lists only the partition database record.

**COVERED | GSG(gsgname)**
Mutually exclusive, optional parameters you use to qualify the database records in RECON that are to be displayed. Neither COVERED nor GSG(gsgname) can be specified if DBD is specified.

**COVERED**
Specifies that all RSR-covered databases are to be displayed.

**GSG(gsgname)**
Specifies that only databases covered by the specified global service group are to be displayed.

**DBDS**
Optional parameter you use to display those DBDSs or areas in RECON that are associated with the specified database. DBDS information includes recovery-related records (ALLOC, IC, RECOV, REORG). If you do not specify this parameter, no DBDS records or area records are displayed.

**TIMEFMT(sublist)**
Optional parameter you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

**Related Reading:**
- See “TIMEFMT Parameter” on page 125 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 124 for examples of the different output forms.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

**Example of Displaying a Database and Its DBDS Records**
In this example, database HDAMVSAM and its associated DBDS records are displayed.

```
//LISTDB
:
//SYSIN DD *
   LIST.DB DBD(HDAMVSAM) DBDS
/*

**LIST.DBDS**
Use a LIST.DBDS command to display a list of all records in RECON that contain information about a specific DBDS or DEDB area. For the format of the records listed by this command, see the Appendix B, “Sample Listings from the RECON Data Set,” on page 469.
LIST.DBDS Syntax

```
LIST.DBDS DBD(name) | GROUP(name) | DDN(name) | AREA(name) | TIMEFMT(sublist)
```

Parameters for LIST.DBDS

**DBD(name) | GROUP(name)**

Mutually exclusive, required parameters you use to identify the DBDS or DEDB area being listed.

**DBD**

Specifies the database name of the DBDS or DEDB area being displayed.

**Restrictions for HALDBs:** For HALDBs, you can specify either a HALDB master database name or a HALDB partition name when you do not specify the DDN parameter. When you specify the DDN parameter, you can only specify a HALDB partition name.

**GROUP**

Specifies that all DBDSs or DEDB areas of the named DBDS group are to be displayed. If GROUP is specified, the LIST.DBDS command is executed for each member of the identified group.

**DDN(name) | AREA(name)**

Mutually exclusive, optional parameters you use to identify the DBDS or DEDB area to be displayed. You specify one of these parameters only when you specify the DBD parameter.

**DDN(name)**

Specifies the name of the DBDS to display.

**Restrictions for HALDBs:** For HALDBs, you must specify a HALDB partition name (not a HALDB master name) with the DBD parameter in order to use the DDN parameter. The DDN parameter value is the HALDB partition DDN. The LIST.DBDS command is performed for the identified DBDS of the partition. The LIST.DBDS command fails if DDN does not identify a DBDS in the partition.

**AREA(name)**

Specifies the name of the DEDB area to display.

If neither DDN nor AREA is specified, the LIST.DBDS command is executed for each DBDS or DEDB area of the specified database.

**Restrictions for HALDBs:** If you specify a HALDB master name, the LIST.DBDS command is performed for each DBDS for each HALDB partition in the HALDB master. If you specify a HALDB partition name, the LIST.DBDS command is performed for each DBDS of the identified partition.

**TIMEFMT(sublist)**

Optional parameter you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

**Related Reading:**

- See “TIMEFMT Parameter” on page 125 for a description of the TIMEFMT parameter sublist format.
LIST.DBDS

- See “Standard Time Stamp Format” on page 124 for examples of the different output forms.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

Example of Displaying AREA Parameters
In this example, information about the DEDB area specified in the DBD and AREA parameters is displayed.

```
//LISTDBDS JOB :  
//SYSIN DD *  
  LISTDBS DBD(FPEDBD02) AREA(AREA01) /*
```

LIST.DBDSGRP

Use a LIST.DBDSGRP command to display a list of any of the following:
- All three kinds of data group records (DB groups, DBDS groups, and recovery groups) in RECON
- The members of a single data group record
- All data group records containing a specified member or members

Related Commands:
- “INIT.DBDSGRP” on page 271
- “CHANGE.DBDSGRP” on page 157

LIST.DBDSGRP Syntax

```
LIST.DBDSGRP [GRPNAME(name)] ALL(ALL | (dbname, areaname, ddname)) TIMEFMT(sublist)
```

Parameters for LIST.DBDSGRP

**GRPNAME(name)** | **ALL**
--- | ---
Mutually exclusive, optional parameters you use to specify the groups to be listed.

**ALL**
- Produces a list of all the DB, DBDS, and recovery groups identified in RECON. All is the default.

**GRPNAME**
- Produces a list of the members of the specified group. The named group must exist in RECON.

**ALL(dbname, areaname, ddname)**
- Produces a limited list of the DB, DBDS, and recovery groups identified in the RECON. Use the parameters (dbname, areaname, ddname) to limit the number of records listed. A group is listed only if it contains one or more of the databases, DBDSs, or areas specified.
Restriction for HALDBs: For the LIST.DBDSGRP command, dbname can specify either the name of a HALDB partition or a HALDB master database.

TIMEFMT(sublist)
Optional parameter you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

Related Reading:
• See “TIMEFMT Parameter” on page 125 for a description of the TIMEFMT parameter sublist format.
• See “Standard Time Stamp Format” on page 124 for examples of the different output forms.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

Example of Displaying the Members of a DBDS Group
In this example, the members of a specified DBDS group are displayed.

//LISTDBGP  JOB
:
//SYSIN DD *
   LIST.DBDSGRP GRPNAME(DBDSG1)
/*

LIST.GSG
Use a LIST.GSG command to receive a list of the global service group records in RECON. This command fails if RSRFEAT=NO is specified in the IMSCTRL macro. For the format of the records listed by this command, see the Listings from the RECON Data Set,” on page 469.

LIST.GSG Syntax

Parameters for LIST.GSG

ALL | GSGNAME(name)
Mutually exclusive, optional parameters you use to specify which GSG records in RECON are to be displayed.

ALL
Specifies that all GSG records in RECON are to be displayed.

GSGNAME(name)
Displays a specific GSG record.

TIMEFMT(sublist)
Optional parameter you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

Related Reading:
• See “TIMEFMT Parameter” on page 125 for a description of the TIMEFMT parameter sublist format.
LIST.GSG

- See "Standard Time Stamp Format" on page 124 for examples of the different output forms.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

Example of Listing a Global Service Group
In this example a single GSG record is displayed.

//LISTGSG JOB
::
//SYSIN DD *
LIST.GSG GSGNAME(GSG1)
/*

LIST.HISTORY

Use the LIST.HISTORY command to produce a history-of-activity listing for DBDSs or DEDB areas. The output of the LIST.HISTORY command consists of:

- The database record listing.
- The DBDS record listing. (For a DEDB area, the area authorization and area recovery records are combined to form a DBDS record listing.)
- The following records (if present), listed in time sequence order under each DBDS or area:
  - ALLOC records.
  - IMAGE records.
  - CA execution records, showing the purge time only for the current DBDS or area.
  - RECOV records.
  - REORG records.
- The PRILOG records associated with all ALLOC records listed.
- A time line summary that interrelates all the events represented by the records listed above.

You can use the FROMTIME and/orTOTIME parameters to define a time range that excludes these records:

- ALLOC records for USIDs that are not active within the range. If any ALLOC record is active within the time range, all ALLOCS for the same USID are listed.
- IMAGE records with RUN times (or for CICs, an effective purge time) outside the range.
- CA execution records with STOP and PURGE times outside the range.
- RECOV records with RUN and RECOV TO times outside the range.
- REORG records with RUN times outside the range.

See "Sample Listing of LIST.HISTORY Output" on page 470 for an illustration of sample output produced by the LIST.HISTORY command.

LIST.HISTORY Syntax
Parameters for LIST.HISTORY

**DBD**(name) | **GROUP**(name)

Mutually exclusive, required parameters you use to identify the DBDS or DEDB area to list.

**DBD**

Specifies the database name of the DBDS or DEDB area to be listed.

**Restrictions for HALDBs:** For HALDBs, you can specify either a HALDB master database name or a HALDB partition name when you do not specify the DDN parameter. When you specify the DDN parameter, you can only specify a HALDB partition name.

**GROUP**

Specifies that all the DBDSs or DEDB areas of a DBDS group or CA group are to be listed. If GROUP is specified, the **LIST.HISTORY** command is executed for each member of the identified group.

**DDN**(name) | **AREA**(name)

Mutually exclusive, optional parameters you use to identify the DBDS or DEDB area to be listed. You specify one of these parameters only when you specify the DBD parameter.

**DDN**

Specifies the name of the DBDS to list.

**Restrictions for HALDBs:** For HALDBs, you must specify a partition database name with the DBD parameter in order to use the DDN parameter. The DDN parameter value is the partition DDN. The **LIST.HISTORY** command is performed for the identified DBDS of the partition. The **LIST.HISTORY** command fails if DDN does not identify a DBDS in the partition.

**AREA**

Specifies the name of the DEDB area to list.

If neither DDN nor AREA is specified, the **LIST.HISTORY** command is executed for each DBDS or DEDB area of the specified database.

**Restrictions for HALDBs:** If you specify a HALDB master name, the **LIST.HISTORY** command is performed for each DBDS for each partition in the HALDB master. If you specify a HALDB partition name, the **LIST.HISTORY** command is performed for each DBDS of the identified partition.

**FROMTIME**(time_stamp)

Optional parameter you use to specify the time stamps of the DBDS or DEDB area records that are to be listed in time sequence order. The time stamp must be in standard form (see "Standard Time Stamp Format" on page 124). Those records that are not listed in time-sequence order are listed regardless of whether FROMTIME or TOTIME are specified. FROMTIME specifies the time stamp of the oldest record to be listed. If you specify only FROMTIME, all subsequent, pertinent records in RECON are listed.

You can combine the FROMTIME and TOTIME parameters in order to specify a range of records to display.
LIST.HISTORY

If you specify neither FROMTIME nor TOTIME, all the records that exist in RECON for the specified DBDSs or DEDB areas are listed.

TOTIME\((time\_stamp)\)
Optional parameter you use to specify the time stamps of the DBDS or DEDB area records to be listed in time-sequence order. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124). Those records not listed in time-sequence order are listed regardless of whether FROMTIME or TOTIME are specified. TOTIME specifies the time stamp of the last record to be listed. If you specify only TOTIME, that record plus all prior, pertinent records in RECON are listed.

You can combine the FROMTIME and TOTIME parameters in order to specify a range of records to display.

If you specify neither FROMTIME nor TOTIME, all the records that exist in RECON for the specified DBDSs or DEDB areas are listed.

TIMEFMT\((sublist)\)
Optional parameter you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

Related Reading:
- See “TIMEFMT Parameter” on page 125 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 124 for examples of the different output forms.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

Timeline
Optional parameter you use to specify that only the time line summary should be listed in the output.

Example of Displaying a DBDSs Activity History
In this example, the activity history of a specified DBDS is displayed.

```
//LISTHIST JOB
:
//SYSIN DD *
   LIST.HISTORY DBD(DB1) DDN(NAME1)
/*
```

LIST.LOG (for a PRILOG Family)
Use this command to display the PRILOG record and any of the following records having the specified start time.
- LOGALL
- SECLOG
- PRISLD
- SECSLD

LIST.LOG (for a PRILOG Family) Syntax

```
LIST.LOG STARTIME(time_stamp) SSID(name) GSG(gsname) TRACKING
```
Parameters for LIST.LOG (for a PRILOG Family)

STARTIME(time_stamp)
   Required parameter you use to specify the start time of the records that you want displayed.

SSID(name)
   Optional parameter that limits the display of log records or OLDS entries to those associated with the specified subsystem.

GSG(gsgname) | TRACKING
   Mutually exclusive, optional parameters that you use to specify that log data sets for an RSR tracking subsystem are to be displayed. LIST.LOG ALL does not show RSR tracking subsystem information.

   GSG(gsgname)
   Optional parameter that limits the display of log records to those associated with the specified GSG.

   If the name in any of the records listed does not match the specified name, message DSP0144 is issued and processing continues.

TRACKING
   Indicates that only records of log data sets created by tracking subsystems in the global service group are to be displayed.

TIMEFMT(sublist)
   Optional parameter you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

   Related Reading:
   • See “TIMEFMT Parameter” on page 125 for a description of the TIMEFMT parameter sublist format.
   • See “Standard Time Stamp Format” on page 124 for examples of the different output forms.

   The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

Attention: If the ERROR, OPEN, or UNARCH parameters are coded, message DSP0141I is issued and the command fails.

If the FROMTIME, TOTIME, INTERIM, or TRACKING parameters are coded, STARTIME is ignored and the command defaults to ALL processing as described under the next section.

Example of Listing a PRILOG Family of Records

In the example, all PRILOG, SECLOG, PRISLD, SECSDL, and LOGALL records that have the specified start time are listed.

//LISTLOG   JOB
  :
//SYSIN DD *
LIST.LOG STARTIME('97.023 12:12:12.1 PST')
/*
LIST.LOG (for a category of records)

LIST.LOG (for a Category of Records)
Each command form is followed by a list of the records it displays.

LIST.LOG (for a Category of Records) Syntax

- PRILOG
- LOGALL
- SECLOG
- PRISLD
- SECSLD
- PRIOLD
- SECOLD

- IPRI
- ISEC
- IPRISL
- ISECSL
- IPRIOL
- ISECOL

- PRITSLDS
- SECTSLDS

- IPRITSLD
- ISECTSLD

- PRIOLD
- SECOLD

- IPRIOL
LIST.LOG (for a category of records)

- ISECOL

\[ \text{LIST.LOG} \text{ OLDS(ddname)} \text{ SSID(name)} \]

See "Optional Filters"

- PRIOLD

- SECOLD

LIST.LOG OLDS displays only the data set entries with matching DD names and subsystem names. If SSID is omitted, processing is the same as for ALLOLDS.

Optional Filters: The LIST.LOG command can be further qualified with one or more of the following optional parameters. For example, combining SSID and OPEN limits the display to logs and OLDS entries that belong to a specified subsystem and to those that are not closed.

The optional parameters for LIST.LOG ALL, ALLOLDS, and OLDS are:

\[ \text{FROMTIME(time stamp)} \text{ TOTIME(time stamp)} \text{ SSID(name)} \text{ GSG(gsgname)} \]

Parameters for LIST.LOG (for a Category of Records)

FROMTIME(time stamp)

Optional parameter that limits the display to log records or OLDS entries starting at, or after, this time. The time stamp must be in standard form (see "Standard Time Stamp Format" on page 124).

TOTIME(time stamp)

Optional parameter that limits the display to log records or OLDS entries starting at, or before, this time. The time stamp must be in standard form (see "Standard Time Stamp Format" on page 124).

SSID(name)

Optional parameter that limits the display to log records or OLDS entries associated with the specified subsystem.

GSG(gsgname)

Optional parameter that limits the display of log records to those associated with the specified GSG.

OPEN

Optional parameter that limits the display to log records or OLDS entries that are not closed.

ERROR

Optional parameter that limits the display to log records having one or more data set entries marked in error, and to OLDS entries marked in error.

UNARCH

Optional parameter that limits the display of OLDS entries to those that are not archived.

Attention: Specifying UNARCH without the ERROR or OPEN parameters causes ALL to be processed like ALLOLDS; that is, no log records are listed, only unarchived OLDS entries are listed.
LIST.LOG (for a category of records)

The following parameter can be used on all forms of the command.

```
TIMEFMT(sublist)
```

**TIMEFMT(sublist)**
Optional parameter you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

**Related Reading:**
- See "TIMEFMT Parameter" on page 125 for a description of the TIMEFMT parameter sublist format.
- See "Standard Time Stamp Format" on page 124 for examples of the different output forms.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

**Attention:** If the ERROR, OPEN, or UNARCH parameters are coded, message DSP0141I is issued and the command fails.

If the FROMTIME,TOTIME,INTERIM,or TRACKING parameters are coded, STARTIME is ignored and the command defaults to ALL processing as described under the next section.

**Example of Displaying RECON Records Specified by STARTIME**
In this example, the RLDSs, SLDS, and corresponding LOGALL record with the time stamp specified in the STARTIME parameter are to be displayed.

```
//LISTRCON JOB
:
//SYSIN DD *
   LIST.LOG STARTIME(840311313130)
/*
```

**Example of Displaying a Subsystem’s OLDS Records**
In this example, the OLDS records from subsystem IMSA are to be displayed.

```
//LISTRCON JOB
:
//SYSIN DD *
   LIST.LOG ALLOLDS SSSID(IMSA)
/*
```

**LIST.RECON**
Use the LIST.RECON command to obtain a display of the RECON’s current status and a formatted display of all records it contains.

Use the STATUS keyword to display RECON status only.

RECON status information includes the following items:
- The contents of the Time Zone Label Table
- The TIMEZIN and TIMEFMT settings
- The minimum IMS release level to coexist with this RECON
• The status of each of the three RECONs as follows:

<table>
<thead>
<tr>
<th>RECON status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY1</td>
<td>PRIMARY ACTIVE RECON</td>
</tr>
<tr>
<td>COPY2</td>
<td>SECONDARY ACTIVE RECON</td>
</tr>
<tr>
<td>SPARE</td>
<td>AVAILABLE RECON</td>
</tr>
<tr>
<td>UNAVAILABLE</td>
<td>UNAVAILABLE RECON</td>
</tr>
<tr>
<td>DISCARDED</td>
<td>UNUSABLE RECON</td>
</tr>
</tbody>
</table>

The RECON is unavailable when the resource is allocated in another system.

**LIST.RECON Syntax**

```
LIST.RECON [STATUS] [TIMEFMT(sublist)]
```

**Parameters for LIST.RECON**

**STATUS**
Optional parameter you use to request the RECON header record information and the status of all RECONS. If you specify this parameter the listing of the remainder of the records is suppressed.

**TIMEFMT(sublist)**
Optional parameter you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

**Related Reading:**

- See “TIMEFMT Parameter” on page 125 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 124 for examples of the different output forms.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

**Examples of Displaying the RECONs**

Here are some examples of using the LIST.RECON command.

**Example of Displaying the RECONs:** In this example, the status and contents of RECON are displayed.

```
//LISTRCON JOB
:
//SYSIN DD *
LIST.RECON
/*
```

For a sample of RECON, see the Appendix B, “Sample Listings from the RECON Data Set,” on page 469.

**Example of Displaying RECON Header and Status Information:** In this example, only RECON header status information is displayed.
LIST.RECON

//LISTRCON JOB
::
//SYSIN DD *
   LIST.RECON STATUS
/*

Only the first segment of output shown in the Appendix B, “Sample Listings from the RECON Data Set,” on page 469 is produced in this case.

Related Reading: See Chapter 4, “Initializing and Maintaining the RECON Data Set,” on page 55 for an explanation of the possible RECON states.

LIST.SUBSYS

Use a LIST.SUBSYS command to receive a formatted list of the subsystems registered in RECON. For the format of the records listed by this command, see the Appendix B, “Sample Listings from the RECON Data Set,” on page 469.

LIST.SUBSYS Syntax

```
LIST.SUBSYS ONLINE
   SSID(name)
     ALL
     BATCH
   TIMEFMT(sublist)
     TRACKING
```

Parameters for LIST.SUBSYS

**SSID(name)** | **ALL** | **ONLINE** | **BATCH**
---|---|---|---
Mutually exclusive, optional parameters you use to specify which subsystem information is to be displayed.

**SSID(name)**
Specifies the name of the subsystem for which information is to be displayed.

**ALL**
Specifies that all subsystem information, both batch and online, is to be displayed.

**ONLINE**
Specifies that all online subsystem information is to be displayed.

**BATCH**
Specifies that all batch subsystem information is to be displayed.

**TIMEFMT(sublist)**
Optional parameter you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

Related Reading:
- See “TIMEFMT Parameter” on page 125 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 124 for examples of the different output forms.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.
TRACKING

Specifies that all RSR tracking subsystem information is to be displayed.

Example of Displaying All Online Subsystem Records

In this example, all online subsystem records are displayed.

```
//LISTSS JOB

//SYSIN DD *
  LIST.SUBSYS
/*
```
Chapter 14. NOTIFY Commands

Use the NOTIFY commands to add information to a RECON that is normally written there automatically.

NOTIFY Commands In This Chapter

The following is a list of all the NOTIFY commands:

- “NOTIFYALLOC”
- “NOTIFY.BKOUT” on page 307
- “NOTIFY.CA” on page 308
- “NOTIFY.IC” on page 310
- “NOTIFY.PRILOG (for OLDS)” on page 313
- “NOTIFY.PRILOG (for RLDS)” on page 317
- “NOTIFY.PRILOG (for SLDS and TSLDS)” on page 322
- “NOTIFY.RECOV” on page 326
- “NOTIFY.REORG” on page 328
- “NOTIFY.SECLOG (for OLDS)” on page 331
- “NOTIFY.SECLOG (for RLDS)” on page 334
- “NOTIFY.SECLOG (for SLDS and TSLDS)” on page 338
- “NOTIFY.SUBSYS” on page 341
- “NOTIFY.UIC” on page 342

NOTIFYALLOC

Use a NOTIFYALLOC command to add information to RECON about either a specific database allocation or a specific database deallocation of a DBDS or DEDB area. This addition of information is required only when RECON was not updated during a run of IMS that resulted in an allocation of the DBDS or DEDB area for updates. You should not need to use this command under normal operating conditions.

The NOTIFYALLOC command fails if the DBDS or DEDB area is nonrecoverable, user-recoverable, or in use.

HALDB Restriction: This command is not allowed for ILDS or Index DBDSs of HALDB partitions.

NOTIFYALLOC Syntax

```
NOTIFYALLOC ALLTIME(time_stamp) DBD(name) DDN(name) AREA(name)
```

```
DEALTIME(time_stamp)
STARTIME(time_stamp)
DSSN(value)
USID(value)
```

Parameters for NOTIFYALLOC

ALLTIME(time_stamp)

Required parameter you use to specify the time stamp of the allocation of the
NOTIFY.ALLOC

database that contains the DBDS or DEDB area that is specified in this command. The time stamp must be in standard form (see "Standard Time Stamp Format" on page 124).

When used with the STARTIME parameter, ALLTIME causes a new allocation record to be written in RECON. When used with a DEALTIME parameter, it identifies the allocation record in RECON for which a deallocation time is being added.

DBD(name)
Required parameter you use to specify the database name of the DBDS or DEDB area for which you are adding allocation information to RECON.

Restriction for HALDBs: When using the NOTIFY.ALLOC command for HALDBs, name must specify the name of a HALDB partition.

DDN(name) | AREA(name)
Required parameter you use to specify the data set ddname of the DBDS or DEDB area for which you are adding allocation information to RECON.

DEALTIME(time_stamp) | STARTIME(time_stamp)
Mutually exclusive, required parameters. The time stamp must be in standard form (see "Standard Time Stamp Format" on page 124).

DEALTIME
Specifies the time stamp of the deallocation of the database for the specified DBDS or DEDB area. This addition to RECON is required only if the database is allocated for updates and explicitly deallocated before the end of an IMS run.

STARTIME
Specifies the starting time stamp of the log data set that was active at the time of the allocation specified in the ALLTIME parameter.

DSSN(value)
Optional parameter you use to specify which data set sequence number is placed in the allocation record to be created. If you do not specify the DSSN parameter, the data set sequence number for the new allocation record is 0, indicating no data sharing. If you are using data sharing, you must specify the appropriate DSSN. You use this parameter for log-merge processing.

USID(value)
Optional parameter you use to specify the update set identifier of the database or area when the update occurred.

USID is required if the database or area is assigned to a GSG. If the database or area is not assigned to a GSG, USID cannot be specified.

Example of Adding Allocation Information to RECON
In this example, information about an allocation of a specified DBDS is to be added to RECON. The ALLTIME parameter specifies the time stamp of the allocation of the DBDS or DEDB area; the STARTIME parameter specifies the time stamp of the start of the log data set that was active at the time of the allocation.

//NOTYALLOC JOB

::

//SYSIN DD * NOTIFY.ALLOC DBD(DB1) DDN(DD1) -
STARTIME(820670201010) -
ALLTIME(820670308200)
/*
NOTIFY.BKOUT

Use the NOTIFY.BKOUT command to create a backout record for a specified subsystem and to add a single unit of recovery (UOR) entry to the record that is created. Additional UOR entries can be added to the backout record by using the CHANGE.BKOUT command.

NOTIFY.BKOUT Syntax

```
NOTIFY.BKOUT SSID(name) UOR(uor) UORTIME(time_stamp) PSB(name)
```

Parameters for NOTIFY.BKOUT

SSID(name)

Required parameter you use to specify the subsystem for which the backout record is to be created. The name is an eight-character, alphanumeric string that represents any valid subsystem name.

UOR(uor)

Required parameter you use in conjunction with the UORTIME parameter to identify a unit of recovery in the backout record. The recovery token (uor) is a 16-byte field that describes a specific UOR that is to be included with the backout record. uor must be 32 hexadecimal digits expressed as a character string; for example, UOR(E2E8E2F3404040400000000600000003).

The recovery token is intended to be a unique identifier, but it can be duplicated across restarts. When you include UORTIME, you eliminate the problem of possible duplication.

UORTIME(time_stamp)

Required parameter you use to specify the time of the UOR to be added to the backout record. The value is the time stamp of the beginning of the UOR (found in the X’5607’ log record). The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

PSB(name)

Required parameter you use to identify the PSB associated with the UOR.

DBD(name)

Optional parameter you use to identify up to eight databases having changes associated with the unit of recovery that require backout.

Restriction for HALDBs: When using the NOTIFY.BKOUT command for HALDBs, name must specify the name of a HALDB partition.

BKO(name)

Optional parameter you use to identify up to eight databases having changes associated with the unit of recovery that have already been backed out.

Use BKO to identify databases that have already been backed out from this UOR-UORTIME combination. You can specify either the BKO parameter, the DBD parameter, or both. A database name can appear in either BKO or DBD, but not both. (A database cannot both be backed out and require a backout at the same time.)
Example of Adding a Backout Record to RECON
In this example, a backout record for subsystem SYS3 is added to the RECON.

```
//NFYBKOUT JOB
:
//SYSIN DD *
   NOTIFY.BKOUT SSID(SYS3)
   UOR(E2E82F34040400000000600000003)
   UORTIME(930931345027) PSB(APPL34)
   DBD(DATA1,DATA2,DATA3C)
   BKO(DATA4,DATA5,DATA3A)
/*
```

NOTIFY.CA
Use a NOTIFY.CA command to add information about a run of the Database Change Accumulation utility for a specified CA group.

NOTIFY.CA Syntax

```
NOTIFY.CA—CADSN(name)—GRPNAME(name)—RUNTIME(time_stamp)—STOPTIME(time_stamp)

VOLLIST(volser)
FILESEQ(value)

PURGLIST(time_stamp, NO, COMPLETE, INCOMPLETE)
UNIT(unittype)

COMP
SUBSET
```

Parameters for NOTIFY.CA

**CADSN(name)**
Required parameter you use to specify the data set name of the change accumulation data set in the identified record. If the CA group is defined as reusable, the data set name must be unique. DBRC does not check for duplicate data set names.

**GRPNAME(name)**
Required parameter you use to specify the name of the CA group for which information is to be added.

**RUNTIME(time_stamp)**
Required parameter you use to identify the specific change accumulation run record to be added. The time stamp represents the time at which the Database Change Accumulation utility was run, and it must be in standard form (see “Standard Time Stamp Format” on page 124).

**STOPTIME(time_stamp)**
Required parameter you use to specify the time stamp of the change
accumulation run record for which information is to be added. The time stamp is the stop time of the last log volume that was processed by the specified run of the Change Accumulation utility, and it must be in standard form (see “Standard Time Stamp Format” on page 124).

**VOLLIST(volser)**

Required parameter you use to specify the volume serial numbers of the change accumulation data set in the specified change accumulation run record. You can specify a maximum of 255 volume serial numbers for volser. Each volume serial number can be a maximum of six alphanumeric characters.

**FILESEQ(1 | value)**

Optional parameter you use to specify the file sequence number of the identified change accumulation data set.

value must be a decimal number from 1 to 9999.

**PURGLIST(time_stamp,YES | NO,COMPLETE | INCOMPLETE)**

Optional parameter you use to specify the purge time, which is the point in time in the input log records where change accumulation started and to specify whether the logs form a complete subset.

The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124). If you do not specify a time stamp, the time is set to 0.

If you are using the accumulated changes as input to recovery, you must choose a purge time that satisfies the DBRC input requirements for recovery. Recovery first chooses an image copy and then uses a change accumulation whose purge time for that DBDS matches the run time of the image copy.

**YES | NO**

Mutually exclusive subparameters you use to specify whether any changes for the corresponding DBDS have been accumulated.

**YES**

Specifies that some changes have been accumulated for the corresponding DBDS.

**NO**

Specifies that no changes have been accumulated for the corresponding DBDS.

**COMPLETE | INCOMPLETE**

Mutually exclusive subparameters you use to specify whether the logs form a complete subset. To determine whether a log subset is complete, use the LIST.CAGRP command. See “LIST.CAGRP” on page 288 for more information.

**COMPLETE**

Specifies that the logs form a complete subset. When you specify COMPLETE, the time stamp of the STOPTIME parameter is the stop time of the last log input to the Change Accumulation utility.

**INCOMPLETE**

Specifies that the logs form an incomplete subset. When you specify INCOMPLETE, the time stamp of the STOPTIME parameter is the start time of the earliest unselected (open) log volume. This volume should be the first one that is selected at a later run.

If you specify the PURGLIST parameter, the order of the time stamp and the change indicator in the purge list corresponds to the order of the DBDS names specified in the GRPMEM parameter of the INIT.CAGRP command. For
example, the third purge time and change indicator is the purge time for the
third DBDS that is specified in the GRPMEM parameter of the INIT.CAGRP
command.

If you specify fewer subparameters with the PURGLIST parameter than you
specified with the GRPMEM parameter of the INIT.CAGRP command, DBRC
uses the defaults of NO and COMPLETE for each DBDS that you omit.
Similarly, if you do not specify the PURGLIST parameter, DBRC uses the
defaults of NO and COMPLETE for each DBDS specified with the GRPMEM
parameter of the INIT.CAGRP command. To use a default of NO for certain
DBDSs, use commas to indicate which DBDSs are subject to the default.

UNIT(3400 | unittype)
Optional parameter you use to specify the unit type of the volumes on which
the change accumulation data set resides. unittype can be up to eight
alphanumeric characters.

COMP | SUBSET
Mutually exclusive, optional parameters you use to indicate that the change
accumulation record’s stop time is a log volume start time.

COMP
Indicates that when the CA was created, a complete set of logs was
processed and that the CA’s stop time is the stop time of the last log
volume processed.

SUBSET
Indicates that when the CA was created, a subset of logs was processed
and the CA’s stop time is the start time of the first unprocessed log
volume. Specifying INCOMPLETE in the PURGLIST parameter does not
automatically cause SUBSET to be set.

You do not need to use this parameter under normal conditions. Checking is
not done to verify that the use of this parameter is consistent with the value of
the CA stop time. This parameter value is used by the GENJCL.CA and
GENJCL.RECOV processes. Incorrect use of this parameter can result in invalid
generated JCL.

Example of Adding CADSN Information to RECON
In this example, information about a change accumulation data set is to be added
to RECON.

```
//NFYCA   JOB

; SYSIN DD *
//SYSIN   DD *
NOTIFY.CA   GRPNAME(CAGRP2) -
STOPTIME(8402402020) -
RUNTIME(840250305029) -
VOLLIST(VOL005) -
PURGLIST((840240302005,YES),,(840250420256,))
/*
```

NOTIFY.IC
Use a NOTIFY.IC command to add information about an image copy.

HALDB Restrictions:
• The NOTIFY.IC command is not allowed for ILDS or Index DBDSs of HALDB
  partitions
The NOTIFY.IC command will fail if it is attempting to record an image copy of a HALDB that has an online reorganization in progress. This is because an image copy of a HALDB cannot be taken while the HALDB is being reorganized online and the reorganization has not completed (either it is in progress or was terminated).

**NOTIFY.IC Syntax**

```
NOTIFY.IC DBD(name) DDN(name) AREA(name) ICDSN(name) RUNTIME(time_stamp)
```

**Parameters for NOTIFY.IC**

**DBD(name)**
Required parameter you use to specify the database name of the DBDS or area for which an image copy run record is to be added.

**Restriction for HALDBs:** When using the NOTIFY.IC command for HALDBs, `name` must specify the name of a HALDB partition.

**DDN(name) | AREA(name)**
Mutually exclusive, required parameters you use to specify the data set ddbname of the DBDS (use DDN) or DEDB area (use AREA) for which an image copy run record is to be added.

**ICDSN(name)**
Required parameter you use to specify the data set name of the image copy data set that contains the image copy whose run record is being added. `name` can be a maximum of 44 characters.

**RUNTIME(time Stamp)**
Required parameter you use to specify the time the image copy utility was run. The `time stamp` must be in standard form (see “Standard Time Stamp Format” on page 124).

**FILESEQ(value)**
Optional parameter you use to specify the file sequence number of the identified image copy data set. `value` must be a decimal number from 1 to 9999.
FILESEQ2(value)
Optional parameter you use to specify the file sequence number of the identified duplicate image copy data set. value can be a decimal number from 1 to 9999.

You can specify this parameter only if you specify the VOLLIST2 parameter. If the VOLLIST2 parameter is specified, then FILESEQ2(1) is the default for this parameter.

ICDSN2(name)
Optional parameter you use to specify the data set name of the duplicate image copy data set that is to contain the image copy whose run record is being added. name can be a maximum of 44 characters.

BATCH | ONLINE | CIC | SMSCIC | SMSNOIC
Mutually exclusive, optional parameters you use to specify the type of image copy that the data set contains.

Restriction: CIC, SMSCIC, and ONLINE cannot be used for user-recoverable databases.

BATCH
Indicates that the Database Image Copy (DFSUDMP0) utility was used to create the image copy while the database was unavailable for update processing (the CIC parameter was not specified). BATCH can also be specified to record the output of the HISAM Reorganization Unload utility image copy.

ONLINE
Specifies that the image copy data set was obtained by executing the Online Database Image Copy utility. You must use the STOPTIME parameter when you specify ONLINE.

CIC
Indicates that a concurrent image copy was taken. A concurrent image copy is a “fuzzy” copy, so the data set uses logs in order to complete the image. STOPTIME must be used if CIC is specified. CIC cannot be used to copy VSAM KSDS databases.

SMSCIC
Indicates that the Database Image Copy 2 was used to create the image copy while the database was available for update processing (‘S’ was specified on the utility control statement). The image copy is in DFSMS dump format. The image copy is a “fuzzy” copy so logs must be applied to recover the data set to a usable state. The STOPTIME parameter must be specified when you specify SMSCIC.

SMSNOIC
Indicates that the Database Image Copy 2 utility was used to create the image copy while the database was unavailable for update processing (‘X’ was specified on the utility control statement). The image copy is in DFSMS dump format.

RECDCT(value)
Optional parameter you use to specify the count of the records in the image copy data set. value must be a decimal number from 1 to 2147483647.

STOPTIME(time_stamp)
Optional parameter you use to specify the stop time of the online or
concurrent image copy. You must specify this parameter when online, CIC, or SMS/CIC is specified. The time stamp must be in standard form (see "Standard Time Stamp Format" on page 124).

UNIT(3400 | unittype)
Optional parameter you use to specify the unit type of the image copy data set. The unit type can be up to eight alphanumeric characters.

UNIT2(3400 | unittype)
Optional parameter you use to specify the unit type of the duplicate image copy data set. The unit type can be up to eight alphanumeric characters long.

VOLLIST(volser)
Optional parameter you use to specify (in the batch or online image copy record) the volume serial numbers of the volumes on which the specified image copy data set resides. This parameter is required when either ONLINE or BATCH parameters are specified.

VOLLIST2(volser)
Optional parameter you use to specify the volume serial numbers (in the image copy record) of the volumes on which the specified duplicate image copy data set resides. This parameter is required if ICDSN2 is specified, and when either online or batch parameters are specified.

USID(value)
Optional parameter you use to specify the update set identifier of the database or area when the reorganization occurred.

USID is required if the database or area is assigned to a GSG. If the database or area is not assigned to a GSG, USID is optional.

Example of Notifying DBRC of Concurrent Image Copy Completion
In this example, DBRC is notified of the successful completion of a concurrent image copy for the area specified. RUNTIME refers to the time the image copy started. STOPTIME refers to the time the image copy ended.

```
//NFYIC JOB

//SYSIN DD *
NOTIFY.IC DBD(DBD001) AREA(AREA1)
RUNTIME(B5200002020)
STOPTIME(B5200004040)
ICDSN(IC0005) CIC
/*
```

NOTIFY.PRILOG (for OLDS)
Use a NOTIFY.PRILOG command to add information about a primary OLDS to RECON and to manually create interim PRILOG in RECON. You would do this in a case where the log processing exit routines of the IMS system failed to do so.

NOTIFY.PRILOG (for OLDS) Syntax

```
NOTIFY.PRILOG—OLDS(ddname) DSN(name) STARTIME(time_stamp) |
| RUNTIME(time_stamp) |
| FIRSTREC(number) | GSG(gsname) | INTERIM | LASTREC(number) |
```
**NOTIFY.PRILOG (for OLDS)**

Parameters for NOTIFY.PRILOG (for OLDS)

**OLDS(ddname)**
Required parameter you use to specify that a record is to be created in RECON for an OLDS. If you do not specify OLDS, the default is RLDS (see "NOTIFY.PRILOG (for RLDS)" on page 317). *ddname* is the DD statement that the IMS online control region used when it used the OLDS.

**DSN(name) | RUNTIME(time_stamp)**
Mutually exclusive, required parameters.

**DSN**
Specifies the data set name of the primary OLDS for which a log record is being created in RECON.

**RUNTIME**
Specifies the time stamp of a close operation for the specified primary OLDS. The time stamp must be in standard form (see "Standard Time Stamp Format" on page 124).

These two parameters are used in conjunction with the STARTIME, FIRSTREC, and NXTOLDS parameters to identify what type of primary OLDS entry is to be added to the RECON. [Table 14](#) indicates which parameter combinations are required for each type of primary OLDS entry.

**Definitions:**

**OLDS Open**
Starting to use an OLDS and inserting a starting time.

**OLDS Switch**
Stopping the use of the current OLDS (inserting a stop time) and starting to use the next OLDS (inserting a starting time into the new OLDS).

**OLDS Close**
Stopping the use of an OLDS (inserting a stop time)

<table>
<thead>
<tr>
<th>Type of Log Entry</th>
<th>Required Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLDS Open</td>
<td>STARTIME, DSN, FIRSTREC</td>
</tr>
<tr>
<td>OLDS Switch</td>
<td>STARTIME, DSN, FIRSTREC, NXTOLDS</td>
</tr>
<tr>
<td>OLDS Close</td>
<td>LASTREC, STARTIME, RUNTIME</td>
</tr>
</tbody>
</table>

For each primary OLDS, you must issue a separate NOTIFY.PRILOG command for open, switch, and close.

**STARTIME(time_stamp)**
Required parameter you use to specify the starting time of a primary OLDS. The time stamp must be in standard form, see "Standard Time Stamp Format" on page 124.

See [Table 14](#) for a description of the use of this parameter with other parameters in the NOTIFY.PRILOG command.
FIRSTREC\textit{(number)}

Optional parameter you use to specify the log record sequence number of the first log record of the OLDS. For the first OLDS of the PRILOG, it corresponds to the first log record that was written during initialization of the IMS subsystem.

FIRSTREC is required for OLDS\texttt{OPEN} and \texttt{SWITCH} commands. It specifies the first log record sequence number on the OLDS that is being opened. It is invalid for a \texttt{CLOSE} command.

The log record sequence number can be one of the following:

- A hexadecimal number
  
  This number is 1 to 16 characters, enclosed in single quotes and preceded by the letter, X. For example: \texttt{FIRSTREC(X'10B9C')}.

- A decimal number
  
  This number is a decimal number from 0 to \((2^{64})-1\), without delimiters. For example: \texttt{FIRSTREC(68508)}.

In either case, leading zeros can be omitted.

GSG\textit{(gsgname)}

Optional parameter you use to specify the GSG name of the IMS subsystem that produced the OLDS.

GSG is required if \texttt{LOGTOKEN} is specified.

INTERIM

Optional parameter you use to specify that an interim log data set record is to be created.

Before you specify \texttt{NOTIFY.PRILOG INTERIM}, a corresponding primary log record must exist.

LASTREC\textit{(number)}

Optional parameter you use to specify the log record sequence number of the last log record of the OLDS.

LASTREC is required for the OLDS \texttt{CLOSE} command. It is optional for the \texttt{SWITCH} command; if it is omitted, the FIRSTREC value minus 1, is recorded for the OLDS that is being closed. It is invalid for an \texttt{OPEN} command.

The log record sequence number can be one of the following:

- A hexadecimal number
  
  This number is 1 to 16 characters, enclosed in single quotes and preceded by the letter, X. For example: \texttt{LASTREC(X'10B9C')}.

- A decimal number
  
  This number is a decimal number from 0 to \((2^{64})-1\), without delimiters. For example: \texttt{LASTREC(68508)}.

In either case, leading zeros can be omitted.

LOGTOKEN\textit{(number)}

Optional parameter you use to specify the log token to be inserted into the PRILOG record and, if necessary, into the GSG record. It is valid only on an OLDS Open command.
NOTIFY.PRILOG (for OLDS)

Log tokens are numeric, assigned sequentially within PRILOG records for the same GSG, and used during recovery to ensure that all logs produced by members of the GSG have been included. The highest token assigned to any PRILOG is recorded in the GSG record.

The log token must satisfy all of the following conditions:

- Must be greater than that contained in the previous PRILOG record for the same GSG, if any.
- Must be less than that contained in the next PRILOG record for the same GSG, if any.
- Must not be more than one greater than the high PRILOG token contained in the specified GSG record.

NXTOLDS(ddname)

Optional parameter you use when RECON is to be updated to reflect an OLDS switch. The current OLDS is closed and an IMS online control region opens a new OLDS. ddname is the DD statement of the OLDS being opened. You specify the OLDS being closed with the OLDS(ddname) parameter. Use the DSN(name) parameter to specify the data set name of the OLDS being opened. Use the STARTIME(time_stamp) parameter to specify the close time of the OLDS being closed and the open time of the OLDS being opened.

SSID(name)

Optional parameter you use to specify the name of the IMS subsystem that created the log data set.

The SSID is an eight-character, alphanumeric string that represents a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. You use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

Examples of Using the NOTIFY.PRILOG (for OLDS) Command

Here are some examples of using the NOTIFY.PRILOG (for OLDS) command.

Example of Creating a PRIOLDS for an Online Subsystem OLDS: In this example, you create a PRIOLDS for an OLDS that belongs to IMS online subsystem IMSA.

```
//NFYPRILG JOB

; //SYSIN DD *
NOTIFY.PRILOG STARTIME(831230554321) -
DSN(IMS.OLDSBP13) OLDS(DFSOLP13) -
FIRSTREC(001) -
SSID(IMS)
/*
```

Example of Adding Information about Two Primary OLDSs to RECON: In this example, you create a PRIOLDS for two OLDSs that belong to the IMS online subsystem IMSA. Both OLDSs are closed. The first STARTIME parameter specifies the time stamp of the opening of the primary OLDS. The DSN parameter indicates that information that is added relates to the opening of the OLDS. NXTOLDS indicates an OLDS switch. The second STARTIME parameter and second DSN indicate the start time and DSN of the next OLDS. The third STARTIME parameter indicates the start time of the OLDS to be closed. The RUNTIME parameter is the time stamp of the closing volume.
Example of Creating a PRILOG to Record 2 OLDSs Opening and Closing: In this example, you create a PRILOG to record the opening and closing of two OLDSs. The new PRILOG follows an existing PRILOG record for GSG OURGRP, which contains a log token of 1. The three commands are, respectively, OPEN, SWITCH, and CLOSE.

```
NOTIFY.PRILOG SSID(IMSA)
  OLDS(OLDS001) DSN(IMS.OLDS.A01) STARTIME(930140830000) FIRSTREC(1)
  GSG(OURGRP) LOGTOKEN(2)

NOTIFY.PRILOG SSID(IMSA)
  OLDS(OLDS001)
  NXTOLDS(OLDS002) DSN(IMS.OLDS.A02) STARTIME(930140930000) FIRSTREC(5000)

NOTIFY.PRILOG SSID(IMSA)
  OLDS(OLDS002) STARTIME(930140930000)
  RUNTIME(930141030000) LASTREC(9999)
```

**NOTIFY.PRILOG (for OLDS)**

Use a NOTIFY.PRILOG command to add information about a primary RLDS (or a primary SLDS that a batch subsystem created) to RECON and to manually create interim-primary log data set records in RECON.

This is information that could not be added to RECON from the IMS system log processing exit routines. If you are processing DBDSs with IMS, you should not need to use this command under normal operating conditions. You must specify a NOTIFY.ALLOC command for each DBDS for which change records might exist on the primary RLDS being added.

This command adds or completes a data set entry in a PRILOG record. If you are modifying an existing completed data set entry, you should use the CHANGE.PRILOG(RLDS) command.

**NOTIFY.PRILOG (for RLDS) Syntax**

```
NOTIFY.PRILOG DSN(name) STARTIME(time_stamp) -
  RUNTIME(time_stamp)

CHKPTCT(value) CHKPTID(chkptid) FILESEQ(value)

FIRSTREC(number) GSG(gsname) INTERIM LASTREC(number)

LOGTOKEN(number) NONLOCAL RLDS SSID(name)
  UNIT(unittype)
```

Chapter 14. NOTIFY Commands 317
Parameters for NOTIFY.PRILOG (for RLDS)

**DSN**(*name*) | **RUNTIME**(*time stamp*)
---|---
Mutually exclusive, required parameters.

**DSN**
Specifies the data set name of the primary RLDS for which a log record is being created in RECON.

**RUNTIME**
Specifies the time stamp of a close or end-of-volume (EOV) operation for the specified primary RLDS. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

These two parameters are used in conjunction with the STARTIME and VOLSER parameters to identify what type of primary-recovery-log-data set entry is to be added to RECON.

Table 15 indicates which parameter combinations are required for each type of primary-recovery-log-data-set entry.

**Definitions:**

**RLDS Open**
Starting to use an RLDS and inserting a starting time.

**RLDS EOV**
RLDS end of volume (EOV) notification from the operating system. This is analogous to an OLDS Switch.

**RLDS Close**
Stopping the use of an RLDS (inserting a stop time)

<table>
<thead>
<tr>
<th>Type of Log Entry</th>
<th>Required Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>RLDS Open</td>
<td>STARTIME, DSN, VOLSER, FIRSTREC</td>
</tr>
<tr>
<td>RLDS EOV</td>
<td>STARTIME, VOLSER, RUNTIME</td>
</tr>
<tr>
<td>RLDS Close</td>
<td>STARTIME, RUNTIME, LASTREC</td>
</tr>
</tbody>
</table>

For each primary RLDS, you must issue a separate NOTIFY.PRILOG command for open, zero or more EOVs, and close.

**STARTIME**(*time stamp*)
Required parameter you use to specify the starting time of a primary RLDS.

The time stamp must be in standard form, see “Standard Time Stamp Format” on page 124.

See Table 15 for a description of the use of this parameter with other parameters in the NOTIFY.PRILOG command.

**CHKPTCT**(*value*)
Optional parameter you use to specify the number of checkpoints completed on the RLDS volumes.

The valid values for CHKPTCT are:
0 No checkpoints in the RLDS volume
1 A single checkpoint in the RLDS volume
2 More than one checkpoint in the RLDS volume

IMS uses the value of CHKPTCT to determine which logs are necessary to recover a Fast Path area with concurrent image copy.

**CHKPTID**(chkptid)
Optional parameter you use to specify the oldest checkpoint ID for an active partition specification table (PST) on an RLDS volume. The checkpoint ID must be in the standard form for a time stamp (see \[Standard Time Stamp Format\] on page 124).

**FILESEQ**(1\ | value)
Optional parameter you use to specify the file sequence number of the primary RLDS that is identified. You can specify this parameter only if you have specified the VOLSER parameter.

**FIRSTREC**(number)
Optional parameter you use to specify the log record sequence number of the first log record of the RLDS. For the first RLDS of the PRILOG, it corresponds to the first log record that was written during initialization of the IMS subsystem.

FIRSTREC is required if DSN is specified and is invalid if RUNTIME is specified.

The log record sequence number can be one of the following:
- A hexadecimal number
  This number is 1 to 16 characters, enclosed in single quotes and preceded by the letter, X. For example: `FIRSTREC(X'10B9C')`.
- A decimal number
  This number is a decimal number from 0 to \((2^{*}64)-1\), without delimiters. For example: `FIRSTREC(68508)`.

In either case, leading zeros can be omitted.

**GSG**(gsgname)
Optional parameter you use to specify the GSG name of the IMS subsystem that produced the RLDS.

GSG is required if NONLOCAL or LOGTOKEN is specified.

**INTERIM**
Optional parameter you use to specify that an interim log data set record is to be created. Before you issue the NOTIFY.PRILOG INTERIM command, you must create a corresponding primary recovery log record.

**LASTREC**(number)
Optional parameter you use to specify the log record sequence number of the last log record of the RLDS.

LASTREC is required if RUNTIME is specified and VOLSER is not specified (that is, on a \texttt{CLOSE} call). LASTREC is invalid if DSN is specified.

The log record sequence number can be one of the following:
- A hexadecimal number
  This number is 1 to 16 characters, enclosed in single quotes and preceded by the letter, X. For example: `LASTREC(X'10B9C')`.  

Chapter 14. NOTIFY Commands
NOTIFY.PRILOG (for RLDS)

- A decimal number
  This number is a decimal number from 0 to \((2^{64})-1\), without delimiters. For example: LASTREC(68508).

In either case, leading zeros can be omitted.

LOGTOKEN\((\text{number})\)
Optional parameter you use to specify the log token that is to be inserted into the PRILOG record and, if necessary, into the GSG record. It is valid only on an RLDS OPEN command; otherwise, it is ignored.

Log tokens are numeric, assigned sequentially within PRILOG records for the same GSG, and used during recovery to ensure that all logs produced by members of the GSG have been included. The highest token assigned to any PRILOG is recorded in the GSG record.

The log token must satisfy all of the following conditions:
- Must be greater than that contained in the previous PRILOG record for the same GSG, if any.
- Must be less than that contained in the next PRILOG record for the same GSG, if any.
- Must not be more than one greater than the high PRILOG token contained in the specified GSG record.

LOCAL | NONLOCAL
Mutually exclusive, optional parameters you use to specify where the RLDS data was originally created. LOCAL is used if the RLDS was created by an active IMS subsystem of the local service group. NONLOCAL is used if the RLDS was originally created by an active IMS subsystem of the non-local service group and transported to the tracking site.

LOCAL or NONLOCAL need only be specified when creating the PRILOG record. The LOCAL and NONLOCAL keywords are ignored on subsequent NOTIFY.PRILOG invocations for the PRILOG record.

If NONLOCAL is specified, none of the keywords CHKPTID, FILESEQ, UNIT, or VOLSER can be specified (the data sets must be cataloged) on any NOTIFY.PRILOG invocation for the PRILOG record.

RLDS
Optional parameter you use to specify that an RLDS record is to be created or updated.

SSID\((\text{name})\)
Optional parameter you use to specify the name of the IMS subsystem that created the log data set.

The SSID is an eight-character alphanumeric string that represents a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. You use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

UNIT\((3400 | \text{unittype})\)
Optional parameter you use to specify the unit type of the RLDSs. You only specify the UNIT parameter if you specify the DSN parameter. The unit type can be up to eight alphanumeric characters long.
VOLSER\text{(}volser\text{)}

Optional parameter you use to specify the volume serial number of the log volume being recorded for the identified primary RLDS. For an EOV notification, this volume serial number is that of the volume being started.

You must use the VOLSER parameter during RLDS open and EOV processing.

**Examples of Using the NOTIFY.PRILOG (for RLDS) Command**

Here are some examples of using the NOTIFY.PRILOG (for RLDS) command.

**Example of Adding Primary RLDS Information to RECON:** In this example, information about a primary RLDS is to be added to the RECON. The VOLSER and DSN parameters indicate that the information to be added relates to the opening of the primary RLDS. The STARTIME parameter specifies the time stamp of the opening of the primary RLDS. The first RUNTIME parameter specifies the time stamp of the EOV of the first volume of the primary RLDS. The second RUNTIME parameter specifies the time stamp of the closing volume of the primary RLDS.

```
//NFYPRILG JOB
:
//SYSIN DD *
NOTIFY.PRILOG RLDS STARTIME(820670201010) -
  VOLSER(VOL001) DSN(PRILOG1) FIRSTREC(001)
NOTIFY.PRILOG RLDS STARTIME(820670201010) -
  VOLSER(VOL002) RUNTIME(820670202020)
NOTIFY.PRILOG RLDS STARTIME(820670201010) -
  LASTREC(9999) RUNTIME(820670303030)
/*
```

**Example of Adding Interim-Primary RLDS Information to RECON:** In this example, information about the interim-primary RLDS is to be added to RECON. The STARTIME parameter specifies the time stamp of the opening of the interim primary RLDS.

```
//NFYPRILG JOB
:
//SYSIN DD *
NOTIFY.PRILOG RLDS STARTIME(822541234561) -
  DSN(DSNIRLDS) -
  VOLSER(VOL008) -
  FIRSTREC(077) -
  INTERIM
/*
```

**Example of Creating a PRILOG Record for 2 Tracking Log DSs:** In this example, the sequence of NOTIFY.PRILOG commands create a PRILOG record for two log data sets that were received at a tracking site.

```
NOTIFY.PRILOG RLDS DSN(RECEIVED.DSN1) STARTIME(911230405235) -
  NONLOCAL SSID(IMSA) GSG(MYGSG) FIRSTREC(1) -
  VOLSER(VOL003)
NOTIFY.PRILOG RLDS RUNTIME(911230500000) STARTIME(911230405235) -
  LASTREC(2376)
NOTIFY.PRILOG RLDS DSN(RECEIVED.DSN2) STARTIME(911230405235) -
  FIRSTREC(2377) VOLSER(VOL002)
NOTIFY.PRILOG RLDS RUNTIME(911230700000) STARTIME(911230405235) -
  LASTREC(4378)
```
NOTIFY.PRILOG (for SLDS and TSLDS)

Use a NOTIFY.PRILOG command to add information about a primary SLDS or TSLDS to RECON and to manually create interim-primary log data set records in RECON. This is information that could not be added to RECON from the IMS system log processing exit routines.

If you are processing DBDSs with IMS, you should not need to use this command under normal operating conditions. You must specify a NOTIFY.ALLOC command for each DBDS for which change records might exist on the primary SLDS being added.

This command adds or completes a data set entry in the PRISLD or PRITSLDS record. If you are modifying an existing completed data set entry, you should use the CHANGE.PRILOG (for SLDS) or CHANGE.PRILOG (for TSLDS) command.

When you issue a NOTIFY.PRILOG for a SLDS, a PRILOG record must exist for the corresponding RLDS. Use NOTIFY.PRILOG (for RLDS) to add information about a SLDS that a batch subsystem creates, because DBRC considers such a data set to be an RLDS.

NOTIFY.PRILOG (for SLDS and TSLDS) Syntax

```
NOTIFY.PRILOG SLDS [TSLDS] DSN(name) [RUNTIME(time_stamp)] STARTIME(time_stamp)
```

```
<table>
<thead>
<tr>
<th>CHKPTCT(value)</th>
<th>CHKPID(chkptid)</th>
<th>FILESEQ(value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRSTREC(number)</td>
<td>GSG(gsgname)</td>
<td>INTERIM</td>
</tr>
<tr>
<td>SSID(name)</td>
<td>UNIT(unittype)</td>
<td>VOLSER(volser)</td>
</tr>
</tbody>
</table>
```

Parameters for NOTIFY.PRILOG (for SLDS and TSLDS)

**SLDS**
Required parameter you use to specify that a record is to be created or updated for a SLDS.

**Important:** If you do not specify SLDS or TSLDS, the default is RLDS (see “NOTIFY.PRILOG (for RLDS)” on page 317).

**TSLDS**
Required parameter you use to specify that a record is to be created or updated for a SLDS on an RSR tracking subsystem.

**Important:** If you do not specify SLDS or TSLDS, the default is RLDS (see “NOTIFY.PRILOG (for RLDS)” on page 317).

**DSN(name) | RUNTIME(time_stamp)**
Mutually exclusive, required parameters.
DSN
Specifies the data set name of the primary SLDS or TSLDS for which a log record is being created in RECON.

RUNTIME
Specifies the time stamp of a close or EOV operation for the specified primary SLDS or TSLDS. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

These two parameters are used in conjunction with the STARTIME, FIRSTREC, LASTREC, and VOLSER parameters to identify what type of primary-system-log-data-set entry is to be added to RECON. Table 16 indicates which parameter combinations are required for each type of primary-system-log-data set entry.

Definitions:
SLDS Open
Starting to use an SLDS and inserting a starting time.

SLDS EOV
SLDS end of volume (EOV) notification from the operating system. This is analogous to an OLDS Switch.

SLDS Close
Stopping the use of an RLDS (inserting a stop time)

Table 16. Parameters of NOTIFY.PRILOG (SLDS or TSLDS) Command for Open, EOV, and Close

<table>
<thead>
<tr>
<th>Type of Log Entry</th>
<th>Required Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLDS Open</td>
<td>STARTIME, DSN, VOLSER, FIRSTREC</td>
</tr>
<tr>
<td>SLDS EOV</td>
<td>STARTIME, VOLSER, RUNTIME</td>
</tr>
<tr>
<td>SLDS Close</td>
<td>STARTIME, RUNTIME, LASTREC</td>
</tr>
</tbody>
</table>

For each primary SLDS or TSLDS, you must issue a separate NOTIFY.PRILOG command for open, zero or more EOVs, and close.

STARTIME(time_stamp)
Required parameter you use to specify the starting time of a primary SLDS or TSLDS. Use the log start time from the subsystem record or the PRILOG record. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

See Table 16 for a description of the use of this parameter with other parameters in the NOTIFY.PRILOG command.

CHKPTCT(value)
Optional parameter you use to change the number of checkpoints completed on the SLDS or TSLDS volumes. You specify a value for each SLDS or TSLDS volume that is designated.

The valid values for CHKPTCT are:
0   No checkpoints in the SLDS or TSLDS volume
1   A single checkpoint in the SLDS or TSLDS volume
2   More than one checkpoint in the SLDS or TSLDS volume
IMS uses the value of CHKPTCT to determine which logs are necessary to recover a Fast Path area with concurrent image copy.

**CHKPTID(chkptid)**
Optional parameter you use to specify the oldest checkpoint ID for an active PST on an SLDS or TSLDS volume. The checkpoint ID must be in the standard form for a time stamp (see “Standard Time Stamp Format” on page 124).

**FILESEQ( value)**
Optional parameter you use to specify the file sequence number of the primary SLDS or TSLDS that is identified. You specify this parameter only if you have also specified the VOLSER parameter.

**FIRSTREC(number)**
Optional parameter you use to specify the log record sequence number of the first log record of the SLDS or TSLDS. For the first SLDS of the PRISLD or TSLDS of the PRITSLDS, FIRSTREC corresponds to the first log record that was written during initialization of the IMS subsystem.

FIRSTREC is required if DSN is specified and is invalid if RUNTIME is specified.

The log record sequence number can be one of the following:
- A hexadecimal number
  This number is 1 to 16 characters, enclosed in single quotes and preceded by the letter, X. For example: FIRSTREC(X'10B9C').
- A decimal number
  This number is a decimal number from 0 to (2**64)-1, without delimiters. For example: FIRSTREC(68508).

In either case, leading zeros can be omitted.

**GSG(gsgname)**
Optional parameter you use to specify the GSG name of the IMS subsystem that produced the SLDS or TSLDS.

GSG is required if NONLOCAL is specified.

**INTERIM**
Optional parameter you use to specify that an interim log data set record is to be created.

**LASTREC(number)**
Optional parameter you use to specify the log record sequence number of the last log record of the SLDS or TSLDS.

LASTREC is required if RUNTIME is specified and VOLSER is not specified (that is, on a CLOSE call). LASTREC is invalid if DSN is specified.

The log record sequence number can be one of the following:
- A hexadecimal number
  This number is 1 to 16 characters, enclosed in single quotes and preceded by the letter, X. For example: LASTREC(X'10B9C').
- A decimal number
  This number is a decimal number from 0 to (2**64)-1, without delimiters. For example: LASTREC(68508).

In either case, leading zeros can be omitted.
LOCAL | NONLOCAL
Mutually exclusive, optional parameters you use to specify where the SLDS or TSLDS data was originally created. LOCAL is used if the SLDS or TSLDS was created by an active IMS subsystem of the local service group. NONLOCAL is used if the SLDS or TSLDS was originally created by an active IMS subsystem of the non-local service group and transported to the tracking site.

LOCAL or NONLOCAL need only be specified when creating the PRISLDS or PRITSLDS record. The LOCAL and NONLOCAL keywords are ignored on subsequent NOTIFY.PRIL0G invocations for the PRISLDS or PRITSLDS record.

If NONLOCAL is specified, none of the keywords CHKPTID, FILESEQ, UNIT, or VOLSER can be specified (the data sets must be cataloged) on any NOTIFY.PRIL0G invocation for the PRISLDS or PRITSLDS record.

SSID(name)
Optional parameter you use to specify the name of the IMS subsystem that created the log data set.

The SSID is an eight-character, alphanumeric string that represents a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. You use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

UNIT(3400 | unitype)
Optional parameter you use to specify the unit type of the SLDSs or TSLDSs. You only specify the UNIT parameter if you specify the DSN parameter. The unit type can be up to eight alphanumeric characters.

VOLSER(volser)
Optional parameter you use to specify the volume serial number of the log volume being recorded for the identified primary SLDS or TSLDS. For an EOV notification, this volume serial number is that of the volume being started.

Note: You must use the VOLSER parameter during SLDS or TSLDS open and EOV.

Example of Adding Primary SLDS Information to RECON
In this example, information about a primary SLDS is to be added to RECON. The VOLSER and DSN parameters indicate that the information to be added relates to the opening of the primary SLDS. The STARTIME parameter specifies the time stamp of the opening of the primary SLDS. The first RUNTIME parameter specifies the time stamp of the EOV of the first volume of the primary SLDS. The second RUNTIME parameter specifies the time stamp of the closing volume of the primary SLDS.

//NFYPRLG JOB

: //SYSIN DD *
NOTIFY.PRILOG SLDS STARTIME(820670201010) -
VOLSER(VOL004) DSN(PRIL0G4) FIRSTREC(7000)
NOTIFY.PRILOG SLDS STARTIME(820670201010) -
VOLSER(VOL005) RUNTIME(820670202020)
NOTIFY.PRILOG SLDS STARTIME(820670201010) -
RUNTIME(820670303030) LASTREC(8889)
/*
NOTIFY.RECOV

NOTIFY.RECOV

Use a NOTIFY.RECOV command to add information about recovery of a specified DBDS or DEDB area to RECON. You must use this command whenever you perform the recovery of a DBDS or DEDB area in any way other than using the Database Recovery utility (for example, by restoring the DASD volume on which the DBDS or area resides). In addition, you can notify DBRC when you recover a DBDS or DEDB area using the Database Recovery utility.

When specifying the RCVTIME parameter to inform DBRC of a time-stamp recovery, RECON must contain a record of the image copy data set that you used to restore the DBDS or DEDB area. The image copy record can be either a standard or a nonstandard image copy. If it is a nonstandard image copy, then its time stamp cannot fall within the range of an existing time-stamp recovery (the time between the RECOV TO and RUN times). The time stamp of the image copy record must be equal to that specified in the RCVTIME parameter of the NOTIFY.RECOV command.

In a data sharing environment, after you notify DBRC of a nonstandard recovery or an IMS recovery, DBRC turns off the recovery-needed flag and decreases the counter in the appropriate DBDS and DB records in RECON.

Related Reading: See [IMS V9: Operations Guide](#) for more information about recovery in a data sharing environment.

Restriction for HALDBs: This command is not allowed for ILDS or Index DBDSs of HALDB partitions.

NOTIFY.RECOV Syntax

```
NOTIFY.RECOV DBD(name) DDN(name) AREA(name) RCVTIME(time_stamp) RCVUSID(usid)
```

```
RUNUSID(usid) ADDN(name) RUNTIME(time_stamp) PITR TRACK
```

Parameters for NOTIFY.RECOV

**DBD(name)**

Required parameter you use to specify the database name of the DBDS or DEDB area.

**Restriction for HALDBs:** When using the NOTIFY.RECOV command for HALDBs, name must specify the name of a HALDB partition.

**DDN(name) | AREA(name)**

Mutually exclusive, required parameters you use to specify the ddname of the DBDS or DEDB area for which DBRC is to add the database recovery record to RECON.

**ADDN(name)**

Optional parameter you use to specify the ADS DD name of the ADS for which a Fast Path DEDB area recovery record is being added to RECON.

You can specify this parameter only when you specify the AREA(name) parameter.
CURRENT | RUNTIME(time_stamp)PITR
Mutually exclusive, optional parameters you use to specify the time stamp at which the DBDS or DEDB area was recovered.

CURRENT
Specifies that the current time stamp is to be used as the time stamp of the recovery. You can add the recovery information to RECON in a later step of the same job that performs the recovery if you specify CURRENT.

RUNTIME
Specifies the actual time stamp of a recovery of the DBDS or DEDB area. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

The optional PITR parameter specifies a point-in-time recovery. You must use the RCVTIME parameter if you use PITR.

DBDS | TRACK
Mutually exclusive, optional parameters you use to specify the type of IMS recovery that was done on a DBDS or DEDB area.

DBDS
Specifies that the recovery was a full recovery or a time-stamp recovery.

TRACK
Specifies that the recovery was a track recovery.

If you specify this parameter and the RCVTIME parameter, the command fails.

RCVTIME(time_stamp)
Optional parameter you use to specify the point in time to which the DBDS or DEDB area was restored. It can be any time when the DBDS or area was not being updated, that is, a time that is not covered by an active ALLOC record in RECON.

If you do not specify the RCVTIME parameter, you are notifying DBRC of a full recovery.

If you specify RCVTIME and the database or DEDB area is covered by RSR, you must also specify RCVUSID.

You must use RCVTIME if you use the RUNTIME parameter with the PITR parameter.

Restriction: Do not use the RCVTIME parameter when recovering from a nonstandard image copy at a tracking subsystem.

See 247 for more information on the RCVTIME parameter.

RCVUSID(usid)
Optional parameter you use to specify the effective update set identifier (USID) to which the DBD or DEDB area was recovered.

This parameter must be specified if the database or DEDB area is covered by RSR and RCVTIME was specified; it is not allowed if RCVTIME is not specified. The usid you use is the one in the listing of the IMAGE record.

RUNUSID(usid)
Optional parameter you use to specify the current update set identifier (usid) at the time the database or DEDB area was recovered. If this is a receive, RUNUSID is the usid of the image copy that was used for recovery.

RUNUSID must be specified for recovery of an RSR-covered database or DEDB area.
Example of Adding DBDS Recovery Information to RECON

In this example, information about recovery of a specified DBDS is to be added to RECON. The RUNTIME parameter specifies the time stamp of the recovery of the DBDS. The PITR parameter specifies a point-in-time recovery. The RCVTIME parameter specifies the time stamp to which the specified DBDS was recovered.

```//NFYRECOV JOB
  ...
  //SYSIN DD * NOTIFY.RECOV DBD(DB1) DDN(DDN1) -
  RUNTIME(971351015366) -
  RCVTIME(971350905297) -
  PITR
/*
```

After the command is executed, a listing of the RECON shows the RECOV record, as illustrated below.

```
RECOV
RUN = 1997.135 10:15:36.6 -08:00 * RUN USID = 0000000005
RECOV TO= 1997.135 09:05:29.7 -08:00 RECOV TO USID = 0000000004
POINT-IN-TIME
```

NOTIFY.REORG

Use a NOTIFY.REORG command to add a record to RECON about the reorganization of the database to which an identified DBDS belongs.

The information in the reorganization record is used by DBRC to determine which image copy data sets, change accumulation data sets, and log data sets are valid as input to a subsequent recovery of the identified DBDS.

Restrictions:

- This command should not be used following the reorganization of a Fast Path DEDB. Such databases can be recovered after a reorganization.
- This command turns on the flag needed by the image copy process in the DBDS record. You must either run an image copy or issue the CHANGE.DBDS ICOFF command to turn off the flag.
- This command can also be used to record in RECON the equivalent of an image copy data set that was created for the HISAM Reorganization Reload utility. Use this command only if you are using these logs as an image copy data set.
- All optional parameters except CURRENT or RUNTIME apply only to the image copy data set that was created as part of the processing by the HISAM Reorganization Reload utility.
- You must specify a NOTIFY.REORG command for each DBDS in the database that was reorganized. A DD statement for the IMS DBDLIB data set must be provided in the job stream of the NOTIFY.REORG command.
- The NOTIFY.REORG command, and database reorganization in general, are invalid for databases at an RSR tracking site.
- The NOTIFY.REORG command is not allowed for ILDS or Index DBDSs of HALDB partitions.

NOTIFY.REORG Syntax

```
NOTIFY.REORG—DBD(name)—DDN(name)—CURRENT—RUNTIME(time_stamp)
```
Parameters for NOTIFY.REORG

**DBD**(name)

Required parameter you use to identify the database name of the DBDS that was reorganized.

**Restriction for HALDBs:** When using the NOTIFY.REORG command for HALDBs, name must specify the name of a HALDB partition.

**DDN**(name)

Required parameter you use to identify the data set ddname of the DBDS that was reorganized.

**CURRENT | RUNTIME**(time_stamp)

Mutually exclusive, optional parameters you use to specify the time stamp of the reorganization of the identified DBDS.

**CURRENT**

Specifies that the current time stamp is to be placed in the reorganization record. You can specify a NOTIFY.REORG command as a later step in the same job that performs the reorganization if you specify CURRENT.

**RUNTIME**

Specifies that the actual time stamp of the reorganization is to be placed in the reorganization record. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

**FILESEQ**(1 | value)

Optional parameter you use to specify the file sequence number of the identified DBDS that was included in the logs that were used as input to a run of the HISAM Reorganization Reload utility. The description of the ICDSN parameter contains information about the log data set with which this parameter is associated. You use this parameter only when the VOLLIST parameter has also been specified.

**FILESEQ2**(1 | value)

Optional parameter you use to specify the file sequence number of the identified DBDS when it was included in the logs that were used as input to the HISAM Reorganization Reload utility. The description of the ICDSN2 parameter contains information about the log data set with which this parameter is associated. You use this parameter only when the VOLLIST parameter has also been specified.
parameter contains information about the log data set with which this parameter is associated. You use this parameter only when the VOLLIST2 parameter has also been specified.

ICDSN(name)
Optional parameter you use to specify the data set name of the image copy data set that was created as part of a HISAM reorganization of a database. (If you reorganized your database using the HISAM Reorganization Reload utility, the logs that were used as input to that utility can be used as image copy data sets.)

You can specify an ICDSN parameter only if the corresponding DBDS was identified to RECON with the NOREUSE attribute in an INIT.DBS command.

ICDSN2(name)
Optional parameter you use to specify the data set name of the duplicate image copy data set that was created as part of a HISAM reorganization of a database. (If you reorganized your database using the HISAM Reorganization Reload utility, the logs that were used as input to that utility can be used as image copy data sets.)

You can specify an ICDSN2 parameter only if you have specified an ICDSN parameter.

OFFLINE | ONLINE
Mutually exclusive, optional parameters that specify whether the reorganization is a HALDB OLR or an offline reorganization.

OFFLINE
Specifies that the DBDS was reorganized offline.

ONLINE
Specifies that the reorganization is a HALDB OLR.

STOPTIME(time_stamp)
Optional parameter that specifies the time when a HALDB OLR completed successfully.

NORECOV | RECOV
Mutually exclusive, optional parameters that specify whether the online reorganization can be used as input for recovery.

NORECOV
Specifies that the HALDB OLR cannot be used as input to recovery.

RECOV
Specifies that the HALDB OLR may be used as input to recovery.

RECDCT(value)
Optional parameter you use to specify the number of records that are contained in the identified DBDS. value must be a decimal number from 1 to 2147483647.

UNIT(3400 | unittype)
Optional parameter you use to specify the unit type of the volumes on which the image copy data set resides. The unit type can be up to eight alphanumeric characters.

UNIT2(3400 | unittype)
Optional parameter you use to specify the unit type of the volumes on which the duplicate image copy data set resides. The unit type can be up to eight alphanumeric characters.
VOLLIST(volser)
Optional parameter you use to specify the volume serial numbers of the volumes on which the image copy data set identified by the ICDSN keyword resides. You can specify up to 255 volume serial numbers for volser; each volser can be up to six alphanumeric characters.

VOLLIST2(volser)
Optional parameter you use to specify the volume serial numbers of the volumes on which the duplicate image copy data set, identified by the ICDSN2 parameter, resides. You can specify up to 255 volume serial numbers for volser; each can be up to six alphanumeric characters.

USID(value)
Optional parameter you use to specify the update set identifier of the database or area when the reorganization occurred.

USID is required if the database or area is assigned to a global service group. If the database or area is not assigned to a GSG, USID is optional.

Example Adding DBDS Reorganization Information to RECON
In this example, information about a reorganization of a specified DBDS is to be added to RECON. The DBDLIB data set is specified, because DBRC requests a search of it to verify that the reorganization occurred. The names of two image copy data sets for the reorganized DBDS are given. They both follow the data set naming convention, and a list of volumes is provided for both image copy data sets.

```
//NFYREORG JOB

//IMS DD  DSN=IMS.DBDLIB,DISP=SHR
//SYSIN DD *
   NOTIFY.REORG DBD(001)  DDN(001) -
      ICDSN(IMS.DB1.DD1.IC.ICDSN) -
      VOLLIST(VOL001,VOL002,VOL003) -
      FILESEQ(4) -
      ICDSN2(IMS.DB1.DD1.IC2.ICDSN2) -
      VOLLIST2(VOL004,VOL005,VOL006,VOL007) -
      FILESEQ2(4)  RECDCT(12345)
/*
```

NOTIFY.SECLOG (for OLDS)
Use a NOTIFY.SECLOG command to add information about a secondary OLDS to RECON and to manually create an ISECOLDS record in RECON. RECON must already contain a PRIOLDS record with the same SSID and STARTIME. This is information that could not be added from the IMS log data exit routines. This command is not normally required.

NOTIFY.SECLOG (for OLDS) Syntax

```
NOTIFY.SECLOG(OLDS(ddname)) DSN(name) STARTIME(time_stamp) RUNTIME(time_stamp)
```

```
FIRSTREC(number)  GSG(gsname)  INTERIM  LASTREC(number)
```

```
NXTOLDS(ddname)  SSID(name)
```
Parameters for NOTIFY.SECLOG (for OLDS)

**OLDS(ddname)**
Required parameter you use to specify that a record is to be created or updated in RECON for the OLDS.

*ddname* is the ddname that the IMS online control region used when it used the OLDS.

**DSN(name) | RUNTIME(time_stamp)**
Mutually exclusive, required parameters.

**DSN**
Specifies the data set name of the secondary OLDS for which an online log record is being created in RECON.

**RUNTIME**
Specifies the time stamp of an open or close operation of the specified secondary OLDS. The time stamp must be in standard form (see "Standard Time Stamp Format" on page 124).

DSN and RUNTIME are used in conjunction with the STARTIME, FIRSTREC, LASTREC, and NXTOLDS parameters to identify which type of secondary online-log-data-set entry is to be added to RECON. Table 17 indicates the keyword combinations that correspond to the type of secondary-online-log data-set entry.

Table 17. Parameters of NOTIFY.SECLOG (for OLDS) Command for Open, Switch, and Close

<table>
<thead>
<tr>
<th>Type of Online Log Entry</th>
<th>Required Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLDS Open</td>
<td>STARTIME, DSN, FIRSTREC</td>
</tr>
<tr>
<td>OLDS Switch</td>
<td>FIRSTREC, STARTIME, DSN, NXTOLDS</td>
</tr>
<tr>
<td>OLDS Close</td>
<td>LASTREC, STARTIME, RUNTIME</td>
</tr>
</tbody>
</table>

For each secondary OLDS, you must issue a separate NOTIFY.SECLOG command for open, switch, and close operations.

**STARTIME(time_stamp)**
Required parameter you use to specify the starting time of a secondary OLDS. The time stamp must be in standard form (see "Standard Time Stamp Format" on page 124).

See Table 17 for a description of the use of this parameter with other parameters in the NOTIFY.SECLOG command.

**FIRSTREC(number)**
Optional parameter you use to specify the log record sequence number of the first log record of the OLDS. For the first OLDS of the SECLOG, FIRSTREC corresponds to the first log record that was written during initialization of the IMS subsystem.

FIRSTREC is required for OLDS OPEN and SWITCH commands. It specifies the first log record sequence number on the OLDS being opened. It is invalid for a CLOSE command.

The log record sequence number can be one of the following:
- A hexadecimal number
  This number is 1 to 16 characters, enclosed in single quotes and preceded by the letter, X. For example: FIRSTREC(X'10B9C').
NOTIFY.SECLOG (for OLDS)

- A decimal number
  This number is a decimal number from 0 to \((2^{*64})-1\), without delimiters. For example: \texttt{FIRSTREC(68508)}.

In either case, leading zeros can be omitted.

\textbf{GSG}(\texttt{gsiname})
Optional parameter you use to specify the GSG name of the IMS subsystem that produced the OLDS.

GSG is required if LOGTOKEN is specified.

\textbf{INTERIM}
Optional parameter you use to specify that an interim log data set record is to be created. Before you create an interim log data set, you must create a secondary OLDS.

\textbf{LASTREC}(\texttt{number})
Optional parameter you use to specify the log record sequence number of the last log record of the OLDS.

LASTREC is required for the OLDS \texttt{CLOSE} command. It is optional for the \texttt{SWITCH} command; if it is omitted, the FIRSTREC value minus 1, is recorded for the OLDS being closed. It is invalid for an \texttt{OPEN} command.

The log record sequence number can be one of the following:
- A hexadecimal number
  This number is 1 to 16 characters, enclosed in single quotes and preceded by the letter, \texttt{X}. For example: \texttt{LASTREC(\texttt{X'}10B9C')}.
- A decimal number
  This number is a decimal number from 0 to \((2^{*64})-1\), without delimiters. For example: \texttt{LASTREC(68508)}.

In either case, leading zeros can be omitted.

\textbf{NXTOLDS}(\texttt{ddname})
Optional parameter you use when RECON is to be updated to reflect an OLDS switch. The current OLDS is closed and an IMS online control region opens a new OLDS. \texttt{ddname} is the DD statement of the OLDS being opened. You specify the OLDS being closed with the OLDS(\texttt{ddname}) parameter. Use the DSN parameter to specify the data set name of the OLDS being opened. Use the STARTIME parameter to specify the close time of the OLDS being closed and the open time of the OLDS being opened.

\textbf{SSID}(\texttt{name})
Optional parameter you use to specify the name of the IMS subsystem that created the log data set.

The SSID is an eight-character alphanumeric string that represents a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. You use the \texttt{INIT.RECON} or \texttt{CHANGE.RECON} command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

\textbf{Examples of Using the NOTIFY.SECLOG (for OLDS) Command}
Here are some examples of using the NOTIFY.SECLOG (for OLDS) command.
Example of Creating the ISECOLDS Record That Corresponds to the OLDS: In this example, an ISECOLDS record corresponding to the OLDS is created.

```//NFYSECLG JOB
:
//SYSIN DD *
NOTIFY.SECLOG SSID(IMSA) OLDS(DFSOLS03) -
  DSN(IMS.INTERIM.LOG) -
  STARTIME(823220522348) -
  INTERIM
/*
```

Example of Creating a SECOLDS Record for 2 Secondary OLDSs: In this example, you create a SECOLDS record for two secondary OLDSs that belong to IMS online subsystem IMSA. Both secondary OLDSs are closed. The first STARTIME parameter specifies the time stamp of the opening of the primary OLDS. The DSN parameter indicates that information added relates to the opening of the OLDS. NXTOLDS indicates an OLDS switch. The second STARTIME parameter and second DSN indicate the start time and DSN of the next OLDS. The third STARTIME parameter indicates the start time of the OLDS to be closed. The RUNTIME parameter is the time stamp of the closing volume.

```NOTIFY.SECLOG SSID(IMSA) STARTIME(812171212120) OLDS(DFSOLS01) -
  DSN(IMS.OLSS01)
NOTIFY.SECLOG SSID(IMSA) STARTIME(812181212120) OLDS(DFSOLS01) -
  DSN(IMS.OLSS02) NXTOLDS(DFSOLS02)
NOTIFY.SECLOG SSID(IMSA) STARTIME(812181212120) OLDS(DFSOLS02) -
  RUNTIME(932191010101)
```

**NOTIFY.SECLOG (for RLDS)**

Use a NOTIFY.SECLOG command to add information about a secondary RLDS to RECON and to manually create an ISECLOG record in RECON. RECON must already contain a PRILOG with the same SSID and STARTIME. This is information that could not be added from the IMS log data exit routines. This command is not normally required.

This command adds or completes a data set entry in the Primary Log record. If you are modifying an existing completed data set entry, you should use the CHANGE.SECLOG (RLDS) command.

**NOTIFY.SECLOG (for RLDS) Syntax**

```
NOTIFY.SECLOG DSN(name) STARTIME(time_stamp) RUNDIME(time_stamp)
```

```
CHRPTCT(value) CHKPTID(chkptid) FILESEQ(value)
```

```FIRSTREC(number) GSG(gsgname) INTERIM LASTREC(number) LOCAL
```

```RLDS SSID(name) UNIT(unittype) VOLSER(volser)
```

```3400
```
Parameters for NOTIFY.SECLOG (for RLDS)

DSN(name) | RUNTIME(time_stamp)
Mutually exclusive, required parameters.

DSN
Specifies the data set name of the secondary RLDS for which a recovery log record is being created in RECON.

RUNTIME
Specifies the time stamp of an open, close, or EOV operation of the specified secondary RLDS. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

DSN and RUNTIME are used in conjunction with the STARTIME and VOLSER parameters to identify which type of secondary recovery log data set entry is to be added to RECON. Table 18 indicates the keyword combinations that correspond to the type of secondary recovery log data set entry.

Table 18. Parameters of NOTIFY.SECLOG (for RLDS) Command for Open, EOV, and Close

<table>
<thead>
<tr>
<th>Type of Recovery Log Entry</th>
<th>Required Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>RLDS Open</td>
<td>STARTIME, DSN, VOLSER</td>
</tr>
<tr>
<td>RLDS EOV</td>
<td>STARTIME, VOLSER, RUNTIME</td>
</tr>
<tr>
<td>RLDS Close</td>
<td>STARTIME, RUNTIME</td>
</tr>
</tbody>
</table>

For each secondary RLDS, you must issue a separate NOTIFY.SECLOG command for open, zero or more ends-of-volume, and close processing.

STARTIME(time_stamp)
Required parameter you use to specify the starting time of a secondary RLDS. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

If you issue a subsequent STARTIME parameter, that time is the start time of the volume. See Table 18 for a description of the use of this parameter with other parameters in the NOTIFY.SECLOG command.

CHKPTCT(0 | value)
Optional parameter you use to change the number of checkpoints completed on the RLDS volumes.

The valid values for CHKPTCT are:

- 0     No checkpoints in the RLDS volume
- 1     A single checkpoint in the RLDS volume
- 2     More than one checkpoint in the RLDS volume

IMS uses the value of CHKPTCT to determine which logs are necessary to recover a Fast Path area with concurrent image copy.

CHKPTID(chkptid)
Optional parameter you use to specify the oldest checkpoint ID for an active PST on an RLDS volume. The checkpoint ID must be in standard form for a time stamp (see “Standard Time Stamp Format” on page 124).

FILESEQ(1 | value)
Optional parameter you use to specify the file sequence number of the secondary RLDS that is identified. You specify this parameter only if you have also specified the VOLSER parameter.
**FIRSTREC**(number)
Optional parameter you use to specify the log record sequence number of the first log record of the RLDS. For the first RLDS of the SECLOG, it corresponds to the first log record that was written during initialization of the IMS subsystem.

FIRSTREC is required if DSN is specified and is invalid if RUNTIME is specified.

The log record sequence number can be one of the following:
- A hexadecimal number
  This number is 1 to 16 characters, enclosed in single quotes and preceded by the letter, X. For example: `FIRSTREC(X’10B9C’)`.
- A decimal number
  This number is a decimal number from 0 to \((2^{64})-1\), without delimiters. For example: `FIRSTREC(68508)`.

In either case, leading zeros can be omitted.

**GSG**(gsgname)
Optional parameter you use to specify the GSG name of the IMS subsystem that produced the RLDS.

GSG is required if NONLOCAL is specified.

**INTERIM**
Optional parameter you use to specify that an interim log data set record is to be created.

**LASTREC**(number)
Optional parameter you use to specify the log record sequence number of the last log record of the RLDS.

LASTREC is required if RUNTIME is specified and VOLSER is not specified (that is, on a Close call). LASTREC is invalid if DSN is specified.

The log record sequence number can be one of the following:
- A hexadecimal number
  This number is 1 to 16 characters, enclosed in single quotes and preceded by the letter, X. For example: `LASTREC(X’10B9C’)`.
- A decimal number
  This number is a decimal number from 0 to \((2^{64})-1\), without delimiters. For example: `LASTREC(68508)`.

In either case, leading zeros can be omitted.

**LOCAL | NONLOCAL**
Mutually exclusive, optional parameters you use to specify where the RLDS data was originally created. LOCAL is used if the RLDS was created by an active IMS subsystem of the local service group. NONLOCAL is used if the RLDS was originally created by an active IMS subsystem of the non-local service group and transported to the tracking site.

LOCAL or NONLOCAL need only be specified when creating the SECLOG record. The LOCAL and NONLOCAL keywords are ignored on subsequent NOTIFY.SECLOG invocations for the SECLOG record.
NOTIFY.SECLOG (for RLDS)

If NONLOCAL is specified, none of the keywords CHKPTID, FILESEQ, UNIT, or VOLSER can be specified (the data sets must be cataloged) on any NOTIFY.SECLOG invocation for the SECLOG record.

**RLDS**
Optional parameter you use to specify that a record is to be created or updated in RECON for an IMS RLDS.

**SSID(name)**
Optional parameter you use to specify the name of the IMS subsystem that created the log data set.

The SSID is an eight-character alphanumeric string that represents a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. You use the INIT, RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

**UNIT(3400 | unitype)**
Optional parameter you use to specify the unit type of the volumes on which the secondary RLDSs reside. You only specify the UNIT parameter if you specify the DSN parameter. The unit type can be up to eight alphanumeric characters.

**VOLSER(volser)**
Optional parameter you use to specify the volume serial number of the recovery log volume that is being recorded for the identified secondary RLDS. For an EOV notification, this volume serial number is that of the volume being started. Table 18 on page 335 identifies when you use the VOLSER parameter.

**Examples of Using the NOTIFY.SECLOG (for RLDS) Command**
Here are some examples of using the NOTIFY.SECLOG (for RLDS) command.

**Example of Adding Secondary RLDS Information to RECON:** In this example, information about a secondary RLDS is to be added to RECON. The STARTIME parameter identifies the secondary RLDS by its opening time stamp. The VOLSER and DSN parameters indicate that the information to be added relates to the opening of the primary RLDS. The first RUNTIME parameter specifies the time stamp of the EOV of the secondary RLDS. The second RUNTIME parameter specifies the time stamp of the closing of the secondary RLDS.

```
//NFYSECLG JOB

; //SYSIN DD *
NOTIFY.SECLOG RLDS STARTIME(820670201010) -
   DSN(DSN003) VOLSER(VOL001)
NOTIFY.SECLOG RLDS STARTIME(820670201010) -
   RUNTIME(820680204500) VOLSER(VOL002)
NOTIFY.SECLOG RLDS STARTIME(820670201010) -
   RUNTIME(820682030000)
/*
```

**Example of Adding Interim Secondary RLDS Information to RECON:** In this example, information about the interim secondary RLDS is to be added to RECON. The STARTIME parameter specifies the time stamp of the opening of the interim-secondary RLDS, and the RUNTIME parameter specifies the time stamp of the closing of the interim-secondary RLDS.
NOTIFY.SECLOG (for RLDS)

//NFYSECLG JOB
:
://SYSIN DD *
   NOTIFY.SECLOG RLDS RUNTIME(822561630000)-STARTIME(822541234561) INTERIM
/*

NOTIFY.SECLOG (for SLDS and TSLDS)

Use a NOTIFY.SECLOG command to add information about a secondary SLDS or TSLDS to RECON and to manually create an ISECSLDS record in RECON. RECON must already contain a primary log data set record with the same SSID and STARTIME. This is information that could not be added from the log data exists of the IMS system. This command is not normally required.

This command adds or completes a data set entry in the Primary or Secondary Log record. If you are modifying an existing completed data set entry, use the CHANGE.SECLOG (for SLDS) command or the CHANGE.SECLOG (for TSLDS) command.

NOTIFY SECLOG (for SLDS and TSLDS) Syntax

```
NOTIFY.SECLOG SLDS DSN(name) | TSLDS RUNTIME(time_stamp)

  CHKPTCT(value) | CHKPTID(chkptid) | FILESEQ(value)

  FIRSTREC(number) | GSG(gsgname) | INTERIM | LASTREC(number)

  SSID(name) | UNIT(unittype) | VOLSER(volser)
```

Parameters for NOTIFY SECLOG (for SLDS and TSLDS)

**SLDS**

Required parameter you use to specify that an SLDS record is to be created.

If you do not specify SLDS or TSLDS, the default is RLDS. See "NOTIFY.PRILOG (for RLDS)” on page 317 for more information on the RLDS parameter.

**TSLDS**

Required parameter you use to specify that a TSLDS record is to be created.

If you do not specify SLDS or TSLDS, the default is RLDS. See "NOTIFY.PRILOG (for RLDS)” on page 317 for more information on the RLDS parameter.

**DSN(name) | RUNTIME(time_stamp)**

Mutually exclusive, required parameters.

**DSN**

Specifies the data set name of the secondary SLDS or TSLDS for which a system log record is being created in RECON.
NOTIFY.SECLOG (for SLDS and TSLDS)

RUNTIME
Specifies the time stamp of an open, close, or EOV operation of the specified secondary SLDS. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

DSN and RUNTIME are used in conjunction with the STARTIME and VOLSER parameters to identify which type of secondary system log data set entry is to be added to RECON. Table 19 indicates the keyword combinations that correspond to the type of secondary system log data set entry.

Table 19. Parameters of NOTIFY.SECLOG (for SLDS or TSLDS) Command for Open, EOV, and Close

<table>
<thead>
<tr>
<th>Type of System Log Entry</th>
<th>Required Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLDS Open</td>
<td>STARTIME, DSN, VOLSER</td>
</tr>
<tr>
<td>SLDS EOV</td>
<td>STARTIME, VOLSER, RUNTIME</td>
</tr>
<tr>
<td>SLDS Close</td>
<td>STARTIME, RUNTIME</td>
</tr>
</tbody>
</table>

For each secondary SLDS or TSLDS, you must issue a separate NOTIFY.SECLOG command for open, zero or more ends-of-volume, and close processing.

STARTIME(time_stamp)
Required parameter you use to specify the starting time of a secondary SLDS or TSLDS. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

See Table 19 for a description of the use of this parameter with other parameters in the NOTIFY.SECLOG command.

CHKPTCT(value)
Optional parameter you use to change the number of checkpoints completed on the SLDS or TSLDS volumes.

The valid values for CHKPTCT are:

0 No checkpoints in the SLDS or TSLDS volume
1 A single checkpoint in the SLDS or TSLDS volume
2 More than one checkpoint in the SLDS or TSLDS volume

IMS uses the value of CHKPTCT to determine which logs are necessary to recover a Fast Path area with concurrent image copy.

CHKPTID(chkptid)
Optional parameter you use to specify the oldest checkpoint ID for an active PST on an SLDS or TSLDS volume. The checkpoint ID must be in standard form for a time stamp (see “Standard Time Stamp Format” on page 124).

FILESEQ(value)
Optional parameter you use to specify the file sequence number of the secondary SLDS or TSLDS that is identified. You specify this parameter only if you specify the VOLSER parameter.

FIRSTREC(number)
Optional parameter you use to specify the log record sequence number of the first log record of the SLDS or TSLDS. For the first SLDS or TSLDS of the SECSLD or SECTSLDS, FIRSTREC corresponds to the first log record that was written during initialization of the IMS subsystem.
FIRSTREC is required if DSN is specified and is invalid if RUNTIME is specified.

The log record sequence number can be one of the following:

- A hexadecimal number
  This number is 1 to 16 characters, enclosed in single quotes and preceded by the letter, X. For example: `FIRSTREC('10B9C')`.

- A decimal number
  This number is a decimal number from 0 to \((2^{64})-1\), without delimiters. For example: `FIRSTREC(68508)`.

In either case, leading zeros can be omitted.

**GSG(gsgname)**
Optional parameter you use to specify the GSG name of the IMS subsystem that produced the SLDS or TSLDS.

GSG is required if NONLOCAL is specified.

**INTERIM**
Optional parameter you use to specify that an interim log data set record is to be created.

**LASTREC(number)**
Optional parameter you use to specify the log record sequence number of the last log record of the SLDS.

LASTREC is required if RUNTIME is specified and VOLSER is not specified (that is, on a close call). LASTREC is invalid if DSN is specified.

The log record sequence number can be one of the following:

- A hexadecimal number
  This number is 1 to 16 characters, enclosed in single quotes and preceded by the letter, X. For example: `LASTREC('10B9C')`.

- A decimal number
  This number is a decimal number from 0 to \((2^{64})-1\), without delimiters. For example: `LASTREC(68508)`.

In either case, leading zeros can be omitted.

**LOCAL | NONLOCAL**
Mutually exclusive, optional parameters you use to specify where the SLDS or TSLDS data was originally created. LOCAL is used if the SLDS or TSLDS was created by an active IMS subsystem of the local service group. NONLOCAL is used if the SLDS or TSLDS was originally created by an active IMS subsystem of the non-local service group and transported to the tracking site.

LOCAL or NONLOCAL need only be specified when creating the SECSLD or SECTSLDS record. The LOCAL and NONLOCAL keywords are ignored on subsequent NOTIFY.SECLOG invocations for the SECSLD or SECTSLDS record.

If NONLOCAL is specified, none of the keywords CHKPTID, FILESEQ, UNIT, or VOLSER can be specified (the data sets must be cataloged) on any NOTIFY.SECLOG invocation for the SECSLD or SECTSLDS record.

**SSID(name)**
Optional parameter you use to specify the name of the IMS subsystem that created the log data set.
The SSID is an eight-character alphanumeric string that represents a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. You use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

**UNIT**

Optional parameter you use to specify the unit type of the volumes on which the secondary SLDSs reside. You only specify the UNIT parameter if you specify the DSN parameter. The unit type can be up to eight alphanumeric characters.

**VOLSER**

Optional parameter you use to specify the volume serial number of the system log volume that is being recorded for the identified secondary SLDS. For an EOV notification, this volume serial number is that of the volume being started. Table 19 on page 339 identifies when you use the VOLSER parameter.

### Example of Adding Secondary SLDS Information to RECON

In this example, information about a secondary SLDS is to be added to RECON. The STARTIME parameter identifies the secondary SLDS by its opening time stamp. The VOLSER and DSN parameters indicate that the information to be added relates to the opening of the secondary SLDS. The first RUNTIME parameter specifies the time stamp of the EOV of the secondary SLDS or TSLDS. The second RUNTIME parameter specifies the time stamp of the closing of the secondary SLDS or TSLDS.

```bash
//NFYSECLG JOB

//SYSIN DD *

NOTIFY.SECCLOG SLDS STARTIME(820670201010) -
SSID(IMSC) DSN(DSN006) VOLSER(VOL009)
NOTIFY.SECCLOG SLDS STARTIME(820670201010) -
RUNTIME(820680204500) VOLSER(VOL003)
NOTIFY.SECCLOG SLDS STARTIME(820670201010) -
RUNTIME(820682030000)
/*
```

### NOTIFY.SUBSYS

Use a NOTIFY.SUBSYS command to create a subsystem entry in RECON. A check is made to ensure that a subsystem entry for the specified subsystem does not exist in RECON. This command is not normally required.

**NOTIFY.SUBSYS Syntax**

```
       NOTIFY.SUBSYS—SSID(name)
       IRLMID(name) NORMAL ONLINE ENDRECOV STARTRCV
```  

**Parameters for NOTIFY.SUBSYS**

**SSID(name)**

Required parameter you use to specify the name of the subsystem for which information is to be added. The SSID is an eight-character alphanumeric string that represents a valid z/OS and IMS subsystem identification name.
NOTIFY.SUBSYS

IRLMID(name)
Optional parameter you use to specify the name of the IRLM with which the subsystem is communicating. The IRLMID is a five-character alphanumeric string.

If IRLMID is not specified, the subsystem is not using an IRLM.

NORMAL | ABNORMAL
Mutually exclusive, optional parameters you use to specify the status of the subsystem.

NORMAL
Specifies that the previous run of the subsystem ended normally and that the subsystem is to continue normal processing.

ABNORMAL
Specifies that the previous run of the subsystem ended abnormally and recovery processing is required.

ONLINE | BATCH
Mutually exclusive, optional parameters you use to specify the type of subsystem from which notification is made.

ONLINE
Specifies that notification is made from an online subsystem.

BATCH
Specifies that notification is made from a batch subsystem.

STARTRCV | ENDRECOV
Mutually exclusive, optional parameters you use to specify the sign-on state of the subsystem.

STARTRCV
Specifies that the subsystem has signed on for recovery-start processing.

ENDRECOV
Specifies that the subsystem has signed on normally or that a sign-on recovery-complete call was successful.

Example of Adding a New Subsystem Record to RECON
In this example, a new subsystem record identified by the SSID parameter is added to RECON. In addition, the subsystem record is marked as online.

```
//NOTIFYSS JOB
;
//SYSIN DD *
NOTIFY.SUBSYS SSID(IMS34) ONLINE
/*
```

NOTIFY.UIC
Use a NOTIFY.UIC command to add information to RECON about a nonstandard image copy data set related to the DBDS or DEDB area that is identified in the command. A nonstandard image copy data set is one that was not created by the supported image copy utility such as, one created via a tape dump of the DASD volume that contains the identified DBDS or DEDB area. Using the NOTIFY.UIC command is the only way you can record in RECON the existence of nonstandard image copy data sets.

You cannot issue this command for a DBDS defined with the REUSE attribute.
Restrictions:

- A nonstandard image copy data set cannot be used as input to the Database Recovery utility (see Chapter 2, “Database Recovery Control Utility (DSPURX00),” on page 27 for more information).
- This command is not allowed for ILDS or Index DBDSs of HALDB partitions.
- The NOTIFY.UIC command will fail if it is attempting to record an image copy of a HALDB that has an online reorganization in progress. This is because an image copy of a HALDB cannot be taken while the HALDB is being reorganized online and the reorganization has not completed (either it is in progress or was terminated).

**NOTIFY.UIC Syntax**

```
NOTIFY.UIC DBD(name) DDN(name) AREA(name) CURRENT RUNTIME(time_stamp) UDATA('string')
```

**Parameters for NOTIFY.UIC**

**DBD(name)**

Required parameter you use to specify the database name of the DBDS or DEDB area for which the nonstandard image copy data set was created.

**Restriction for HALDBs**: When using the NOTIFY.UIC command for HALDBs, `name` must specify the name of a HALDB partition.

**DDN(name) | AREA(name)**

Mutually exclusive, required parameters you use to identify, by its name, the DBDS or DEDB area for which the nonstandard image copy data set was created.

**CURRENT | RUNTIME(time_stamp)**

Mutually exclusive, optional parameters you use to specify the time stamp of the creation of the nonstandard image copy data set.

**CURRENT**

Specifies that the current time stamp is to be used as the time stamp of the creation of the specified image copy data set. You can create the nonstandard image copy data set and record its creation in RECON as separate steps of a single job if you specify CURRENT.

**RUNTIME**

Specifies the actual time stamp of the creation of the identified nonstandard image copy data set. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 124).

**UDATA('string')**

Optional parameter you use to specify up to 80 bytes of information about the identified, nonstandard image copy data set. You can use the variable field of this parameter to describe how the nonstandard image copy data set was created.

**Restrictions** `string` must:

- Be in character format in order to be visible in a listing of the RECON data set.
• Be enclosed in parentheses.
• Appear in single quotation marks.

**USID***(number)*

Optional parameter you use to specify the value of the update set identifier of the database or area when the image copy data set was created.

USID is required if the database or area is assigned to a GSG.

**Example of Adding Nonstandard ICDSN Information to RECON**

In this example, information about a nonstandard image copy data set is to be added to RECON. The RUNTIME parameter specifies the time stamp of the creation of the nonstandard image copy data set. The UDATA parameter specifies the user data to be recorded in the record in RECON that is updated by this command.

```plaintext
//NFYUIC JOB

//SYSIN DD *
NOTIFY.UIC DBD(DB1) DDN(DD1) -
    RUNTIME(820670201010) -
    UDATA('DUMP OF VOLUME VOL001 AT 820670201010')
/*
```

```
```
Chapter 15. RESET Command

Use a RESET.GSG command to remove obsolete recovery information from an original active site RECON.

In This Chapter:
* "RESET.GSG"

RESET.GSG

"RESET.GSG Usage" contains the details for the RESET.GSG command.

RESET.GSG Usage

Use a RESET.GSG command after an unplanned RSR takeover to remove obsolete recovery information about RSR-covered databases from the original active site RECON. For each database or area that is assigned to the specified global service group (GSG), all IC, UIC, ALLOC, RECOV, and REORG records are deleted. In addition, all subsystem records of the GSG and related database authorization information along with all OLDS, RLDS, or SLDS records are deleted. Affected CA records are cleaned up, made available, or deleted as needed.

Before deleting the obsolete information a backup copy of the RECON is created (RESET.GSG issues an internal BACKUP.RECON command). BACKUP.RECON invokes the z/OS AMS REPRO command, with its normal defaults, in order to create the backup copy. Any restrictions applicable to the normal use of the REPRO command apply also to this command.

Attention: Note that if your RECON RECORDSIZE is greater than 32K that IDCAMS REPRO can handle the RECORDSIZE as long as the output data set is NOT a sequential file (such as a tape file). Keeping the RECONS on DASD works well.

If any failure occurs while the RESET.GSG command is processing, but after the backup copy has been created (as indicated by the REPRO Completion message), use the following procedure:
1. Correct the condition that caused the failure.
2. Restore the RECON from the backup data set.
3. Delete and re-allocate the backup data set.
4. Reissue the RESET.GSG command.

If RESET.GSG fails before the backup copy is complete, follow the same procedure, omitting the third step.

This command fails if RSRFEAT=NO is specified in the IMSCTRL macro.

RESET.GSG Syntax

/SM590000/SM590000
RESET.GSG GSGNAME(gsgname) BOTH
RECON1
RECON2

© Copyright IBM Corp. 1974, 2006
Parameters for RESET.GSG

GSGNAME(gsname)
   Required parameter you use to specify the GSGNAME.

BOTH | RECON1 | RECON2
   Mutually exclusive, required parameters you use to specify which copy of the RECON to use.

   BOTH
      Indicates that the RECON is to be copied to the data sets specified by the BACKUP1 and BACKUP2 DD statements.

   RECON1
      Indicates that the RECON is to be copied to the data set specified by the BACKUP1 DD statement.

   RECON2
      Indicates that the RECON is to be copied to the data set specified by the BACKUP2 DD statement.

Example of the RESET.GSG Command
In the example, the RECON is copied to the BACKUP1 data set, and then the obsolete information is deleted.

```// JOB
  ...
//BACKUP1 DD ...
//SYSIN DD *
RESET.GSG GSGNAME(IMSGSG1) RECON1
/*```
Chapter 16. Using the DBRC API

The DBRC application programming interface (API) is a release-independent, assembler macro interface to DBRC that customer-written application programs use to obtain services from DBRC. The application obtains these services by issuing DBRC API requests to DBRC, and DBRC returns the results to an area in storage where the application can retrieve them. The services available through the DBRC API are to:

- Initialize the API and start DBRC (see Chapter 19, “DBRC Start Request,” on page 423)
- Obtain information that DBRC maintains in the RECON data sets (see Chapter 17, “DBRC Query Request,” on page 357)
- Return buffer storage (see Chapter 18, “DBRC Release Buffer Request,” on page 419)
- Shutdown DBRC and the API (see Chapter 20, “DBRC Stop Request,” on page 427)

The API is provided with IMS in the DSPAPI macro. A sample application program (DSPAPSMP) that uses the DBRC API is included in the IMS.ADFSSMPL (also known as IMS.SDFSSMPL) library.

**Prerequisite Knowledge:** To write a program that uses the DBRC API, you must have a working knowledge of:

- Assembler language programming
- z/OS and the services it supplies
- IMS
- DBRC

**Related Reading:** For information about assembler language programming, see High Level Assembler for MVS & VM & VSE Programmer’s Guide and High Level Assembler for MVS & VM & VSE Language Reference.

**In This Chapter:**
- “Writing an Application to Access the DBRC API”
- “Runtime Considerations of the DBRC API” on page 353

**Writing an Application to Access the DBRC API**

The general structure of an application that uses the DBRC API is as follows:

1. Include the API DSECTS (DSPAPI FUNC=DSECT)
2. Initialize the API, start DBRC, establish a connection to the RECON data sets, and receive the API token (DSPAPI FUNC=STARTDBRC)
3. Issue one or more Query requests (DSPAPI FUNC=QUERY)
4. Process the information returned from the Query requests
5. Return buffer storage (DSPAPI FUNC=RELBUF)
6. Terminate DBRC and the DBRC API (DSPAPI FUNC=STOPDBRC)

The following sections discuss some of the programming considerations for applications that access the DBRC API.
• “Establishing the DBRC API Environment”
• “Ending the DBRC API Environment”
• “Addressing and Residency Mode When Using the DBRC API”
• “Address Space Control (ASC) Mode and State When Using the DBRC API”
• “How the DBRC API Uses Registers”
• “Including EQU Statements In Your DBRC API Application”
• “Coding Parameters In Your DBRC API Application” on page 351
• “Macro Versions of the DBRC API Macro” on page 351
• “The DBRC API Token” on page 352
• “Macro Forms of the DSPAPI Macro” on page 352

Establishing the DBRC API Environment
Your application program establishes the API environment using the DSPAPI FUNC=STARTDBRC macro. This macro function initializes the API, creates a connection to the DBRC RECON data sets, and returns the API token. This token must be specified for all subsequent API calls. See “The DBRC API Token” on page 352 for information about the API token.

Ending the DBRC API Environment
After it has completed its work with DBRC, your application program must end the API environment using the DSPAPI FUNC=STOPDBRC macro. This macro function allows DBRC to terminate, deallocates any data sets that DBRC had allocated, and frees storage allocated to the API environment.

Addressing and Residency Mode When Using the DBRC API
Your application must invoke the DBRC API in 31-bit addressing mode and can reside either above or below the 16 MB line. Parameter addresses passed to the API can be above or below the 16 MB line unless otherwise stated in a macro description. Parameter addresses returned to the user by the API likewise can be above or below the 16 MB line.

Address Space Control (ASC) Mode and State When Using the DBRC API
Your application must be in primary ASC mode to invoke the API services. Cross memory mode is not supported. The application can run in either problem state or supervisor state, and can also run as an APF authorized program.

How the DBRC API Uses Registers
General purpose registers 0, 1, 14, and 15 can be changed by the API. Register 13 must contain the address of a standard (18 word) save area. The contents of registers 2-13 are unchanged by the API.

Including EQU Statements In Your DBRC API Application
Use the DSECT function of a DSPAPI request to include equate (EQU) statements and constants used by the DSPAPI macro in your program. The syntax for the DSECT function of a DSPAPI request is:

```c
DSPAPI FUNC=DSECT
```
Coding Parameters In Your DBRC API Application

Parameters specified on the DSPAPI macro are either literals, addresses, or values. You can pass an address or a parameter value by using a register, a symbol, or a literal.

Use a register

To use a register, you must load the address of a symbol or value into one of the general purpose registers, then use that register (enclosed in parentheses) for the parameter in the DSPAPI request. Use only registers in the range 2-12. Register notation does not work with MF=L because this form does not produce executable code.

Here is an example of passing an address using a register:

```
LA 5,OUTPUTAD
DSPAPI FUNC=QUERY,OUTPUT=(5),...
.
.
OUTPUTAD DS A
```

This example generates the following instruction:

```
ST 5,DSPAPI_Plist_Output
```

Use a symbol

To use a symbol name, you must define a named area of storage that either contains the desired value, or will receive a returned address or value, then use that symbol name for the parameter in the DSPAPI request.

Here is an example of using a symbol name:

```
DSPAPI FUNC=QUERY,OUTPUT=OUTPUTAD,..
.
.
OUTPUTAD DS A
```

This example generates the following instructions:

```
LA 0,OUTPUTAD
ST 0,DSPAPI_Plist_Output
```

Use a literal

You can use literals for certain parameter values, for example time stamps. To use a literal, pass the literal as a hex string for the parameter in the DSPAPI request.

Here is an example of using a literal:

```
DSPAPI FUNC=QUERY,TYPE=LOG,
STARTIME==XL12'1980030F191212009999028D'
```

This example generates the following instructions:

```
LA 0,=XL12'198030F191212009999028D'
ST 0,DSPAPI_Plist_Output
```

Name Fields

Unless specifically noted, name fields are 8-character fields, left justified and padded with blanks.

Macro Versions of the DBRC API Macro

The functions provided by the DSPAPI macro have a version associated with them because parameter lists can change from one release to the next (parameter lists are
discussed in “Macro Forms of the DSPAPI Macro”. To use the parameters associated with a version, you must specify that version number or a later version. If you specify an earlier version level, the parameter is not accepted by the macro and an MNOTE error message is issued at assembly time. If parameters have a version dependency, the parameter descriptions with each request type identify the version number required.

The DBRC API Token

The API token is a four-byte field used to relate a series of API requests. It is returned to your program when a DSPAPI FUNC=STARTDBRC macro is issued. This token must be supplied on all other macro calls associated with this instance of STARTDBRC. The token is no longer valid after a DSPAPI FUNC=STOPDBRC macro call.

Macro Forms of the DSPAPI Macro

There are four different macro forms for the DSPAPI macro: Standard (S), List (L), Modify (M), and Execute (E), with two variations, COMPLETE and NOCHECK. The List, Modify, and Execute forms are usually used in combinations when writing reentrant programs or when the application issues multiple requests. The following are explanations about when and why to use each form.

Standard Form

Use the standard form of the macro (MF=S or MF=S,list) to generate and modify an inline parameter list. If your program is reentrant, do not use the standard form of the macro because reentrant code cannot be modified. With few exceptions, if you use the standard form of the macro in writing reentrant code, the execution of the code results in an abend. The standard form of the macro does three things. It:

• Creates an inline parameter list
• Modifies the parameter list with the parameters specified on the request
• Sends the request to the API

Using the standard form, you can optionally assign a label to the generated parameter list by specifying MF=(S,list) where list specifies the name of the label assigned to the parameter list created by this form of the macro. The standard MF=S form of the macro is the default.

List Form

Use the list form of the macro (MF=L,list) to generate a labeled, inline parameter list. This list is populated with the parameter values specified on the macro. The list form does not modify an existing list and does not send a request to the API. In effect, the list form creates a template that can be used as the target of a modify form or an execute form (the real list you plan to use). If the parameter list is generated in reentrant code, it cannot be modified. Therefore, you must obtain enough storage to hold the parameter list and use this storage as the target of the modify or list form.

list specifies the name of the label assigned to the parameter list created by this form of the macro.

Note: The expansion of the list form does not contain any executable instructions. Therefore, you cannot use register notation in the list form.

Modify Form

Use the modify form of the macro (MF=M,list,COMPLETE | NOCHECK)
to change the values specified on the macro in the parameter list specified by the list parameter. The modify form does not generate a parameter list and does not issue requests to DBRC.

| list | Can be a symbol or a general purpose register in the range 2 to 12 and specifies the address of the parameter list to be modified. |

**COMPLETE**

Specifies that DBRC will use defaults (for the parameters that have defaults) for all parameters that are not in the list and will perform validity checking for the parameters that are specified in the list. COMPLETE is the default.

**NOCHECK**

Specifies that the parameter list has been checked and that DBRC should use the list as is. No validity checking is performed.

**Execute Form**

Use the Execute form of the macro (MF=E,list,COMplete | NOCHECK) to:

- Modify a parameter list (generated by the List form) with new and additional allowable parameters you might not have specified on the List form
- Issue a request to the API

You can change the parameters on the macro with each subsequent invocation of the Execute form of the macro.

| list | Can be a symbol or a general purpose register in the range 2 to 12 and specifies the address of the parameter list to be used. |

**COMPLETE**

Specifies that DBRC will use defaults (for the parameters that have defaults) for all parameters that are not in the list and will perform validity checking for the parameters that are specified in the list. COMPLETE is the default.

**NOCHECK**

Specifies that the parameter list has been checked and that DBRC should use the list as is. No validity checking is performed.

---

**Runtime Considerations of the DBRC API**

The following sections discuss the runtime considerations for the DBRC API.

- "Accessing the DSPAPI Macro"
- "Accessing the RECON Data Sets"
- "Time-Stamp Format for DBRC Requests" on page 354
- "How DBRC Uses the Output Data Set" on page 355
- "Using the API In an IMSplex" on page 355

**Accessing the DSPAPI Macro**

Allocate IMS.SDFSRESL as either a JOBLIB or STEPLIB in your program’s JCL because the API (DSPAPI) is distributed with IMS.

**Accessing the RECON Data Sets**

While the API is running, it uses up to 3 RECON data sets allocated by the DD names RECON1, RECON2, and RECON3. See Chapter 4, “Initializing and
Maintaining the RECON Data Set,” on page 55 for a complete understanding of the RECONs. The RECONs can be allocated:

- Dynamically by the API when the DSPAPI FUNC=STARTDBRC request is issued (recommended)
- Dynamically by your application program
- Through your JCL

If the API allocates the RECONs, they are deallocated when the DSPAPI FUNC=STOPDBRC request is issued. Before the API can allocate the RECONs, MDA members must have been created for RECON1, RECON2, and RECON3. See “Dynamic Allocation Macro (DFSMDA)” in the IMS V9: Utilities Reference: System for more information about creating MDA members. These members must exist in either a library allocated as //IMSDALIB or in the //JOBLIB or //STEPLIB libraries. The API searches in //IMSDALIB (if it exists) first for the MDA members.

The IMS libraries concatenated to JOBLIB or STEPLIB are usually APF Authorized. If your program runs APF Authorized and the library containing the MDA members is not APF Authorized, allocate it using IMSDALIB as the DD name.

**Requirement:** All jobs accessing a set of RECON data sets must allocate the same data set by the same DD name. For example, RECON1=dsn1, RECON2=dsn2, and RECON3=dsn3. Failure to follow this convention causes serious damage to the RECONs.

The API uses only one set of RECONs between a FUNC=STARTDBRC request and its associated FUNC=STOPDBRC request. Your program, however, can use multiple sets of RECONs by deallocating the RECONs after the FUNC=STOPDBRC request and then allocating a new set of RECONs before issuing the next FUNC=STARTDBRC request. In order for this to work, your program must dynamically allocate the RECONs.

### Time-Stamp Format for DBRC Requests

The format for time stamps associated with DBRC requests (either input or output) are in the following packed decimal UTC time format:

`yyyydddffhmmssstxxxxxfqqs`

Where:

- `yyyy` is the year (0000 to 9999)
- `ddd` is the day (000 to 366)
- `F` is a hexadecimal character for padding purposes (X'F')
- `hh` is the hour (0 to 23)
- `mm` is the minute (0 to 59)
- `ss` is the second (0 to 59)
- `t` is the tenth of a second (0 to 9)
- `xxxxx` low-order digits ignored by DBRC
- `fqqs` Time zone offset where:
  - `f` Flag bits, normally zero for UTC representation
  - `qq` Quarter hours (32/4=8)
  - `s` Sign (D is negative, C is positive)

DBRC ignores the value of the offset field in time stamps provided on Query requests.
How DBRC Uses the Output Data Set

While running under the API, DBRC might put out messages and other information to a data set defined by the DD name SYSPRINT, or by a DD name you specify. This output data set might be on tape, DASD, a printer, or routed through the output stream (SYSOUT). The data set might be allocated by your program through JCL or dynamically allocated prior to invoking the API. If your program already uses the DD name SYSPRINT, you can specify an alternate DD name for the API to use (see Chapter 19, “DBRC Start Request,” on page 423). The attributes for this data set are RECFM=FBA, LRECL=121. Do not specify the blocksize because DBRC will use a system determined blocksize (the system determines an optimal blocksize for the device).

Using the API In an IMSplex

If the set of RECONs used by your job belong to an IMSplex, you must either pass the PLEX name as a parameter on your EXEC JCL statement or use the SCI Registration Exit (DSPSCIX0). To specify the parameter on your EXEC statement, code IMSPLEX=imsplex_name. For example:

```
//STEP1 EXEC PGM=MYPROG,PARM='MYPARM1=EXAMPLE,IMSPLEX=PLEX1'
```

You do not need to specify PARM if your installation uses the SCI Registration Exit (DSPSCIX0). See IMS V9: Customization Guide for more information about this exit.
Chapter 17. DBRC Query Request

Use the DBRC Query request (DSPAPI FUNC=QUERY) along with the TYPE parameter to retrieve the following types of information from the RECON:

- **Backout (TYPE=BACKOUT)** - See “Backout Query” on page 359
- **Database (TYPE=DB)** - See “Database Query” on page 363. This variation of QUERY returns database registration and status information for:
  - Full Function databases
  - Fast Path databases
  - HALDB databases
  - DBDS or area information and supporting recovery-related information for each DBDS or area (allocation, image copy, recovery, and reorganization)
- **Group and member information** - See “Group Query” on page 386 for the following group types:
  - Change Accumulation (TYPE=CAGROUP). CA execution information can also be returned.
  - DBDS (TYPE=DBDSGROUP).
  - Database (TYPE=DBGROUP).
  - Recovery (TYPE=RECOVGROUP).
  - Global Service Group (TYPE=GSGROUP).
- **Log, Recovery and System Log Data Set (TYPE=LOG)** - See “Log Query” on page 395.
- **Online Log Data Set (TYPE=OLDS)** - See “OLDS Query” on page 404.
- **RECON status (TYPE=RECON)** - See “RECON Status Query” on page 409. This variation of QUERY returns RECON header information, as well as the status of the RECON configuration.
- **Subsystem (TYPE=SUBSYS)** - See “Subsystem Query” on page 413.

**Related Reading:** See “Records in the RECON Data Set” on page 64 for a description of the information available in the RECON data set, and “Fields Present in a Listing of a RECON by Record Type” on page 528 for a description of the individual fields present in each RECON record type.

**In This Chapter:**
- “Output from Query Requests”
- “Query Requests Reference” on page 358

**Output from Query Requests**

The requested information is returned to the calling application in a chain of one or more blocks in storage. The pointer to the beginning of this chain is returned in the area specified by the OUTPUT parameter of the Query request.

The storage for these blocks is not preallocated by the calling application. DBRC will acquire private storage for these blocks. It is the responsibility of the calling application to free this storage using the Buffer Release request (DSPAPI FUNC=RELBUF). For more information about freeing this storage, see Chapter 18, “DBRC Release Buffer Request” on page 419.
Figure 15 illustrates the general format of the output from a Query request. The sample DSECT that follows describes in detail the fields of the storage blocks and their relationship to each other.

Each storage block begins with a standard header mapped by DSPAPQHD, as illustrated in Figure 16. The following sections contain an overview of, the syntax for, and parameter descriptions for each query request.

In this section:
- “Backout Query” on page 359
- “Database Query” on page 363
- “Group Query” on page 386
- “Log Query” on page 395
- “OLDS Query” on page 404
- “RECON Status Query” on page 409
- “Subsystem Query” on page 413
Backout Query

The Backout query (DSPAPI FUNC=QUERY TYPE=BACKOUT) request is used to retrieve backout information from RECON for a specific subsystem or all subsystems.

Syntax for the TYPE=BACKOUT Query Request

```
name DSPAPI FUNC=QUERY TYPE=BACKOUT TOKEN=address

SSID=* MF=S
SSID=subsystem_ID MF=(S,list)
SSID=subsystem_ID MF=(L,list)
SSID=subsystem_ID MF=(M,list)
SSID=subsystem_ID MF=(E,list)

OUTPUT=output RETCODE=returncode RSNCODE=reasoncode

VERSION=1.0 VERSION=number
```

Parameters for the TYPE=BACKOUT Query Request

The following are the possible parameters for QUERY TYPE=BACKOUT.

- **name**      Optional symbol you can specify. If used, begins in column 1.
- **TYPE=BACKOUT**
  Specifies that backout information is requested.
- **TOKEN=**    Required parameter that specifies the address of a 4-byte field that was returned on the FUNC=STARTDBRC request.
- **SSID=**     Specifications subsystem name for the backout being queried. The default value (SSID=*) requests all of the backout information.
- **MF=S | L | M | E**
  Optional parameter that specifies the macro form of the request. See "Macro Forms of the DSPAPI Macro" on page 352 for the details of the parameter values.
- **OUTPUT=** Required parameter that specifies a field to receive a pointer to the first block of backout information blocks. See "Output for TYPE=BACKOUT QUERY Request" on page 361 for a detailed description of the information blocks returned.

The output address is zero if no output was built. This can occur if nothing in the RECON satisfies the request or if an error occurs before any output could be built.

The storage for the output blocks is not preallocated. DBRC acquires private storage for these blocks. The requesting application is responsible...
for freeing this storage by using the Buffer Release request (DSAPI Func=RELBUF), specifying the returned output address.

**RETCODE**=symbol | (2 - 12)
Optional parameter that specifies a place in storage to receive the return code. If specified as a symbol, the symbol must be the label of a word of storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the return code is placed in register 15.

**RSNCODE**=symbol | (2 - 12)
Optional parameter that specifies a place in storage to receive the reason code. If specified as a symbol, the symbol must be the label of a word of storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the reason code is placed in register 0.

**VERSION**=1.0 | number
Optional parameter that specifies the version number of the parameter list to be generated by this request.

To use the parameters associated with a version, you must specify the number of that version or a later version. If you specify an earlier version level, the parameter is not accepted for processing and an error message is issued at assembly time. If parameters have a version dependency, the parameter descriptions with each request type identify the version number required.

The default version is 1.0.

**Return and Reason Codes for the TYPE=BACKOUT Query Request**

Table 20 contains most of the return and reason codes for TYPE=BACKOUT query requests. The other possible return and reason codes relate to DBRC, not the query request. For a list of these return and reason codes, see “DBRC Service Aids” in the [IMS V9: Diagnosis Guide and Reference](#).

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000000'</td>
<td>X'00000000'</td>
<td>Request completed successfully.</td>
</tr>
</tbody>
</table>

**Warning**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000008'</td>
<td>X'D87000001'</td>
<td>No backout records exist.</td>
</tr>
</tbody>
</table>

**Severe Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'000000C'</td>
<td>X'C9000001'</td>
<td>Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.</td>
</tr>
</tbody>
</table>

**Storage Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000028'</td>
<td>X'D87000001'</td>
<td>Error obtaining storage for BACKOUT block.</td>
</tr>
<tr>
<td>X'00000028'</td>
<td>X'D91000001'</td>
<td>An error occurred processing the request. DBRC will release storage that was obtained up to this point. However, another error was encountered during the attempt to release storage.</td>
</tr>
</tbody>
</table>

**Internal Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0000002C'</td>
<td>DBRC internal error. Refer to “Database Recovery Control Service Aids” in the <a href="#">IMS V9: Diagnosis Guide and Reference</a> for further information.</td>
</tr>
</tbody>
</table>

**Parameter Error**
Table 20. Return and Reason Codes for TYPE=BACKOUT Query Requests (continued)

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000002'</td>
<td>Invalid TOKEN field address. The address of the field containing the API TOKEN failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000003'</td>
<td>Invalid RETCODE field address. The address of the field to contain the API RETCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000004'</td>
<td>Invalid RSNCODE field address. The address of the field to contain the API RSNCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000005'</td>
<td>Invalid OUTPUT field address. The address of the field to contain the OUTPUT address failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D80000001'</td>
<td>Parameter error. Missing or invalid OUTPUT parameter.</td>
</tr>
</tbody>
</table>

Output for TYPE=BACKOUT QUERY Request

Figure 17 illustrates the format of the output from a QUERY TYPE=BACKOUT request. Figure 18 on page 362 is a sample DSECT describes in detail the fields of the storage blocks and their relationship to each other.
The following structure maps the unit of recovery entries. There is one such entry for each unit of recovery (i.e. there are `apqbo_UORcount` entries). Each unit of recovery entry contains the offset within the backout block to the previous and following entries. Field `apqbo_PrevUOR` is the offset of the previous entry and `apqbo_NextUOR` is the offset of the following entry. For the first unit of recovery (UOR) entry, `apqbo_PrevUOR` will be zero. Similarly, `apqbo_NextUOR` will be zero for the last entry. Addressability to the first UOR entry is given by:

```
rfy apqbo_UORentry based(addr(DSPAPQBO) + apqbo_FirstUOR)
```

Addressability to the last UOR entry is given by:

```
rfy apqbo_UORentry based(addr(DSPAPQBO) + apqbo_LastUOR)
```

Addressability to the next UOR entry, if one exists (that is: `apqbo_NextUOR` not equal to 0), is given by:

```
rfy apqbo_UORentry based(addr(DSPAPQBO) + apqbo_NextUOR)
```

Similarly, to address the previous entry (when `apqbo_PrevUOR` not equal to 0):

```
rfy apqbo_UORentry based(addr(DSPAPQBO) + apqbo_PrevUOR)
```

Once addressability has been established to a UOR entry, addressability to the `i`th database for this UOR is given by:

```
rfy apqbo_DBentry based(addr(apqbo_UORentry) + apqbo_DBoffset + (i-1) apqbo_DBLength)
```

---

**Figure 18.** DSECT of DSPAPQBO (Part 1 of 2)
### QUERY TYPE=DB

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>STRUCTURE</td>
<td>64</td>
<td>APQBO_UORENTRY</td>
<td>Unit of Recovery entry</td>
</tr>
<tr>
<td>0</td>
<td>CHARACTER</td>
<td>64</td>
<td>APQBO_PREFIX</td>
<td>Prefix section</td>
</tr>
<tr>
<td>0</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQBO_NEXTUOR</td>
<td>Offset of next UOR entry</td>
</tr>
<tr>
<td>4</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQBO_PREVUOR</td>
<td>Offset of previous UOR entry</td>
</tr>
<tr>
<td>8</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQBO_DBOFFSET</td>
<td>Offset to DB entries</td>
</tr>
<tr>
<td>12</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQBO_UORTIME</td>
<td>Time stamp for this UOR</td>
</tr>
<tr>
<td>24</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQBO_UORPSB</td>
<td>PSB name</td>
</tr>
<tr>
<td>32</td>
<td>BIT(16)</td>
<td>2</td>
<td>APQBO_UORFLAGS</td>
<td>Deferred backout = dynamic backout failure</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>CHARACTER</td>
<td>6</td>
<td>*</td>
<td>Reserved</td>
</tr>
<tr>
<td>40</td>
<td>CHARACTER</td>
<td>16</td>
<td>APQBO_RTOKN</td>
<td>Recovery token</td>
</tr>
<tr>
<td>40</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQBO_RSSSID</td>
<td>SSID for this token</td>
</tr>
<tr>
<td>48</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQBO_UORID</td>
<td>Unique UOR ID</td>
</tr>
<tr>
<td>56</td>
<td>SIGNED</td>
<td>4</td>
<td>APQBO_DBCOUNT</td>
<td>Number of DBs for this UOR</td>
</tr>
<tr>
<td>60</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQBO_DbLENGTH</td>
<td>Length of each DB entry</td>
</tr>
<tr>
<td>62</td>
<td>CHARACTER</td>
<td>2</td>
<td>*</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

### Database Query

The Database Query request (DSPAPI FUNC=QUERY TYPE=DB) is used to retrieve information from the RECON concerning one or more registered databases. This information includes the following database types:

- Full Function
- Fast Path DEDB
- HALDB (returns information about the HALDB master and all of its partitions)

Optionally, you can request data set and area information. If you do, you have the option of requesting recovery-related information for the data set or area. This includes allocation, image copy, recovery, and reorganization information.

### Syntax for the TYPE=DB Query Request

```
DSPAPI—FUNC=QUERY—TYPE=DB—TOKEN=address
```

Chapter 17. DBRC Query Request 363
Parameters for the TYPE=DB Query Request
The following are the possible parameters for QUERY TYPE=DB.

**name**  Optional symbol you can specify. If used, begins in column 1.

**TYPE=DB**
Specifies that RECON information for one or more databases is requested.

**TOKEN=symbol (2 - 12)**
Required parameter that specifies the address of a 4-byte field that was returned on the FUNC=STARTDBRC request.

**DBNAME=symbol (2 - 12)**
Specifies the name of the database whose information is being queried.
This can be the name of a Full Function database, a Fast Path DEDB, or a HALDB.

You must specify a DBNAME or DBLIST if you do not specify LOC=FIRST.

**DBLIST=symbol (2 - 12)**
Specifies the list of database names whose information is being queried.
Each name in the list can be the name of a Full Function database, a Fast Path DEDB, or a HALDB.

The list consists of a header and one or more eight-character list entries.
The header consists of a fullword containing the number of entries in the list. The name entries are left justified and are padded with blanks.

You must specify a DBNAME or DBLIST if you do not specify LOC=FIRST.

**DDN=symbol (2 - 12) * NULL**
Optional parameter that specifies the DD name of the data set or area. An asterisk (*) without quotes) can be specified to request information about all DBDSs or areas. If DBLIST contains more than one database, specifying a specific DD name is ignored and treated as if DDN=* was specified.
Information is returned for all DBDSs or areas. The same is true if a specific ddname is requested with a HALDB master specified in the DBNAME parameter.

If you specify DDN=NULL, no DBDS or area information is returned.

DDN=NULL is the default.

**LIST=(ALLOC | IC | RECOV | REORG | ALL | LIST=NONE**

Optional parameter that specifies the type (or types) of supporting information to be included in the query output for the specified DBDS or area. If DDN is not specified, this information is returned for all DBDSs or areas in the database. One or more of the specific values, separated by commas, can be included in the list: ALLOC (allocation records), IC (image copy records), RECOV (recovery records), or REORG (reorganization records). LIST=ALL specifies that all supporting information is requested.

If you specify LIST=NONE, no supporting information is returned.

LIST=NONE is the default.

**LOC-FIRST | NEXT | SPEC**

Optional parameter that specifies that the request is for the specified, the first, or the next database defined in the RECON.

DBNAME=dname is required when you specify LOC=NEXT, but it is not allowed for LOC-FIRST.

Databases are in alphanumeric order. The next database might not necessarily be of the same type as the database name specified in the DBNAME parameter. The value in the DBNAME parameter is used as the base of the search and does not need to be a name of a database registered in the RECON.

LOC=SPEC is the default.

**MF=S | L | M | E**

Optional parameter that specifies the macro form of the request. See "Macro Forms of the DSPAPI Macro" on page 352 for the details of the parameter values.

**OUTPUT=symbol | (2 - 12)**

Required parameter that specifies a field to receive a pointer to the first block of a possible chain of database information blocks. See "Output for TYPE=DB Query Request" on page 368 for a detailed description of the information blocks returned.

The output address is zero if no output was built. This can occur if nothing in the RECON satisfies the request or if an error occurs before any output could be built.

The storage for the output blocks is not preallocated. DBRC acquires private storage for these blocks. The requesting application is responsible for freeing this storage by using the Buffer Release request (DSPAPI FUNC=RELBUF), specifying the returned output address.

**RETCODE=symbol | (2 - 12)**

Optional parameter that specifies a place in storage to receive the return code. If specified as a symbol, the symbol must be the label of a word of storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the return code is placed in register 15.

**RSNCODE=symbol | (2 - 12)**

Optional parameter that specifies a place in storage to receive the reason
code. If specified as a symbol, the symbol must be the label of a word of
storage. If specified as a register, the register must contain the address of a
word of storage. If not specified, the reason code is placed in register 0.

VERSION=1.0 | number
Optional parameter that specifies the version number of the parameter list
to be generated by this request.

To use the parameters associated with a version, you must specify the
number of that version or a later version. If you specify an earlier version
level, the parameter is not accepted for processing and an error message is
issued at assembly time. If parameters have a version dependency, the
parameter descriptions with each request type identify the version number
required.

The default version is 1.0.

Return and Reason Codes for the TYPE=DB Query Request

Table 21 contains most of the return and reason codes for TYPE=DB query
requests. The other possible return and reason codes relate to DBRC, not the query
request. For a list of these return and reason codes, see “DBRC Service Aids” in the

<table>
<thead>
<tr>
<th>Table 21. Return and Reason Codes for TYPE=DB Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return Codes</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>X'00000000'</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>Partial Success</td>
</tr>
<tr>
<td>X'00000004'</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>Warning</td>
</tr>
<tr>
<td>X'00000008'</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>X'00000008'</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>Severe Error</td>
</tr>
<tr>
<td>X'0000000C'</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>Storage Error</td>
</tr>
<tr>
<td>X'00000028'</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>X'00000028'</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>X'00000028'</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>X'00000028'</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>X'00000028'</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>X'00000028'</td>
</tr>
</tbody>
</table>
### Table 21. Return and Reason Codes for TYPE=DB Queries (continued)

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000028'</td>
<td>X'D8210003'</td>
<td>Error obtaining storage for RCVINFO block.</td>
</tr>
<tr>
<td>X'00000028'</td>
<td>X'D8210004'</td>
<td>Error obtaining storage for ALLOC block.</td>
</tr>
<tr>
<td>X'00000028'</td>
<td>X'D8210005'</td>
<td>Error obtaining storage for IC block.</td>
</tr>
<tr>
<td>X'00000028'</td>
<td>X'D8210006'</td>
<td>Error obtaining storage for REORG block.</td>
</tr>
<tr>
<td>X'00000028'</td>
<td>X'D8210007'</td>
<td>Error obtaining storage for RECOV block.</td>
</tr>
<tr>
<td>X'00000028'</td>
<td>X'D8220001'</td>
<td>Error obtaining storage for PART block.</td>
</tr>
<tr>
<td>X'00000028'</td>
<td>X'D9100001'</td>
<td>An error occurred processing the request. DBRC will release storage that was obtained up to this point. Another error was encountered, however, during the attempt to release storage.</td>
</tr>
</tbody>
</table>

#### Internal Error

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0000002C'</td>
<td>all</td>
<td>DBRC internal error. Refer to “Database Recovery Control Service Aids” in the <a href="#">IMS V9: Diagnosis Guide and Reference</a> for further information.</td>
</tr>
</tbody>
</table>

#### Parameter Error

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000002'</td>
<td>Invalid TOKEN field address. The address of the field containing the API TOKEN failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000003'</td>
<td>Invalid RETCODE field address. The address of the field to contain the API RETCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000004'</td>
<td>Invalid RSNCODE field address. The address of the field to contain the API RSNCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000005'</td>
<td>Invalid OUTPUT field address. The address of the field to contain the OUTPUT address failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000008'</td>
<td>Invalid DBNAME or DBLIST address. The address of the field containing the DBNAME or DBLIST failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000009'</td>
<td>Invalid DDN address. The address of the field containing the DDN failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D8000001'</td>
<td>Parameter error. Missing or invalid OUTPUT parameter.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D8200001'</td>
<td>Parameter error. LOC parameter is not allowed with DBLIST.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D8200002'</td>
<td>Parameter error. DBNAME parameter is not allowed with LOC=FIRST.</td>
</tr>
</tbody>
</table>
Table 21. Return and Reason Codes for TYPE=DB Queries (continued)

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'D8200003'</td>
<td>Parameter error. DBNAME parameter is required with LOC=NEXT.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D8200004'</td>
<td>Parameter error. DBNAME or DBLIST is required.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D8200005'</td>
<td>Parameter error. Count of databases in DBLIST is zero.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D8200006'</td>
<td>Parameter error. Database information is being requested for a HALDB partition. DBNAME or DBLIST contains a partition name.</td>
</tr>
</tbody>
</table>

Output for TYPE=DB Query Request

The following sections have figures that illustrate the format of output from a QUERY TYPE=DB requests. The sample DSECTs that follow the figures describe in detail the fields of the storage blocks and their relationship to each other.

Full Function Output: [Figure 19](#) shows the fields of the DSPAPQDB block.

![Figure 19. Format of QUERY TYPE=DB (Full Function) Output](#)

The DBDS information is returned only if DDN is specified.

[Figure 20 on page 369](#) describes the fields contained in the DSPAPQDB block shown in [Figure 19](#). [Figure 21 on page 370](#) describes the fields contained in the SS data (DSPAPQSL). Refer to [Figure 29 on page 379](#) for an illustration of the fields of the DBDS output.
<table>
<thead>
<tr>
<th>OFFSET</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>DSPAPQDB</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>STRUCTURE</td>
<td>8</td>
<td>APQDB_DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>8</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQDB_SSLIST</td>
<td>Offset to SS list (DSPAPQSL), zero is no SS auth'd</td>
</tr>
<tr>
<td>12</td>
<td>SIGNED</td>
<td>4</td>
<td>*3(3)</td>
<td>Reserved</td>
</tr>
<tr>
<td>24</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQDB_ICCNT</td>
<td>IC receive needed counter</td>
</tr>
<tr>
<td>26</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQDB_AUFLAG</td>
<td>Authorization flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.1..</td>
<td>APQDB_BKFLG</td>
<td>Backout needed flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.1.</td>
<td>APQDB_PAFLG</td>
<td>Prohibit authorization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...1</td>
<td>APQDB_RDFLG</td>
<td>Read only SS auth</td>
</tr>
<tr>
<td>27</td>
<td>CHARACTER</td>
<td>5</td>
<td>APQDB_IRLMAU</td>
<td>IRLMID of auth SS</td>
</tr>
<tr>
<td>32</td>
<td>SIGNED</td>
<td>2</td>
<td>APQDB_RCVCTR</td>
<td>Recovery needed count</td>
</tr>
<tr>
<td>34</td>
<td>SIGNED</td>
<td>2</td>
<td>APQDB_ICCTR</td>
<td>IC needed count</td>
</tr>
<tr>
<td>36</td>
<td>SIGNED</td>
<td>2</td>
<td>APQDB_ICRECCTR</td>
<td>IC recommended counter</td>
</tr>
<tr>
<td>38</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQDB_SHRLVL</td>
<td>Share level of DB</td>
</tr>
<tr>
<td>39</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQDB_HELDAU</td>
<td>Held auth state</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.1....</td>
<td>APQDB_HAUBIT</td>
<td>High order bit flag</td>
</tr>
<tr>
<td>40</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQDB_DMBNUM</td>
<td>Global DMB number</td>
</tr>
<tr>
<td>42</td>
<td>SIGNED</td>
<td>2</td>
<td>APQDB_SSNUM</td>
<td># of SS auth DB</td>
</tr>
<tr>
<td>44</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQDB_SSENTLEN</td>
<td>Length of each SS entry</td>
</tr>
<tr>
<td>46</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQDB_CACCSS</td>
<td>Access state forchg auth</td>
</tr>
<tr>
<td>47</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQDB_CANCDD</td>
<td>Encode state forchg auth</td>
</tr>
<tr>
<td>48</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQDB_CAHELD</td>
<td>Held state forchg auth</td>
</tr>
<tr>
<td>49</td>
<td>CHARACTER</td>
<td>3</td>
<td>*</td>
<td>Reserved</td>
</tr>
<tr>
<td>52</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQDB_EQECNT</td>
<td>Total EQE count</td>
</tr>
<tr>
<td>54</td>
<td>BIT(16)</td>
<td>2</td>
<td>APQDB_RSRFLG</td>
<td>Flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.1....</td>
<td>APQDB_RCVTRK</td>
<td>Recovery level tracking</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.1.</td>
<td>APQDB_TRKSPN</td>
<td>Tracking is suspended</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...1</td>
<td>APQDB_PUBIT</td>
<td>Suspended by time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>APQDB_ICNDIS</td>
<td>IC needed disabled option</td>
</tr>
<tr>
<td>56</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQDB_GSGNAME</td>
<td>Global Service Group name</td>
</tr>
<tr>
<td>64</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQDB_USID</td>
<td>Last alloc USID</td>
</tr>
<tr>
<td>68</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQDB_AUSID</td>
<td>Last authorized USID</td>
</tr>
<tr>
<td>72</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQDB_RUSID</td>
<td>Last received USID</td>
</tr>
<tr>
<td>76</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQDB_HUSID</td>
<td>Hardened by tracker USID</td>
</tr>
<tr>
<td>80</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQDB_RNUUSID</td>
<td>Receive needed USID</td>
</tr>
<tr>
<td>84</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQDB_RECOVGRP</td>
<td>Recovery Group name</td>
</tr>
<tr>
<td>92</td>
<td>CHARACTER</td>
<td>4</td>
<td>*</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

**CONSTANTS**

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQDB</td>
<td>APQDB_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 20. DSECT of DSPAPQDB*
### QUERY TYPE=DB

**DSPAPQSL**

<table>
<thead>
<tr>
<th>Offset DECIMAL</th>
<th>Offset HEX</th>
<th>Type</th>
<th>Length</th>
<th>Name (Dim)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>STRUCTURE 16</td>
<td>DSPAPQSL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>CHARACTER 16</td>
<td>APQSL_SSINFO</td>
<td>Subsystem list entry</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>CHARACTER 8</td>
<td>APQSL_SSNAME</td>
<td>Subsystem ID</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>UNSIGNED 1</td>
<td>APQSL_ACCESS</td>
<td>Access intent</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>UNSIGNED 1</td>
<td>APQSL_NCDST</td>
<td>Encoded state</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>BIT(8) 1</td>
<td>APQSL_SSFLGS</td>
<td>Flags</td>
<td>0 - Active SS, 1 - Tracking SS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>APQSL_SSROLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0 - Active SS, 1 - Tracking SS</td>
</tr>
<tr>
<td>11</td>
<td>B</td>
<td>BIT(8) 1</td>
<td>*</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>CHARACTER 4</td>
<td>APQSL_BKINFO</td>
<td>Backout information</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>SIGNED 2</td>
<td>APQSL_BKCTR</td>
<td>Backout needed count</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>E</td>
<td>SIGNED 2</td>
<td>APQSL_BKNUM</td>
<td>Backout done count</td>
<td></td>
</tr>
</tbody>
</table>

**CONSTANTS**

<table>
<thead>
<tr>
<th>Len</th>
<th>Type</th>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HEX</td>
<td>01</td>
<td>APQSL_ACCRO</td>
<td>READ-GO</td>
</tr>
<tr>
<td></td>
<td>HEX</td>
<td>02</td>
<td>APQSL_ACCRD</td>
<td>READ</td>
</tr>
<tr>
<td></td>
<td>HEX</td>
<td>03</td>
<td>APQSL_ACCUP</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td>HEX</td>
<td>04</td>
<td>APQSL_ACCEX</td>
<td>EXCLUSIVE</td>
</tr>
</tbody>
</table>

Figure 21. DSECT of DSPAPQSL

**Fast Path DEEB Output:**
The area information is returned only if DDN is specified. Recovery information (RCVINFO) is only returned if the LIST parameter is specified.

Figure 22 on page 372, Figure 24 on page 373, and Figure 25 on page 375 describe the fields contained in the DSPAPQFD and DSPAPQAR blocks shown in Figure 22. Refer to Figure 31 on page 381 for an illustration of the Recovery Information output fields.
QUERY TYPE=DB

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>OFFSET</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECIMAL</td>
<td>HEX TYPE LENGTH NAME (DIM) DESCRIPTION</td>
</tr>
<tr>
<td>========</td>
<td>=========</td>
</tr>
<tr>
<td>0</td>
<td>0 STATES</td>
</tr>
<tr>
<td>0</td>
<td>0 CHARACTER 8</td>
</tr>
<tr>
<td>0</td>
<td>8 SIGNED 4</td>
</tr>
<tr>
<td>24</td>
<td>1B SIGNED 2</td>
</tr>
<tr>
<td>26</td>
<td>1A SIGNED 2</td>
</tr>
<tr>
<td>28</td>
<td>1C SIGNED 2</td>
</tr>
<tr>
<td>30</td>
<td>1E UNSIGNED 2</td>
</tr>
<tr>
<td>32</td>
<td>20 UNSIGNED 2</td>
</tr>
<tr>
<td>34</td>
<td>22 SIGNED 2</td>
</tr>
<tr>
<td>36</td>
<td>24 UNSIGNED 1</td>
</tr>
<tr>
<td>37</td>
<td>25 BIT(8) 1</td>
</tr>
<tr>
<td>1...</td>
<td>1...</td>
</tr>
<tr>
<td>.1...</td>
<td>.1...</td>
</tr>
<tr>
<td>..1.</td>
<td>..1.</td>
</tr>
</tbody>
</table>

CONSTANTS

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER DSPAPQFD</td>
<td>APQFD_EYECATCHER</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 23. DSECT of DSPAPQFD
<table>
<thead>
<tr>
<th>OFFSET</th>
<th>OFFSET</th>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>0</td>
<td>0</td>
<td>STRUCTURE</td>
<td>196</td>
<td>DSPAPQAR</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>8</td>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQAR_DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>16</td>
<td>10</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQAR_EEQELIST</td>
<td>Offset to EEQE list (DSPAPQEL), zero if no EEQEs</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>24</td>
<td>18</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQAR_AREANAME</td>
<td>Area name</td>
</tr>
<tr>
<td>24</td>
<td>(18)</td>
<td>28</td>
<td>22</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQAR_SSLIST</td>
<td>Offset to SS list (DSPAPQSL), zero if no SS auth'd</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>32</td>
<td>20</td>
<td>SIGNED</td>
<td>4</td>
<td>APQAR_HELDAU</td>
<td>High order bit flag</td>
</tr>
<tr>
<td>32</td>
<td>(20)</td>
<td>36</td>
<td>24</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQAR_SSENTLEN</td>
<td>Length of each SS entry</td>
</tr>
<tr>
<td>40</td>
<td>(28)</td>
<td>44</td>
<td>24</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQAR_CACCSS</td>
<td>Access state for CHG AUTH</td>
</tr>
<tr>
<td>41</td>
<td>(29)</td>
<td>46</td>
<td>24</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQAR_CANCDD</td>
<td>Encoded state for CHG AUTH</td>
</tr>
<tr>
<td>42</td>
<td>(2A)</td>
<td>48</td>
<td>24</td>
<td>CHARACTER</td>
<td>5</td>
<td>APQAR_IRLMAU</td>
<td>IRLMID of auth SS</td>
</tr>
<tr>
<td>48</td>
<td>(30)</td>
<td>52</td>
<td>32</td>
<td>BIT(16)</td>
<td>2</td>
<td>APQAR_FLAGS</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>(31)</td>
<td>53</td>
<td>32</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQAR_AUFLAG</td>
<td>Authorization flags</td>
</tr>
<tr>
<td>50</td>
<td>(32)</td>
<td>54</td>
<td>32</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQAR_DSORG</td>
<td>Data set organization</td>
</tr>
</tbody>
</table>

Figure 24. DSECT of DSPAPQAR (Part 1 of 2)
## QUERY TYPE=DB

<table>
<thead>
<tr>
<th>OFFSET DECIMAL</th>
<th>OFFSET HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
<td>35</td>
<td>CHARACTER</td>
<td>1</td>
<td>APQAR_DBORG</td>
<td>IMS DB organization</td>
</tr>
<tr>
<td>54</td>
<td>36</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQAR_GSGNAME</td>
<td>GSG Name</td>
</tr>
<tr>
<td>62</td>
<td>3E</td>
<td>CHARACTER</td>
<td>2</td>
<td>*</td>
<td>Reserved</td>
</tr>
<tr>
<td>64</td>
<td>40</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQAR_USID</td>
<td>Last ALLOC USID</td>
</tr>
<tr>
<td>68</td>
<td>44</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQAR_AUSID</td>
<td>Last authorized USID</td>
</tr>
<tr>
<td>72</td>
<td>48</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQAR_RUSID</td>
<td>Last received USID</td>
</tr>
<tr>
<td>76</td>
<td>4C</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQAR_HUSID</td>
<td>Hardened USID</td>
</tr>
<tr>
<td>80</td>
<td>50</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQAR_RNUSID</td>
<td>Receive needed USID</td>
</tr>
<tr>
<td>84</td>
<td>54</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQAR_RECOVGRP</td>
<td>Recovery Group name</td>
</tr>
<tr>
<td>92</td>
<td>5C</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQAR_CAGRPNAME</td>
<td>Change Accum group name</td>
</tr>
<tr>
<td>100</td>
<td>64</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQAR_GENMX</td>
<td>Max number of ICs that may be predefined for this area</td>
</tr>
<tr>
<td>102</td>
<td>66</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQAR_GENNO</td>
<td>Number of available ICs for this area</td>
</tr>
<tr>
<td>104</td>
<td>68</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQAR_USDIC</td>
<td>Number of ICs used</td>
</tr>
<tr>
<td>106</td>
<td>6A</td>
<td>SIGNED</td>
<td>2</td>
<td>APQAR_EEQECOUNT</td>
<td>EEQE count</td>
</tr>
<tr>
<td>108</td>
<td>6C</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQAR_EEQELENGTH</td>
<td>EEQE entry length</td>
</tr>
<tr>
<td>110</td>
<td>6E</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQAR_NOADS</td>
<td># of ADS in the area</td>
</tr>
<tr>
<td>111</td>
<td>6F</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQAR_AVADS</td>
<td># of available ADS</td>
</tr>
<tr>
<td>112</td>
<td>70</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQAR_ADSLENGTH</td>
<td>ADS entry length</td>
</tr>
<tr>
<td>114</td>
<td>72</td>
<td>CHARACTER</td>
<td>2</td>
<td>*</td>
<td>Reserved</td>
</tr>
<tr>
<td>116</td>
<td>74</td>
<td>CHARACTER</td>
<td>40</td>
<td>APQAR_JCL</td>
<td>GENJCL members</td>
</tr>
<tr>
<td>116</td>
<td>74</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQAR_ICJCL</td>
<td>Image copy member</td>
</tr>
<tr>
<td>124</td>
<td>7C</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQAR_OIJCL</td>
<td>Online IC member</td>
</tr>
<tr>
<td>132</td>
<td>84</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQAR_RCJCL</td>
<td>Recovery member</td>
</tr>
<tr>
<td>140</td>
<td>8C</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQAR_DFJCL</td>
<td>DEFLTJCL member</td>
</tr>
<tr>
<td>148</td>
<td>94</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQAR_RVJCL</td>
<td>Receive JCL member</td>
</tr>
<tr>
<td>156</td>
<td>9C</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQAR_RTPRD</td>
<td>IC retention period</td>
</tr>
<tr>
<td>158</td>
<td>9E</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQAR_DSID</td>
<td>IMS data set ID</td>
</tr>
<tr>
<td>160</td>
<td>A0</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQAR_DSSN</td>
<td>Data set sequence number</td>
</tr>
<tr>
<td>164</td>
<td>A4</td>
<td>CHARACTER</td>
<td>16</td>
<td>APQAR_CFST1</td>
<td>VSO CF Structure 1</td>
</tr>
<tr>
<td>180</td>
<td>B4</td>
<td>CHARACTER</td>
<td>16</td>
<td>APQAR_CFST2</td>
<td>VSO CF Structure 2</td>
</tr>
</tbody>
</table>

### OFFSET DECIMAL | OFFSET HEX | TYPE | LENGTH | NAME (DIM) | DESCRIPTION
--- | --- | --- | --- | --- | ---
0 | 0 | STRUCTURE | 56 | APQAR_ADSLT | Area Data Set List
0 | 0 | CHARACTER | 8 | APQAR_ADSDD | DDNAME of the ADS
8 | 8 | CHARACTER | 44 | APQAR_ADSDN | DSN of the ADS
52 | 34 | BIT(8) | 1 | APQAR_ADSBT | 
... | ... | APQAR_ADSAV | Avail status of ADS
... | ... | APQAR_ADSFM | Format status of create util
... | ... | APQAR_ADSACP | Copy status of create util
53 | 35 | CHARACTER | 3 | * | Reserved

### CONSTANTS

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQAR</td>
<td>APQAR_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 24. DSECT of DSPAPQAR (Part 2 of 2)*
HALDB (Master and All Partitions) Output:

The DBDS information is returned only if DDN is specified.

Figure 25. DSECT of DSPAPQEL

Figure 26. Format of QUERY TYPE=DB (HALDB Master and Partitions) Output
Figure 27, Figure 28 on page 377, and Figure 21 on page 370 describe the fields contained in the DSPAPQHB and DSPAPQHP blocks shown in Figure 26 on page 375. Refer to Figure 29 on page 379 for an illustration of the fields of the DBDS output.

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>STRUCTURE</td>
<td>60</td>
<td>DSPAPQHB</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQHB_DBNAME</td>
<td>HALDB name</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>8</td>
<td>SIGNED</td>
<td>4</td>
<td>*(4) _</td>
<td>Reserved</td>
</tr>
<tr>
<td>24</td>
<td>(18)</td>
<td>18</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQHB_FLAGS</td>
<td>Flags</td>
</tr>
<tr>
<td>25</td>
<td>(19)</td>
<td>19</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQHB_ORG</td>
<td>DB organization</td>
</tr>
<tr>
<td>26</td>
<td>(1A)</td>
<td>1A</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQHB_SHRLVL</td>
<td>Share level</td>
</tr>
<tr>
<td>27</td>
<td>(1B)</td>
<td>1B</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQHB_DSGCNT</td>
<td># DS Group members</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>1C</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQHB_DMBNUM</td>
<td>Global DMB number</td>
</tr>
<tr>
<td>30</td>
<td>(1E)</td>
<td>1E</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQHB_PARTID</td>
<td>Current Partition ID</td>
</tr>
<tr>
<td>32</td>
<td>(20)</td>
<td>20</td>
<td>SIGNED</td>
<td>2</td>
<td>APQHB_PART#</td>
<td>Number of parts in HALDB</td>
</tr>
<tr>
<td>34</td>
<td>(22)</td>
<td>22</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQHB_VERSION#</td>
<td>Version number</td>
</tr>
<tr>
<td>36</td>
<td>(24)</td>
<td>24</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQHB_PSNAME</td>
<td>Name of Part Sel Routine</td>
</tr>
<tr>
<td>44</td>
<td>(2C)</td>
<td>2C</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQHB_GSGNAME</td>
<td>GSG name</td>
</tr>
<tr>
<td>52</td>
<td>(34)</td>
<td>34</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQHB_RECOVGRP</td>
<td>Recovery Group name</td>
</tr>
</tbody>
</table>

CONSTANTS

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQHB</td>
<td>APQHB_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

Figure 27. DSECT of DSPAPQHB
<table>
<thead>
<tr>
<th>OFFSET</th>
<th>OFFSET</th>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>0</td>
<td>0</td>
<td>STRUCTURE</td>
<td>192 DSPAPQHP</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>0</td>
<td>0</td>
<td>CHARACTER</td>
<td>8 APQHP_HALDBNAME</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>0</td>
<td>0</td>
<td>CHARACTER</td>
<td>8 APQHP_PARTNAME</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>0</td>
<td>0</td>
<td>UNSIGNED</td>
<td>4 APQHP_SSLIST</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>0</td>
<td>0</td>
<td>UNSIGNED</td>
<td>4 APQHP_KEYSTRING</td>
</tr>
<tr>
<td>24</td>
<td>(18)</td>
<td>0</td>
<td>0</td>
<td>UNSIGNED</td>
<td>4 APQHP_DSGINFOOFFSET</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>0</td>
<td>0</td>
<td>SIGNED</td>
<td>4 * Reserved</td>
</tr>
<tr>
<td>32</td>
<td>(20)</td>
<td>0</td>
<td>0</td>
<td>CHARACTER</td>
<td>44 APQHP_DSNBASE</td>
</tr>
<tr>
<td>76</td>
<td>(4C)</td>
<td>0</td>
<td>0</td>
<td>CHARACTER</td>
<td>37 APQHP_HDAM</td>
</tr>
<tr>
<td>76</td>
<td>(4C)</td>
<td>0</td>
<td>0</td>
<td>CHARACTER</td>
<td>37 APQHP_RMMNAME</td>
</tr>
<tr>
<td>84</td>
<td>(54)</td>
<td>0</td>
<td>0</td>
<td>SIGNED</td>
<td>4 APQHP_RBN</td>
</tr>
<tr>
<td>88</td>
<td>(58)</td>
<td>0</td>
<td>0</td>
<td>SIGNED</td>
<td>4 APQHP_BYTES</td>
</tr>
<tr>
<td>92</td>
<td>(5C)</td>
<td>0</td>
<td>0</td>
<td>UNSIGNED</td>
<td>2 APQHP_ANCHR</td>
</tr>
<tr>
<td>94</td>
<td>(5E)</td>
<td>0</td>
<td>0</td>
<td>UNSIGNED</td>
<td>1 APQHP_FBFF</td>
</tr>
<tr>
<td>95</td>
<td>(5F)</td>
<td>0</td>
<td>0</td>
<td>UNSIGNED</td>
<td>1 APQHP_FSPF</td>
</tr>
<tr>
<td>96</td>
<td>(60)</td>
<td>0</td>
<td>0</td>
<td>UNSIGNED</td>
<td>2 APQHP_PARTID</td>
</tr>
<tr>
<td>98</td>
<td>(62)</td>
<td>0</td>
<td>0</td>
<td>SIGNED</td>
<td>2 APQHP_PSTLN</td>
</tr>
<tr>
<td>100</td>
<td>(64)</td>
<td>0</td>
<td>0</td>
<td>UNSIGNED</td>
<td>2 APQHP_DSGINFOLEN</td>
</tr>
<tr>
<td>102</td>
<td>(66)</td>
<td>0</td>
<td>0</td>
<td>UNSIGNED</td>
<td>1 APQHP_DSGCNT</td>
</tr>
<tr>
<td>103</td>
<td>(67)</td>
<td>0</td>
<td>0</td>
<td>BIT(8)</td>
<td>1 APQHP_FLAGS</td>
</tr>
<tr>
<td>104</td>
<td>(68)</td>
<td>0</td>
<td>0</td>
<td>CHARACTER</td>
<td>8 APQHP_PREV</td>
</tr>
<tr>
<td>112</td>
<td>(70)</td>
<td>0</td>
<td>0</td>
<td>CHARACTER</td>
<td>8 APQHP_NEXT</td>
</tr>
<tr>
<td>120</td>
<td>(78)</td>
<td>0</td>
<td>0</td>
<td>CHARACTER</td>
<td>8 APQHP_OLRIMS</td>
</tr>
<tr>
<td>128</td>
<td>(80)</td>
<td>0</td>
<td>0</td>
<td>UNSIGNED</td>
<td>2 APQHP_IRCNT</td>
</tr>
<tr>
<td>130</td>
<td>(82)</td>
<td>0</td>
<td>0</td>
<td>BIT(8)</td>
<td>1 APQHP_AUFLAG</td>
</tr>
</tbody>
</table>

*If the Partitioned DB uses high keys, i.e. no Partition Selection routine, the next two fields are used to sort the partitions in key sequence.*

---

Figure 28. DSECT of DSPAPQHP (Part 1 of 2)
## QUERY TYPE=DB

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>STRUCTURE</td>
<td>2</td>
<td>APQHP_DSGINFO</td>
<td>Data set group information</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQHP_BLKSZ</td>
<td>DS block size, OSAM only</td>
</tr>
</tbody>
</table>

## OFFSET DECIMAL HEX TYPE LENGTH NAME (DIM) DESCRIPTION

### CONSTANTS

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQHP</td>
<td>APQHP_EYECATCHER</td>
<td>Figure 28. DSECT of DSPAPQHP (Part 2 of 2)</td>
</tr>
</tbody>
</table>
Recovery information (RCVINFO) is returned only if the LIST parameter is specified.

*Figure 29. Format of QUERY TYPE=DB (DBDS) Output*

Recovery information (RCVINFO) is returned only if the LIST parameter is specified.

*Figure 30 on page 380* and *Figure 25 on page 375* describe the fields contained in the DSPAPQDS block shown in Figure 29. Refer to *Figure 31 on page 381* for an illustration of the Recovery Information output fields.
### QUERY TYPE=DB

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>OFFSET</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>160</td>
<td>DSPAPQDS</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>8</td>
<td>APQDS_DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>8</td>
<td>APQDS_DDNAME</td>
<td>DD name</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>4</td>
<td>APQDS_EEQEIST</td>
<td>Offset to EEQE list (DSPAPQEL), zero if no EEQEs</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>4</td>
<td>*(3)</td>
<td>Reserved</td>
</tr>
<tr>
<td>32</td>
<td>(20)</td>
<td>44</td>
<td>APQDS_DSN</td>
<td>Data set name</td>
</tr>
<tr>
<td>76</td>
<td>(4C)</td>
<td>2</td>
<td>APQDS_RTPRD</td>
<td>IC retention period</td>
</tr>
<tr>
<td>78</td>
<td>(4E)</td>
<td>2</td>
<td>APQDS_DSID</td>
<td>Data set ID number</td>
</tr>
<tr>
<td>80</td>
<td>(50)</td>
<td>4</td>
<td>APQDS_DSN</td>
<td>Recovered-to USID(trkr)</td>
</tr>
<tr>
<td>88</td>
<td>(58)</td>
<td>1</td>
<td>APQDS_FLAGS</td>
<td>BINARY ZEROS</td>
</tr>
<tr>
<td>89</td>
<td>(59)</td>
<td>1</td>
<td>APQDS_DSORG</td>
<td>Data set organization</td>
</tr>
<tr>
<td>90</td>
<td>(5A)</td>
<td>1</td>
<td>APQDS_DBORG</td>
<td>IMS DB organization</td>
</tr>
<tr>
<td>91</td>
<td>(5B)</td>
<td>1</td>
<td>*</td>
<td>Reserved</td>
</tr>
<tr>
<td>92</td>
<td>(5C)</td>
<td>2</td>
<td>APQDS_GENMX</td>
<td>Max number of ICs that may be predefined for this area</td>
</tr>
<tr>
<td>94</td>
<td>(5E)</td>
<td>2</td>
<td>APQDS_AVAILIC#</td>
<td>Number of available ICs for this area</td>
</tr>
<tr>
<td>96</td>
<td>(60)</td>
<td>2</td>
<td>APQDS_USEDIC#</td>
<td>Number of ICs used</td>
</tr>
<tr>
<td>98</td>
<td>(62)</td>
<td>2</td>
<td>APQDS_EEQECOUNT</td>
<td>EEQE count</td>
</tr>
<tr>
<td>100</td>
<td>(64)</td>
<td>2</td>
<td>APQDS_EEQELENGTH</td>
<td>EEQE entry length</td>
</tr>
<tr>
<td>102</td>
<td>(66)</td>
<td>1</td>
<td>APQDS_FLG1</td>
<td>Flags</td>
</tr>
<tr>
<td>103</td>
<td>(67)</td>
<td>1</td>
<td>APQDS_FLG2</td>
<td>DBDS type flags</td>
</tr>
<tr>
<td>104</td>
<td>(68)</td>
<td>8</td>
<td>APQDS_CAGRPNAME</td>
<td>Change Accum group name</td>
</tr>
<tr>
<td>112</td>
<td>(70)</td>
<td>40</td>
<td>APQDS_JCL</td>
<td>GENJCL members</td>
</tr>
<tr>
<td>112</td>
<td>(70)</td>
<td>8</td>
<td>APQDS_ICJCL</td>
<td>Image copy member</td>
</tr>
<tr>
<td>120</td>
<td>(78)</td>
<td>8</td>
<td>APQDS_OIJCL</td>
<td>Online IC member</td>
</tr>
<tr>
<td>128</td>
<td>(80)</td>
<td>8</td>
<td>APQDS_RCJCL</td>
<td>Recovery member</td>
</tr>
<tr>
<td>136</td>
<td>(88)</td>
<td>8</td>
<td>APQDS_DFJCL</td>
<td>DEFLTJCL member</td>
</tr>
<tr>
<td>144</td>
<td>(90)</td>
<td>8</td>
<td>APQDS_RVJCL</td>
<td>Receive JCL member</td>
</tr>
<tr>
<td>152</td>
<td>(98)</td>
<td>8</td>
<td>APQDS_ODDN</td>
<td>OLR partner DBDS</td>
</tr>
</tbody>
</table>

**CONSTANTS**

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHAR</td>
<td>8</td>
<td>APQDS_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 30. DSECT of DSPAPQDS*
Recovery information (RCVINFO) is returned only if the LIST parameter is specified. The pointers will be zero if either the specific information does not exist or it wasn’t requested.

Figure 31. Format of QUERY TYPE=DB (RCVINFO) Output

Figure 32 on page 382, Figure 33 on page 382, Figure 34 on page 383, Figure 35 on page 384, and Figure 36 on page 385 describe the fields contained in the DSPAPQRI block shown in Figure 31.
**QUERY TYPE=DB**

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>OFFSET</th>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>STRUCTURE</td>
<td>32</td>
<td>DSPAPQRI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQRI_DBNAME</td>
<td>Database name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>CHARACTER</td>
<td>8</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQRI_DDNAME</td>
<td>DD name or</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQRI_AREANAME</td>
<td>Area name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>ADDRESS</td>
<td>4</td>
<td>APQRI_ALLOCPT</td>
<td>ptr to ALLOC chain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>ADDRESS</td>
<td>4</td>
<td>APQRI_ICPTR</td>
<td>ptr to IC chain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>(18)</td>
<td>ADDRESS</td>
<td>4</td>
<td>APQRI_RECOVPTR</td>
<td>ptr to RECOV chain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>ADDRESS</td>
<td>4</td>
<td>APQRI_REORGPTR</td>
<td>ptr to REORG chain</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CONSTANTS**

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQRI</td>
<td>APQRI_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 32. DSECT of DSPAPQRI*

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>OFFSET</th>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>STRUCTURE</td>
<td>88</td>
<td>DSPAPQL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQL_DBNAME</td>
<td>Database name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>CHARACTER</td>
<td>8</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQL_DDNAME</td>
<td>DD name or</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQL_AREANAME</td>
<td>Area name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQL_ALLOCTM</td>
<td>Allocation time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQL_DALTM</td>
<td>Deallocation time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>(28)</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQL_STRTM</td>
<td>Log start time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>(34)</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQL_DSSN</td>
<td>Field for DSSN value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>(38)</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQL_USID</td>
<td>Update set identifier</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>(3C)</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQL_ALRID</td>
<td>LRID of begin-upd rec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>(44)</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQL_DLRLID</td>
<td>LRID of end-upd rec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>(4C)</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQL_SLRLID</td>
<td>Last LRID applied if suspended</td>
<td></td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>(54)</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQL_FLAGS</td>
<td>Flags</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1... ....</td>
<td>APQL_TSUSP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.1... ....</td>
<td>APQL_NAPPL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...1. ....</td>
<td>APQL_CICPT</td>
</tr>
<tr>
<td>85</td>
<td>(55)</td>
<td>CHARACTER</td>
<td>3</td>
<td>*</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CONSTANTS**

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQL</td>
<td>APQL_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 33. DSECT of DSPAPQL*
### DSPAPQIC (Part 1 of 2)

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>0</td>
<td>STRUCTURE</td>
<td>64</td>
<td>DSPAPQIC</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>0</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQIC_DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQIC_DDNAME</td>
<td>DD name or area name</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQIC_AREANAME</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>12</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQIC_STARTIME</td>
<td>IC start time, packed decimal</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>12</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQIC_STOPTIME</td>
<td>IC stop time, packed decimal</td>
</tr>
<tr>
<td>40</td>
<td>(28)</td>
<td>1</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQIC_TYPE</td>
<td>IMAGE COPY TYPE</td>
</tr>
<tr>
<td>41</td>
<td>(29)</td>
<td>1</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQIC_STATUS</td>
<td>IC status flags</td>
</tr>
<tr>
<td>42</td>
<td>(2A)</td>
<td>1</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQIC_FLAGS</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>(2B)</td>
<td>1</td>
<td>CHARACTER</td>
<td>1</td>
<td>*</td>
<td>Reserved</td>
</tr>
<tr>
<td>44</td>
<td>(2C)</td>
<td>2</td>
<td>CHARACTER</td>
<td>2</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>(2C)</td>
<td>2</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQIC_OFF1</td>
<td>Offset to image 1 data</td>
</tr>
<tr>
<td>44</td>
<td>(2C)</td>
<td>2</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQIC_OFFU</td>
<td>Offset to user IC data</td>
</tr>
<tr>
<td>46</td>
<td>(2E)</td>
<td>2</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQIC_OFF2</td>
<td>Offset to image 2 data</td>
</tr>
<tr>
<td>48</td>
<td>(30)</td>
<td>4</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQIC_CNT12</td>
<td>Record count</td>
</tr>
<tr>
<td>52</td>
<td>(34)</td>
<td>4</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQIC_USID</td>
<td>Update set ID</td>
</tr>
<tr>
<td>56</td>
<td>(38)</td>
<td>2</td>
<td>CHARACTER</td>
<td>2</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>(38)</td>
<td>2</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQIC_LEN12</td>
<td>Length of image 1/2 data</td>
</tr>
<tr>
<td>58</td>
<td>(3A)</td>
<td>6</td>
<td>CHARACTER</td>
<td>6</td>
<td>*</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

**Figure 34. DSECT of DSPAPQIC (Part 1 of 2)**

**Chapter 17. DBRC Query Request** 383
<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>STRUCTURE 64</td>
<td>Data for image 1 or 2</td>
</tr>
<tr>
<td>0</td>
<td>CHARACTER 44</td>
<td>Data set name</td>
</tr>
<tr>
<td>44</td>
<td>UNSIGNED 2</td>
<td>File sequence number</td>
</tr>
<tr>
<td>46</td>
<td>CHARACTER 8</td>
<td>Unit device type</td>
</tr>
<tr>
<td>54</td>
<td>UNSIGNED 2</td>
<td># of volumes predefined</td>
</tr>
<tr>
<td>56</td>
<td>UNSIGNED 2</td>
<td># of volumes used</td>
</tr>
<tr>
<td>58</td>
<td>UNSIGNED 2</td>
<td>Length of each volume list entry in apqic_VOLS</td>
</tr>
<tr>
<td>60</td>
<td>UNSIGNED 4</td>
<td>Offset to volume list</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>STRUCTURE 80</td>
<td>Data for user IC</td>
</tr>
<tr>
<td>0</td>
<td>CHARACTER 80</td>
<td>User supplied data</td>
</tr>
</tbody>
</table>

**CONSTANTS**

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DECIMAL</td>
<td>255</td>
<td>APQIC_MAXV</td>
<td>max # volumes</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQIC</td>
<td>APQIC_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 34. DSECT of DSPAPQIC (Part 2 of 2)*

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>STRUCTURE 49</td>
<td>Database name</td>
</tr>
<tr>
<td>0</td>
<td>CHARACTER 8</td>
<td>*</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER 8</td>
<td>DD name or</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER 8</td>
<td>AREANAME</td>
</tr>
<tr>
<td>16</td>
<td>CHARACTER 12</td>
<td>APQRV_RUNTIME</td>
</tr>
<tr>
<td>28</td>
<td>CHARACTER 12</td>
<td>APQRV_ENDTIME</td>
</tr>
<tr>
<td>40</td>
<td>UNSIGNED 4</td>
<td>APQRV_FUSID</td>
</tr>
<tr>
<td>44</td>
<td>UNSIGNED 4</td>
<td>APQRV_LUSID</td>
</tr>
<tr>
<td>48</td>
<td>BIT(8) 1</td>
<td>APQRV_FLAGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>APQRV_PITR</td>
</tr>
</tbody>
</table>

**CONSTANTS**

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQRV</td>
<td>APQRV_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 35. DSECT of DSPAPQRV*
This output block is returned when some of the databases specified in the DBLIST block could not be found in the RECON. One block is returned for each database that could not be found. The database name is included in the data area of this block.

When a database is not found, the macro call receives a return code of four (RC=4). If none of the databases in the list are found (RC=8), no output blocks are returned.

Figure 38 on page 386 describes the fields contained in the DSPAPQNF block shown in Figure 37.
QUERY TYPE=xxxxGROUP

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td></td>
<td>STRUCTURE</td>
<td>8</td>
<td>DSPAPQNF</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td></td>
<td>CHARACTER</td>
<td>8</td>
<td>APQNF_DBNAME</td>
<td>DB name</td>
</tr>
</tbody>
</table>

Figure 38. DSECT of DSPAPQNF

Group Query

The Group query (DSPAPI FUNC=QUERY TYPE=xxxxGROUP) request is used to retrieve group and member information for one of the following types of groups registered in RECON:

- DBDS group (TYPE=DBDSGROUP)
- DB group (TYPE=DBGROUP)
- Recovery group (TYPE=RECOVGROUP)
- CA group (TYPE=CAGROUP)
- Global service group (TYPE=GSGROUP)

Syntax for the TYPE=xxxxGROUP Query Request

Parameters for the TYPE=xxxxGROUP Query Request

The following are the possible parameters for QUERY TYPE=xxxxGROUP.

name  Optional symbol you can specify. If used, begins in column 1.

TYPE=DBDSGROUP | DBGROUP | CAGROUP | RECOVGROUP | GSGROUP

Specifies the type of group for which information is requested.

CAINFO= YES | NO

Optional parameter that specifies whether CA execution information is to be included with the CAGROUP information. CAINFO is only valid with
The required parameter level, number to be.

If CAINFO=NO is specified or if CAINFO=YES is specified and no CA
execution information exists, the block dependent pointer (apqhd_depptr)
in the header of the CAGROUP block will be zero.

**TOKEN=***symbol* | (2 - 12)
Required parameter that specifies the address of a 4-byte field that was
returned on the FUNC=STARTDBRC request.

**NAME=*symbol* | (2 - 12)
Optional parameter that specifies the name of the group being queried.
The default (NAME=* ) requests information about all groups of the
specified type.

**MF=S | L | M | E**
Optional parameter that specifies the macro form of the request. See
“Macro Forms of the DSPAPI Macro” on page 352 for the details of the
parameter values.

**OUTPUT=***symbol* | (2 - 12)
Required parameter that specifies a field to receive a pointer to the first
block of a possible chain of group information blocks. See “Output for the
**TYPE=xxxxGROUP Query Request” on page 389 for a detailed description
of the information blocks returned.

The output address is zero if no output was built. This can occur if nothing
in the RECON satisfies the request or if an error occurs before any output
could be built.

The storage for the output blocks is not preallocated. DBRC acquires
private storage for these blocks. The requesting application is responsible
for freeing this storage by using the Buffer Release request (DSPAPI
FUNC=RELBUF), specifying the returned output address.

**RETCODE=***symbol* | (2 - 12)
Optional parameter that specifies a place in storage to receive the return
code. If specified as a symbol, the symbol must be the label of a word of
storage. If specified as a register, the register must contain the address of a
word of storage. If not specified, the return code is placed in register 15.

**RSNRCODE=***symbol* | (2 - 12)
Optional parameter that specifies a place in storage to receive the reason
code. If specified as a symbol, the symbol must be the label of a word of
storage. If specified as a register, the register must contain the address of a
word of storage. If not specified, the reason code is placed in register 0.

**VERSION=1.0 | number
Optional parameter that specifies the version number of the parameter list
to be generated by this request.

To use the parameters associated with a version, you must specify the
number of that version or a later version. If you specify an earlier version
level, the parameter is not accepted for processing and an error message is
issued at assembly time. If parameters have a version dependency, the
parameter descriptions with each request type identify the version number
required.

The default version is 1.0.
**Return and Reason Codes for the TYPE=xxxxGROUP Query Request**

Table 22 contains most of the return and reason codes for TYPE=xxxxGROUP query requests. The other possible return and reason codes relate to DBRC, not the query request. For a list of these return and reason codes, see “DBRC Service Aids” in the [IMS V9: Diagnosis Guide and Reference](#).

Table 22. Return and Reason Codes for TYPE=xxxxGROUP Query Requests

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000000'</td>
<td>X'00000000'</td>
<td>Request completed successfully.</td>
</tr>
<tr>
<td>X'00000008'</td>
<td>X'D8300001'</td>
<td>No group records of the requested type exist in the RECON.</td>
</tr>
<tr>
<td>X'0000000C'</td>
<td>X'C9000001'</td>
<td>Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.</td>
</tr>
</tbody>
</table>

**Storage Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000028'</td>
<td>X'D8300001'</td>
<td>Error obtaining storage for CAGROUP block.</td>
</tr>
<tr>
<td>X'00000028'</td>
<td>X'D8300002'</td>
<td>Error obtaining storage for GSG block.</td>
</tr>
<tr>
<td>X'00000028'</td>
<td>X'D8300003'</td>
<td>Error obtaining storage for DBDGRP, DBGRP, or RECOVGRP block.</td>
</tr>
<tr>
<td>X'00000028'</td>
<td>X'D8310001'</td>
<td>Error obtaining storage for CA block.</td>
</tr>
<tr>
<td>X'00000028'</td>
<td>X'D9100001'</td>
<td>An error occurred processing the request. DBRC will release storage that was obtained up to this point. However, another error was encountered during the attempt to release storage.</td>
</tr>
</tbody>
</table>

**Internal Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0000002C'</td>
<td>all</td>
<td>DBRC internal error. Refer to “Database Recovery Control Service Aids” in the <a href="#">IMS V9: Diagnosis Guide and Reference</a> for further information.</td>
</tr>
</tbody>
</table>

**Parameter Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000002'</td>
<td>Invalid TOKEN field address. The address of the field containing the API TOKEN failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000003'</td>
<td>Invalid RETCODE field address. The address of the field to contain the API RETCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000004'</td>
<td>Invalid RSNCODE field address. The address of the field to contain the API RSNCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000005'</td>
<td>Invalid OUTPUT field address. The address of the field to contain the OUTPUT address failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
</tbody>
</table>
Table 22. Return and Reason Codes for TYPE=xxxxGROUP Query Requests (continued)

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X’00000030’</td>
<td>X’C9000008’</td>
<td>Invalid NAME field address. The address of the field containing the NAME failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X’00000030’</td>
<td>X’D8000001’</td>
<td>Parameter error. Missing or invalid OUTPUT parameter.</td>
</tr>
<tr>
<td>X’00000030’</td>
<td>X’D8300001’</td>
<td>Parameter error. CINFO=YES is only allowed with TYPE=CAGROUP.</td>
</tr>
</tbody>
</table>

Output for the TYPE=xxxxGROUP Query Request

The next few sections illustrate the format of the output from a QUERY TYPE=xxxxGROUP requests. Following the figures that graphically describe the layout of the output are sample DSECTs that describe in detail the fields of the storage blocks and their relationship to each other.

Output for QUERY TYPE=DBDGROUP, DBGROUP, and RECOVGROUP:

```
<table>
<thead>
<tr>
<th>DSPAPQDG</th>
<th>length</th>
<th>DBDGRP, DBGROUP, or RECOVGRP</th>
<th>dep ptr</th>
<th>next ptr</th>
<th>block offset</th>
<th>version</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Group data (DSPAPQDG)  Member info (APQDG_MEMBER)
```

Figure 39. Format for QUERY TYPE=DBDGROUP, DBGROUP, RECOVGROUP Output
### QUERY TYPE=xxxGROUP

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>OFFSET</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td></td>
<td>STRUCTURE</td>
<td>32</td>
<td>DSPAPQDG</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td></td>
<td>CHARACTER</td>
<td>8</td>
<td>APQDG_GROUPNAME</td>
<td>Group name</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td></td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQDG_MEMBERINFO</td>
<td>Offset to group member list</td>
</tr>
<tr>
<td>12</td>
<td>(C)</td>
<td></td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQDG_MEMBERLEN</td>
<td>Length of group member entry</td>
</tr>
<tr>
<td>14</td>
<td>(E)</td>
<td></td>
<td>SIGNED</td>
<td>2</td>
<td>APQDG_MEMBERCOUNT</td>
<td>Number of group members</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td></td>
<td>SIGNED</td>
<td>4</td>
<td>*(3)</td>
<td>Reserved</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td></td>
<td>CHARACTER</td>
<td>4</td>
<td>*</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>OFFSET</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td></td>
<td>STRUCTURE</td>
<td>16</td>
<td>APQDG_MEMBER</td>
<td>List of group members</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td></td>
<td>CHARACTER</td>
<td>16</td>
<td>*</td>
<td>DBDS group</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td></td>
<td>CHARACTER</td>
<td>16</td>
<td>APQDG_DBDSG</td>
<td>DBDS group</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td></td>
<td>CHARACTER</td>
<td>8</td>
<td>APQDG_DBDSG_DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td></td>
<td>CHARACTER</td>
<td>8</td>
<td>*</td>
<td>DBDS group</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td></td>
<td>CHARACTER</td>
<td>8</td>
<td>APQDG_DBDSG_DDNAME</td>
<td>DD name or</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td></td>
<td>CHARACTER</td>
<td>8</td>
<td>APQDG_DBDSG_AREANAME</td>
<td>AREA name</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td></td>
<td>CHARACTER</td>
<td>16</td>
<td>APQDG_DBG</td>
<td>DB group</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td></td>
<td>CHARACTER</td>
<td>8</td>
<td>*</td>
<td>DB group</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td></td>
<td>CHARACTER</td>
<td>8</td>
<td>APQDG_DBG_DBNAME</td>
<td>Database name or</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td></td>
<td>CHARACTER</td>
<td>8</td>
<td>APQDG_DBG_AREANAME</td>
<td>AREA name</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td></td>
<td>CHARACTER</td>
<td>8</td>
<td>*</td>
<td>Not used</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td></td>
<td>CHARACTER</td>
<td>16</td>
<td>APQDG_RECOVG</td>
<td>Recovery group</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td></td>
<td>CHARACTER</td>
<td>8</td>
<td>APQDG_RECOVG_DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td></td>
<td>CHARACTER</td>
<td>8</td>
<td>APQDG_RECOVG_AREANAME</td>
<td>AREA name, null if not Fast Path</td>
</tr>
</tbody>
</table>

### CONSTANTS

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQDG</td>
<td>APQDG_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

Figure 40. **DSECT of DSPAPQDG**

Output for QUERY TYPE=CAGROUP:
The CA block is only returned when CAINFO=YES is specified and records of a change accumulation exist in the RECON.

Figure 41. Format for QUERY TYPE=CAGROUP Output
<table>
<thead>
<tr>
<th>OFFSET</th>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>STRUCTURE</td>
<td>48</td>
<td>DSPAPQCG</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQCG_GROUPNAME</td>
<td>Group name</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQCG_MEMBERINFO</td>
<td>Offset to group member list</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>C</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQCG_MEMBERLEN</td>
<td>Length of group member entry</td>
</tr>
<tr>
<td>14</td>
<td>E</td>
<td>E</td>
<td>SIGNED</td>
<td>2</td>
<td>APQCG_MEMBECOUNT</td>
<td>Number of group members</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>10</td>
<td>SIGNED</td>
<td>4</td>
<td>*(2)</td>
<td>Reserved</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>18</td>
<td>SIGNED</td>
<td>2</td>
<td>APQCG_GRPMAX</td>
<td>Maximum number of CAs that may be predefined for this CA group</td>
</tr>
<tr>
<td>26</td>
<td>1A</td>
<td>1A</td>
<td>SIGNED</td>
<td>2</td>
<td>APQCG_AVAILCA#</td>
<td>Number of available CA data sets for this group</td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
<td>1C</td>
<td>SIGNED</td>
<td>2</td>
<td>APQCG_USEDCA#</td>
<td>Number of used CA data sets</td>
</tr>
<tr>
<td>30</td>
<td>1E</td>
<td>1E</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQCG_FLAGS</td>
<td>Flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>APQCG_REUSE</td>
<td>Reuse CA data sets</td>
</tr>
<tr>
<td>31</td>
<td>1F</td>
<td>1F</td>
<td>CHARACTER</td>
<td>1</td>
<td>*</td>
<td>Reserved</td>
</tr>
<tr>
<td>32</td>
<td>20</td>
<td>20</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQCG_CAJCL</td>
<td>GENJCL CAJCL member name</td>
</tr>
<tr>
<td>40</td>
<td>28</td>
<td>28</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQCG_DFJCL</td>
<td>DEFLTJCL member name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>STRUCTURE</td>
<td>16</td>
<td>APQCG_MEMBER</td>
<td>List of group members</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQCG_DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQCG_DDNAME</td>
<td>DD name or</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQCG_AREANAME</td>
<td>AREA name</td>
</tr>
</tbody>
</table>

**CONSTANTS**

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQCG</td>
<td>APQCG_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 42. DSECT of DSPAPQCG*
<table>
<thead>
<tr>
<th>OFFSET</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>STRUCTURE</td>
<td>96</td>
<td>DSPAPQCA</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQCA_GROUPNAME</td>
<td>Group name</td>
</tr>
<tr>
<td>8</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQCA_MEMBERINFO</td>
<td>Offset to group member list</td>
</tr>
<tr>
<td>12</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQCA_MEMBERLEN</td>
<td>Length of each member entry</td>
</tr>
<tr>
<td>14</td>
<td>SIGNED</td>
<td>2</td>
<td>APQCA_MEMBERCOUNT</td>
<td>Number of group members</td>
</tr>
<tr>
<td>16</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQCA_VOLINFO</td>
<td>Offset to volume information</td>
</tr>
<tr>
<td>20</td>
<td>SIGNED</td>
<td>4</td>
<td>*</td>
<td>Reserved</td>
</tr>
<tr>
<td>24</td>
<td>CHARACTER</td>
<td>44</td>
<td>APQCA_DSN</td>
<td>Data set name</td>
</tr>
<tr>
<td>68</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQCA_STOPTIME</td>
<td>Packed decimal date/time - for predefined datasets, represents record creation time. Otherwise, it is the stoptime of the last logtape volume used as input to the Change Accumulation utility that produced this CA as output. If the CA run included an 'incomplete log subset' it is the start time of the first truncated log volume.</td>
</tr>
<tr>
<td>80</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQCA_RUNTIME</td>
<td>CA run time</td>
</tr>
<tr>
<td>92</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQCA_FLAGS</td>
<td>Flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1... .....</td>
<td>APQCA_ERROR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>.1... .....</td>
<td>APQCA_SUBSET</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>..1... .....</td>
<td>APQCA_COMMAND</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SUBSET/COMP has been set or reset with an external cmd</td>
</tr>
<tr>
<td>93</td>
<td>CHARACTER</td>
<td>3</td>
<td>*</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>STRUCTURE</td>
<td>56</td>
<td>APQCA_MEMBER</td>
<td>List of group members</td>
</tr>
<tr>
<td>0</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQCA_MEM_DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQCA_MEM_DDNAME</td>
<td>DD name or</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQCA_MEM_AREANAME</td>
<td>AREA name</td>
</tr>
<tr>
<td>16</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQCA_MEM_DSSN</td>
<td>Data Set sequence number</td>
</tr>
<tr>
<td>20</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQCA_MEM_USID</td>
<td>USID of last change accumulated</td>
</tr>
<tr>
<td>24</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQCA_MEM_LRID</td>
<td>LRID of last change accumulated</td>
</tr>
<tr>
<td>32</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQCA_MEM_PURGETIME</td>
<td>Purge time</td>
</tr>
<tr>
<td>44</td>
<td>CHARACTER</td>
<td>6</td>
<td>APQCA_MEM_LSN</td>
<td>Lock sequence number</td>
</tr>
<tr>
<td>50</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQCA_MEM_FLAGS</td>
<td>Member flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1... .....</td>
<td>APQCA_MEM_NOCHG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>.1... .....</td>
<td>APQCA_MEM_INDOUBT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>..1... .....</td>
<td>APQCA_MEM_INCOMP</td>
</tr>
<tr>
<td>51</td>
<td>CHARACTER</td>
<td>5</td>
<td>*</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

*Figure 43. DSECT of DSPAPQCA (Part 1 of 2)*
QUERY TYPE=GSGROUP:

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>OFFSET</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>STRUCTURE</td>
<td>24</td>
<td>APQCA_VOL_DATA</td>
<td>Volume data</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQCA_VOL_FILESEQ</td>
<td>File sequence number</td>
</tr>
<tr>
<td>2</td>
<td>(2)</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQCA_VOL_UNIT</td>
<td>Unit type</td>
</tr>
<tr>
<td>10</td>
<td>(A)</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQCA_VOL_COUNT</td>
<td>Number of VOLSERs defined</td>
</tr>
<tr>
<td>12</td>
<td>(C)</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQCA_VOL_USED</td>
<td>Number of VOLSERs used</td>
</tr>
<tr>
<td>14</td>
<td>(E)</td>
<td>SIGNED</td>
<td>2</td>
<td>APQCA_VOL_LENGTH</td>
<td>Length of each VOLSER</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQCA_VOL_OFFSET</td>
<td>Offset to VOLSER information</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>CHARACTER</td>
<td>4</td>
<td>*</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>OFFSET</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>STRUCTURE</td>
<td>6</td>
<td>APQCA_VOLSER</td>
<td>List of VOLSERs</td>
</tr>
</tbody>
</table>

CONSTANTS

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQCA</td>
<td>APQCA_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

Figure 43. DSECT of DSPAPQCA (Part 2 of 2)

Output for QUERY TYPE=GSGROUP:

Figure 44. Format for QUERY TYPE=GSGROUP Output
The Log query (DSPAPI FUNC=QUERY TYPE=LOG) request is used to retrieve log information from RECON for a specific instance of a subsystem.

If the request is for non-RSR logs or RSR active subsystem logs, information from the following RECON records is returned:

- PRILOG
- LOGALL
- SECLOG (if applicable)
- PRISLDS (if applicable)
- SECSLDS (if applicable)
If the request is for RSR tracking subsystem logs, information from the following RECON records is returned:

- PRITSLDS
- SECTSLDS (if applicable)

**Syntax for the TYPE=LOG Query Request**

```
nameDSPAPI—FUNC=QUERY—TYPE=LOG—TOKEN=address—STARTIME=time
```

```
|name|DSPAPI|FUNC=QUERY|TYPE=LOG|TOKEN=address|STARTIME=time
```

```
LOC=SPEC
```

```
LOC=PREV
```

```
LOC=NEXT
```

```
SSID=subsystem_ID
```

```
TRACKER=NO
```

```
TRACKER=YES
```

```
SSID=subsystem_ID
```

```
OUTPUT=output
```

```
RETCODE=returncode
```

```
RSNCODE=reasoncode
```

```
VERSION=1.0
```

```
VERSION=number
```

```
MF=S
```

```
MF=(S,list)
```

```
MF=(L,list)
```

```
MF=(M,list)
```

```
MF=(E,list)
```

```
COMPLETE
```

```
NOCHECK
```

```
LOCATION
```

```
PARAMETERS
```

Parameters for the TYPE=LOG Query Request

The following are the possible parameters for QUERY TYPE=LOG.

**name**  
Optional symbol you can specify. If used, begins in column 1.

**TYPE=LOG**  
Specifies that log information is requested.

**TOKEN=symbol | (2 - 12)**  
Required parameter that specifies the address of a 4-byte field that was returned on the FUNC=STARTDBRC request.

**STARTIME=symbol | (2 - 12)**  
Required parameter that specifies the time-stamp field containing the start time of the log requested. The time is a packed decimal time stamp in UTC format.

**LOC=PREV | NEXT | SPEC**  
Optional parameter that specifies that the request is for the log with a specified start time (LOC=SPEC), a start time preceding the specified start time (LOC=PREV), or a start time following the specified start time (LOC=NEXT). The STARTIME parameter is used as the base of the search and does not need to be the start time of a log in RECON.

**SSID=symbol | (2 - 12)**  
Optional parameter that specifies the subsystem name for the log being queried. The SSID parameter is required for RSR tracking subsystem log
queries (TRACKER=YES). For other queries, you can only specify the SSID
parameter when you request the previous or next log of a specific
subsystem, that is, you specify LOC=PREV or LOC=NEXT.

**TRACKER=YES | NO**
Optional parameter that specifies whether log information is being
requested for an RSR tracking subsystem or not. Specify TRACKER=YES to
request the RSR tracker’s log information. Specify TRACKER=NO to
request log information from a non-RSR subsystem or an active subsystem
of an RSR system. TRACKER=NO is the default.

**MF=S | L | M | E**
Optional parameter that specifies the macro form of the request. See
“Macro Forms of the DSPAPI Macro” on page 352 for the details of the
parameter values.

**OUTPUT=symbol | (2 - 12)**
Required parameter that specifies a field to receive a pointer to the first
block of a possible chain of log information blocks. See “Output for the
TYPE=LOG Query Request” on page 399 for a detailed description of the
information blocks returned.

The output address is zero if no output was built. This can occur if nothing
in the RECON satisfies the request or if an error occurs before any output
could be built.

The storage for the output blocks is not preallocated. DBRC acquires
private storage for these blocks. The requesting application is responsible
for freeing this storage by using the Buffer Release request (DSPAPI
FUNC=RELMBUF), specifying the returned output address.

**RETCODE=symbol | (2 - 12)**
Optional parameter that specifies a place in storage to receive the return
code. If specified as a symbol, the symbol must be the label of a word of
storage. If specified as a register, the register must contain the address of a
word of storage. If not specified, the return code is placed in register 15.

**RSNCODE=symbol | (2 - 12)**
Optional parameter that specifies a place in storage to receive the reason
code. If specified as a symbol, the symbol must be the label of a word of
storage. If specified as a register, the register must contain the address of a
word of storage. If not specified, the reason code is placed in register 0.

**VERSION=1.0 | number**
Optional parameter that specifies the version number of the parameter list
to be generated by this request.

To use the parameters associated with a version, you must specify the
number of that version or a later version. If you specify an earlier version
level, the parameter is not accepted for processing and an error message is
issued at assembly time. If parameters have a version dependency, the
parameter descriptions with each request type identify the version number
required.

The default version is 1.0.

**SSID=symbol | (2-12)**
Optional parameter that specifies the subsystem name for the log being
queried. The SSID parameter is required for RSR tracking subsystem log
queries (TRACKER=YES). Otherwise, you can only specify the SSID
Return and Reason Codes for the TYPE=LOG Query Request

Table 23 contains most of the return and reason codes for TYPE=BACKOUT query requests. The other possible return and reason codes relate to DBRC, not the query request. For a list of these return and reason codes, see “DBRC Service Aids” in the IMS V9: Diagnosis Guide and Reference.

Table 23. Return and Reason Codes for TYPE=LOG Query Requests

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000000'</td>
<td>X'00000000'</td>
<td>Request completed successfully.</td>
</tr>
</tbody>
</table>

**Warning**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000008'</td>
<td>X'D8400001'</td>
<td>No previous or next log record of the requested log type - PRILOG or PRITSLDS.</td>
</tr>
<tr>
<td>X'00000008'</td>
<td>X'D8400002'</td>
<td>The specified log record of the requested log type - PRILOG or PRITSLDS - does not exist.</td>
</tr>
</tbody>
</table>

**Severe Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0000000C'</td>
<td>X'C9000001'</td>
<td>Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.</td>
</tr>
</tbody>
</table>

**Storage Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000028'</td>
<td>X'D8400001'</td>
<td>Error obtaining storage for LOGINFO block.</td>
</tr>
<tr>
<td>X'00000028'</td>
<td>X'D8400002'</td>
<td>Error obtaining storage for PRILOG, SECLOG, PRISLDS, SECSLDS, PRITSLDS, or SECTSLDS block.</td>
</tr>
<tr>
<td>X'00000028'</td>
<td>X'D8400003'</td>
<td>Error obtaining storage for LOGALL block.</td>
</tr>
<tr>
<td>X'00000028'</td>
<td>X'D9100001'</td>
<td>An error occurred processing the request. DBRC will release storage that was obtained up to this point. However, another error was encountered during the attempt to release storage.</td>
</tr>
</tbody>
</table>

**Internal Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0000002C'</td>
<td>all</td>
<td>DBRC internal error. Refer to “Database Recovery Control Service Aids” in the IMS V9: Diagnosis Guide and Reference for further information.</td>
</tr>
</tbody>
</table>

**Parameter Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Invalid TOKEN field address. The address of the field containing the API TOKEN failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Invalid RETCODE field address. The address of the field to contain the API RETCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Invalid RSNCODE field address. The address of the field to contain the API RSNCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
</tbody>
</table>
Table 23. Return and Reason Codes for TYPE=LOG Query Requests (continued)

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Invalid OUTPUT field address. The address of the field to contain the OUTPUT address failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Invalid SSID field address. The address of the field containing the SSID failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Invalid STARTIME field address. The address of the field containing the STARTIME failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D8000001'</td>
<td>Parameter error. Missing or invalid OUTPUT parameter.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D8400001'</td>
<td>Parameter error. STARTIME parameter is required.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D8400002'</td>
<td>Parameter error. SSID is required with TRACKER=YES.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D8400003'</td>
<td>Parameter error. SSID parameter is not allowed for active logs (TRACKER=NO) unless specified with LOC=PREV or NEXT.</td>
</tr>
</tbody>
</table>

Output for the TYPE=LOG Query Request

The following sections have figures that illustrate the format of output from a QUERY TYPE=LOG requests. The sample DSECTs that follow the figures describe in detail the fields of the storage blocks and their relationship to each other.

Log Information Output:
Figure 46. Format for QUERY TYPE=LOG Log Information Output

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>0</td>
<td>STRUCTURE</td>
<td>48</td>
<td>DSPAPQLI</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>0</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQLI_SSID</td>
<td>Log SSID</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>8</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQLI_STARTTIME</td>
<td>Log start time</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>20</td>
<td>ADDRESS</td>
<td>4</td>
<td>APLI_PRILOGPTR</td>
<td>ptr to PRILOG block</td>
</tr>
<tr>
<td>24</td>
<td>(18)</td>
<td>24</td>
<td>ADDRESS</td>
<td>4</td>
<td>APLI_LOGALLPTR</td>
<td>ptr to LOGALL block</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>28</td>
<td>ADDRESS</td>
<td>4</td>
<td>APLI_SECLOGPTR</td>
<td>ptr to SECLOG block</td>
</tr>
<tr>
<td>32</td>
<td>(20)</td>
<td>32</td>
<td>ADDRESS</td>
<td>4</td>
<td>APLI_PRISLDSPTR</td>
<td>ptr to PRISLDS block</td>
</tr>
<tr>
<td>36</td>
<td>(24)</td>
<td>36</td>
<td>ADDRESS</td>
<td>4</td>
<td>APLI_SECSLDSPTR</td>
<td>ptr to SECSLDS block</td>
</tr>
<tr>
<td>40</td>
<td>(28)</td>
<td>40</td>
<td>ADDRESS</td>
<td>4</td>
<td>APLI_PRITSLDSPTR</td>
<td>ptr to PRITSLDS block</td>
</tr>
<tr>
<td>44</td>
<td>(2C)</td>
<td>44</td>
<td>ADDRESS</td>
<td>4</td>
<td>APLI_SECTSLDSPTR</td>
<td>ptr to SECTSLDS block</td>
</tr>
</tbody>
</table>

CONSTANTS

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQLI</td>
<td>APQLI_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

Figure 47. DSECT of DSPAPQLI

QUERY TYPE=LOG

TYPE=LOG Output for PRILOG, SECLOG, PRISLDS, SECSLDS, PRITSLDS, and SECTSLDS:
**QUERY TYPE=LOG**

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>STRUCTURE</td>
<td>96</td>
<td>DSPAPQLG</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQLG_FIRSTLOGDS</td>
<td>Offset to first log DS entry</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQLG_LASTLOGDS</td>
<td>Offset to last log DS entry</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>4</td>
<td>SIGNED</td>
<td>4</td>
<td>*(2)</td>
<td>Reserved</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQLG_SSID</td>
<td>Subsystem ID</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>12</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQLG_STARTTIME</td>
<td>Log start time</td>
</tr>
<tr>
<td>36</td>
<td>24</td>
<td>12</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQLG_ENDTIME</td>
<td>Log end time</td>
</tr>
<tr>
<td>48</td>
<td>30</td>
<td>4</td>
<td>SIGNED</td>
<td>4</td>
<td>APQLG_DSNCOUNT</td>
<td>Number of data sets</td>
</tr>
<tr>
<td>52</td>
<td>34</td>
<td>1</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQLG_RELVL</td>
<td>Log Release Level</td>
</tr>
<tr>
<td>53</td>
<td>35</td>
<td>1</td>
<td>BIT</td>
<td>8</td>
<td>APQLG_FLAGS1</td>
<td>Flags</td>
</tr>
<tr>
<td></td>
<td>1...</td>
<td></td>
<td></td>
<td>..</td>
<td>APQLG_ONLINE</td>
<td>Online log</td>
</tr>
<tr>
<td></td>
<td>.1...</td>
<td></td>
<td></td>
<td>..</td>
<td>APQLG_LSTAR</td>
<td>Last OLDS has been archived</td>
</tr>
<tr>
<td></td>
<td>...1</td>
<td></td>
<td></td>
<td>.</td>
<td>APQLG_LASTNA</td>
<td>Last OLDS has not been archived</td>
</tr>
<tr>
<td></td>
<td>....</td>
<td></td>
<td></td>
<td>1...</td>
<td>APQLG_GAP</td>
<td>There is a gap in this log</td>
</tr>
<tr>
<td></td>
<td>....</td>
<td></td>
<td></td>
<td></td>
<td>APQLG_PRGAP</td>
<td>There is a gap in a prev log</td>
</tr>
<tr>
<td>54</td>
<td>36</td>
<td>1</td>
<td>BIT</td>
<td>8</td>
<td>APQLG_FLAGS2</td>
<td>Flags</td>
</tr>
<tr>
<td></td>
<td>1...</td>
<td></td>
<td></td>
<td>..</td>
<td>APQLG_TRKNG</td>
<td>Tracking log data set</td>
</tr>
<tr>
<td></td>
<td>.1...</td>
<td></td>
<td></td>
<td>..</td>
<td>APQLG_NTERM</td>
<td>IMS subsystem has terminated normally</td>
</tr>
<tr>
<td></td>
<td>...1</td>
<td></td>
<td></td>
<td>.</td>
<td>APQLG_BKLOG</td>
<td>Batch backout log</td>
</tr>
<tr>
<td>55</td>
<td>37</td>
<td>1</td>
<td>CHARACTER</td>
<td>1</td>
<td>*</td>
<td>Reserved</td>
</tr>
<tr>
<td>56</td>
<td>38</td>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQLG_FIRSTLRID</td>
<td>Id of first log record</td>
</tr>
<tr>
<td>64</td>
<td>40</td>
<td>4</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQLG_PTOKEN</td>
<td>PRILOG token</td>
</tr>
<tr>
<td>68</td>
<td>44</td>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQLG_GSGNAME</td>
<td>GSG name</td>
</tr>
<tr>
<td>76</td>
<td>4C</td>
<td>12</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQLG_CHKPT0</td>
<td>Checkpoint 0 time</td>
</tr>
<tr>
<td>88</td>
<td>58</td>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>*</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

**Figure 48. Format for QUERY TYPE=LOG Output for PRILOG, SECLOG, PRISLDS, SECSLDS, PRITSLDS, and SECTSLDS**

**Figure 49. DSECT of DSPAPQLG (Part 1 of 2)**
### OFFSET OFFSET
<table>
<thead>
<tr>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (0)</td>
<td>120</td>
<td>STRUCTURE</td>
<td>120</td>
<td>APQLG_DS_ENTRY</td>
<td>Data set entry</td>
</tr>
<tr>
<td>0 (0)</td>
<td>4</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQLG_DS_NEXT</td>
<td>Offset to next log DS entry</td>
</tr>
<tr>
<td>4 (4)</td>
<td>4</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQLG_DS_PREV</td>
<td>Offset to prev log DS entry</td>
</tr>
<tr>
<td>8 (8)</td>
<td>4</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQLG_DS_VOLINFO</td>
<td>Offset to DS volume info</td>
</tr>
<tr>
<td>12 (C)</td>
<td>44</td>
<td>CHARACTER</td>
<td>44</td>
<td>APQLG_DS_DSNNAME</td>
<td>Data set name</td>
</tr>
<tr>
<td>56 (38)</td>
<td>12</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQLG_DS_STARTTIME</td>
<td>DS start time</td>
</tr>
<tr>
<td>68 (44)</td>
<td>12</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQLG_DS_ENDTIME</td>
<td>DS end time</td>
</tr>
<tr>
<td>80 (50)</td>
<td>1</td>
<td>BIT(8)</td>
<td>1</td>
<td>1 APQLG_DS_FLAGS1</td>
<td>Flags</td>
</tr>
<tr>
<td>81 (51)</td>
<td>1</td>
<td>BIT(8)</td>
<td>1</td>
<td>1 APQLG_DS_FLAGS2</td>
<td>Flags</td>
</tr>
<tr>
<td>82 (52)</td>
<td>2</td>
<td>CHARACTER</td>
<td>2</td>
<td>APQLG_DS_DFLG3</td>
<td>Reserved</td>
</tr>
<tr>
<td>84 (54)</td>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQLG_DS_FRID</td>
<td>First log record ID</td>
</tr>
<tr>
<td>92 (5C)</td>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQLG_DS_LLFRID</td>
<td>Last log record ID</td>
</tr>
<tr>
<td>100 (64)</td>
<td>4</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQLG_DS_LASTBLKSEQNO</td>
<td>Last block seq number</td>
</tr>
<tr>
<td>104 (68)</td>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQLG_DS_UNITTYPE</td>
<td>Unit type name</td>
</tr>
<tr>
<td>112 (70)</td>
<td>2</td>
<td>UNSIGNED</td>
<td>2</td>
<td>1 APQLG_DS_FILESEQ</td>
<td>File sequence no</td>
</tr>
<tr>
<td>114 (72)</td>
<td>2</td>
<td>UNSIGNED</td>
<td>2</td>
<td>1 APQLG_DS_VOLCOUNT</td>
<td>Number of volumes</td>
</tr>
<tr>
<td>116 (74)</td>
<td>1</td>
<td>UNSIGNED</td>
<td>1</td>
<td>1 APQLG_DS_CHKPTCOUNT</td>
<td>Number of chkpts on DSN</td>
</tr>
<tr>
<td>117 (75)</td>
<td>1</td>
<td>BIT(8)</td>
<td>1</td>
<td>1 APQLG_DS_CHKPTTYPES</td>
<td>CHKPT types.</td>
</tr>
<tr>
<td>118 (76)</td>
<td>2</td>
<td>UNSIGNED</td>
<td>2</td>
<td>* Reserved</td>
<td></td>
</tr>
</tbody>
</table>

### OFFSET OFFSET
<table>
<thead>
<tr>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (0)</td>
<td>48</td>
<td>STRUCTURE</td>
<td>48</td>
<td>APQLG_DSVOLUME</td>
<td>Volume information</td>
</tr>
<tr>
<td>0 (0)</td>
<td>4</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQLG_DSVOL_NEXT</td>
<td>Offset to next volume</td>
</tr>
<tr>
<td>4 (4)</td>
<td>6</td>
<td>CHARACTER</td>
<td>6</td>
<td>APQLG_DSVOL_SER</td>
<td>VOLSER</td>
</tr>
<tr>
<td>10 (A)</td>
<td>1</td>
<td>CHARACTER</td>
<td>1</td>
<td>1 APQLG_DSVOL_CHKPTCT</td>
<td>Volume chkpt count</td>
</tr>
<tr>
<td>11 (B)</td>
<td>1</td>
<td>CHARACTER</td>
<td>1</td>
<td>* Reserved</td>
<td></td>
</tr>
<tr>
<td>12 (C)</td>
<td>12</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQLG_DSVOL_ENDTIME</td>
<td>Volume end time</td>
</tr>
<tr>
<td>24 (18)</td>
<td>12</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQLG_DSVOL_CHKPTID</td>
<td>Checkpoint ID</td>
</tr>
<tr>
<td>36 (24)</td>
<td>6</td>
<td>CHARACTER</td>
<td>6</td>
<td>APQLG_DSVOL_LOCKSN</td>
<td>Lock Sequence Number</td>
</tr>
<tr>
<td>42 (2A)</td>
<td>6</td>
<td>CHARACTER</td>
<td>6</td>
<td>* Reserved</td>
<td></td>
</tr>
</tbody>
</table>

### CONSTANTS

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQLG</td>
<td>APQLG_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

Figure 49. DSECT of DSPAPQLG (Part 2 of 2)

TYPE=LOG Output for LOGALL:
Figure 50. Format for QUERY TYPE=LOG Output for LOGALL

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>STRUCTURE</td>
<td>48</td>
<td>DSPAPQLA</td>
<td>Offset to allocated DBDS/Area list</td>
</tr>
<tr>
<td>0</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQLA_DBSDAREAINFO</td>
<td>Reserved</td>
</tr>
<tr>
<td>4</td>
<td>SIGNED</td>
<td>4</td>
<td>*(3)</td>
<td>Reserved</td>
</tr>
<tr>
<td>16</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQLA_PRILOGTIME</td>
<td>PRILOG time</td>
</tr>
<tr>
<td>28</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQLA_FLAGS</td>
<td>Flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>APQLA_NONREGD</td>
<td>Non-registered DB updated</td>
</tr>
<tr>
<td>29</td>
<td>UNSIGNED</td>
<td>3</td>
<td>APQLA_DBSDAREACOUNT</td>
<td>Number of DBDS/Areas allocated on this log</td>
</tr>
<tr>
<td>32</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQLA_DBSDAREALEN</td>
<td>Length of DBDS/Area entry</td>
</tr>
<tr>
<td>36</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQLA_EARLIESTALLOC</td>
<td>Earliest ALLOC time for this log</td>
</tr>
</tbody>
</table>

Figure 51. DSECT of DSPAPQLA

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>STRUCTURE</td>
<td>32</td>
<td>APQLA_DBSDAREA</td>
<td>List of allocated DBDSs and Areas</td>
</tr>
<tr>
<td>0</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQLA_DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>*(3)</td>
<td>DD name or AREA name</td>
</tr>
<tr>
<td>16</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQLA_FIRSTALLOC</td>
<td>Earliest ALLOC time for this DBDS/Area on this log</td>
</tr>
<tr>
<td>28</td>
<td>SIGNED</td>
<td>2</td>
<td>APQLA_ALLNO</td>
<td>Number of ALLOCs for this DBDS/Area on this log</td>
</tr>
<tr>
<td>30</td>
<td>CHARACTER</td>
<td>2</td>
<td>*(3)</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

CONSTANTS

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQLA</td>
<td>APQLA_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>
OLDS Query

The OLDS query (DSPAPI FUNC=QUERY TYPE=OLDS) request is used to retrieve online log data set information from the RECON for a specific subsystem or all subsystems.

Syntax for the TYPE=OLDS Query Request

```
name
DSPAPI--FUNC=QUERY--TYPE=OLDS--TOKEN=address

SSID=*  MF=S
SSID=subsystem_ID  MF=(S,list)
MF=(L,list)
MF=(M,list,COMPLETE)
MF=(E,list,COMPLETE)

OUTPUT=output
VERSION=number
RETCODE=returncode
RSNCODE=reasoncode
```

Parameters for the TYPE=OLDS Query Request

The following are the possible parameters for QUERY TYPE=OLDS.

- **name**  Optional symbol you can specify. If used, begins in column 1.
- **TYPE=OLDS**
  Specifies that online log data set information is requested.
- **TOKEN=**  |  *(2 - 12)*
  Required parameter that specifies the address of a 4-byte field that was returned on the FUNC=STARTDBRC request.
- **SSID=**  |  *(2 - 12)*
  Specifies the subsystem name for the backout being queried. The default (SSID=*) requests all of the OLDS information.
- **MF=**  |  S | L | M | E
  Optional parameter that specifies the macro form of the request. See "Macro Forms of the DSPAPI Macro" on page 352 for the details of the parameter values.
- **OUTPUT=**  |  *(2 - 12)*
  Required parameter that specifies a field to receive a pointer to the first block of a possible chain of OLDS information blocks. See "Output for the TYPE=OLDS Query Request" on page 406 for a detailed description of the information blocks returned.

The output address is zero if no output was built. This can occur if nothing in the RECON satisfies the request or if an error occurs before any output could be built.

The storage for the output blocks is not preallocated. DBRC acquires private storage for these blocks. The requesting application is responsible
for freeing this storage by using the Buffer Release request (DSPAPI FUNC=RELBUF), specifying the returned output address.

**RET CODE=symbol | (2 - 12)**
Optional parameter that specifies a place in storage to receive the return code. If specified as a symbol, the symbol must be the label of a word of storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the return code is placed in register 15.

**RSN CODE=symbol | (2 - 12)**
Optional parameter that specifies a place in storage to receive the reason code. If specified as a symbol, the symbol must be the label of a word of storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the reason code is placed in register 0.

**VERSION=1.0 | number**
Optional parameter that specifies the version number of the parameter list to be generated by this request.

To use the parameters associated with a version, you must specify the number of that version or a later version. If you specify an earlier version level, the parameter is not accepted for processing and an error message is issued at assembly time. If parameters have a version dependency, the parameter descriptions with each request type identify the version number required.

The default version is 1.0.

### Return and Reason Codes for the TYPE=OLDS Query Request

Table 24 contains most of the return and reason codes for TYPE=OLDS query requests. The other possible return and reason codes relate to DBRC, not the query request. For a list of these return and reason codes, see “DBRC Service Aids” in the

[IMS V9: Diagnosis Guide and Reference](#)

**Table 24. Return and Reason Codes for TYPE=OLDS Query Requests**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X’00000000’</td>
<td>X’00000000’</td>
<td>Request completed successfully.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Warning</strong></td>
</tr>
<tr>
<td>X’00000008’</td>
<td>X’D8500001’</td>
<td>No PRIOLDS records exist.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Severe Error</strong></td>
</tr>
<tr>
<td>X’0000000C’</td>
<td>X’C9000001’</td>
<td>Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNCTION=STARTDBRC call.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Storage Error</strong></td>
</tr>
<tr>
<td>X’00000028’</td>
<td>X’D8500001’</td>
<td>Error obtaining storage for OLDS block.</td>
</tr>
<tr>
<td>X’00000028’</td>
<td>X’D9100001’</td>
<td>An error occurred processing the request. DBRC will release storage that was obtained up to this point. However, another error was encountered during the attempt to release storage.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Internal Error</strong></td>
</tr>
<tr>
<td>X’0000002C’</td>
<td>all</td>
<td>DBRC internal error. Refer to “Database Recovery Control Service Aids” in the <a href="#">IMS V9: Diagnosis Guide and Reference</a> for further information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Parameter Error</strong></td>
</tr>
</tbody>
</table>
Table 24. Return and Reason Codes for TYPE=OLDS Query Requests (continued)

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000002'</td>
<td>Invalid TOKEN field address. The address of the field containing the API TOKEN failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000003'</td>
<td>Invalid RETCODE field address. The address of the field to contain the API RETCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000004'</td>
<td>Invalid RSNCODE field address. The address of the field to contain the API RSNCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000005'</td>
<td>Invalid OUTPUT field address. The address of the field to contain the OUTPUT address failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D8000001'</td>
<td>Parameter error. Missing or invalid OUTPUT parameter.</td>
</tr>
</tbody>
</table>

Output for the TYPE=OLDS Query Request

Figure 52 on page 407 illustrates the format of the output from a QUERY TYPE=BACKOUT request. Figure 53 on page 408 is a sample DSECT describes in detail the fields of the storage blocks and their relationship to each other.
Figure 52. Format for QUERY TYPE=OLDS Output
### QUERY TYPE=OLDS

<table>
<thead>
<tr>
<th>Offset</th>
<th>Decimal</th>
<th>Hex</th>
<th>Type</th>
<th>Length</th>
<th>Name (Dim)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>0</td>
<td>STRUCTURE</td>
<td>48</td>
<td>DSPAPQOL</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(4)</td>
<td>4</td>
<td>SIGNED</td>
<td>4</td>
<td>* (3)</td>
<td>Offset to OLDS list</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>16</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQOL_SSID</td>
<td>Subsystem ID</td>
</tr>
<tr>
<td>24</td>
<td>(18)</td>
<td>24</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQOL_OLDSLEN</td>
<td>Length of OLDS entry</td>
</tr>
<tr>
<td>26</td>
<td>(1A)</td>
<td>26</td>
<td>SIGNED</td>
<td>2</td>
<td>APQOL_OLDSCOUNT</td>
<td>Number of OLDS entries</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>28</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQOL_CHKPT0</td>
<td>Checkpoint 0 time</td>
</tr>
<tr>
<td>40</td>
<td>(28)</td>
<td>40</td>
<td>CHARACTER</td>
<td>8</td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Offset</th>
<th>Decimal</th>
<th>Hex</th>
<th>Type</th>
<th>Length</th>
<th>Name (Dim)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>0</td>
<td>STRUCTURE</td>
<td>128</td>
<td>APQOL_OLDSENTRY</td>
<td>OLDS entry</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>0</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQOL_DDNAME</td>
<td>the OLDS DD name</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>0</td>
<td>CHARACTER</td>
<td>44</td>
<td>APQOL_DSNAM</td>
<td>OLDS data set name</td>
</tr>
<tr>
<td>52</td>
<td>(34)</td>
<td>52</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQOL_OPENTIME</td>
<td>OLDS open time</td>
</tr>
<tr>
<td>64</td>
<td>(40)</td>
<td>64</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQOL_CLOSETIME</td>
<td>Close time</td>
</tr>
<tr>
<td>76</td>
<td>(4C)</td>
<td>76</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQOL_PRILOGTIME</td>
<td>Start time of associated PRILOG</td>
</tr>
<tr>
<td>88</td>
<td>(58)</td>
<td>88</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQOL_FLSN</td>
<td>LSN of first record in OLDS</td>
</tr>
<tr>
<td>96</td>
<td>(60)</td>
<td>96</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQOL_LLSN</td>
<td>LSN of last record in OLDS</td>
</tr>
<tr>
<td>104</td>
<td>(68)</td>
<td>104</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQOL_FLAG1</td>
<td>1... ....</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.1... ....</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>..1 ....</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...1...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.... 1...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.... .1..</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.... ..1.</td>
</tr>
<tr>
<td>105</td>
<td>(69)</td>
<td>105</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQOL_FLAG2</td>
<td>1111 ....</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1... ....</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.1... ....</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>..1 ...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...1 ...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.... 1...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.... .1..</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.... ..1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.... ...1</td>
</tr>
<tr>
<td>106</td>
<td>(6A)</td>
<td>106</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQOL_RELVL</td>
<td>Log release level</td>
</tr>
<tr>
<td>107</td>
<td>(6B)</td>
<td>107</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQOL_GAVER</td>
<td>GENJCL.ARCHIVE version</td>
</tr>
<tr>
<td>108</td>
<td>(6C)</td>
<td>108</td>
<td>BIT(32)</td>
<td>4</td>
<td>APQOL_BLOCKSEQNO</td>
<td>Block sequence number</td>
</tr>
<tr>
<td>112</td>
<td>(70)</td>
<td>112</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQOL_ARJOB</td>
<td>Name of the archive job</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Generated by GENJCL.ARCHIVE</td>
</tr>
<tr>
<td>120</td>
<td>(78)</td>
<td>120</td>
<td>CHARACTER</td>
<td>6</td>
<td>APQOL_LOCKSEQNO</td>
<td>Lock sequence number</td>
</tr>
<tr>
<td>126</td>
<td>(7E)</td>
<td>126</td>
<td>CHARACTER</td>
<td>2</td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>

### CONSTANTS

<table>
<thead>
<tr>
<th>Len</th>
<th>Type</th>
<th>Value</th>
<th>Name (Dim)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQOL</td>
<td>APQOL_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 53. DSECT of DSPAPQOL*
RECON Status Query

The RECON status query (DSPAPI FUNC=QUERY TYPE=RECON) request is used to retrieve information pertaining to the RECON data sets. This includes RECON header information, as well as the status of each RECON data set.

Syntax for the TYPE=RECON Query Request

```
name DSPAPI—FUNC=QUERY—TYPE=RECON—TOKEN=address

MF=S

MF=(S,list)

MF=(L,list)

MF=(M,list)

MF=(E,list)

OUTPUT=output

RETCODE=returncode

VERSION=1.0

VERSION=number

RSNCODE=reasoncode
```

Parameters for the TYPE=RECON Query Request

The following are the possible parameters for QUERY TYPE=RECON.

- **name**  
  Optional symbol you can specify. If used, begins in column 1.

- **TYPE=RECON**
  Specifies that RECON status information is requested.

- **TOKEN=-symbol | (2 - 12)**
  Required parameter that specifies the address of a 4-byte field that was returned on the FUNC=STARTDBRC request.

- **MF=S | L | M | E**
  Optional parameter that specifies the macro form of the request. See “Macro Forms of the DSPAPI Macro” on page 352 for the details of the parameter values.

- **OUTPUT=-symbol | (2 - 12)**
  Required parameter that specifies a field to receive a pointer to the RECON status information block. See “Output for the TYPE=RECON Query Request” on page 411 for a detailed description of the information blocks returned.

  The output address is zero if no output was built. This can occur if nothing in the RECON satisfies the request or if an error occurs before any output could be built.

  The storage for the output blocks is not preallocated. DBRC acquires private storage for these blocks. The requesting application is responsible for freeing this storage by using the Buffer Release request (DSPAPI FUNC=RELBUF), specifying the returned output address.

- **RETCODE=-symbol | (2 - 12)**
  Optional parameter that specifies a place in storage to receive the return code.
code. If specified as a symbol, the symbol must be the label of a word of storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the return code is placed in register 15.

**RSNCODE** = *symbol* | (2 - 12)
Optional parameter that specifies a place in storage to receive the reason code. If specified as a symbol, the symbol must be the label of a word of storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the reason code is placed in register 0.

**VERSION** = 1.0 | *number*
Optional parameter that specifies the version number of the parameter list to be generated by this request.

To use the parameters associated with a version, you must specify the number of that version or a later version. If you specify an earlier version level, the parameter is not accepted for processing and an error message is issued at assembly time. If parameters have a version dependency, the parameter descriptions with each request type identify the version number required.

The default version is 1.0.

---

### Return and Reason Codes for the TYPE=RECON Query Request

Table 25 contains most of the return and reason codes for TYPE=RECON queries. The other possible return and reason codes relate to DBRC, not the query request. For a list of these return and reason codes, see “DBRC Service Aids” in the [IMS V9: Diagnosis Guide and Reference](#).

**Table 25. Return and Reason Codes for TYPE=RECON Queries**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000000'</td>
<td>X'00000000'</td>
<td>Request completed successfully.</td>
</tr>
</tbody>
</table>

**Severe Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0000000C'</td>
<td>X'C9000001'</td>
<td>Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.</td>
</tr>
</tbody>
</table>

**Storage Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000028'</td>
<td>X'D8100001'</td>
<td>Error obtaining storage for the RECON block.</td>
</tr>
</tbody>
</table>

**Internal Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0000002C'</td>
<td>all</td>
<td>DBRC internal error. Refer to “Database Recovery Control Service Aids” in the <a href="#">IMS V9: Diagnosis Guide and Reference</a> for further information.</td>
</tr>
</tbody>
</table>

**Parameter Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Invalid TOKEN field address. The address of the field containing the API TOKEN failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Invalid RETCODE field address. The address of the field to contain the API RETCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
</tbody>
</table>
Table 25. Return and Reason Codes for TYPE=RECON Queries (continued)

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Invalid RSNCODE field address. The address of the field to contain the API RSNCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Invalid OUTPUT field address. The address of the field to contain the OUTPUT address failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D8000001'</td>
<td>Parameter error. Missing or invalid OUTPUT parameter.</td>
</tr>
</tbody>
</table>

**Output for the TYPE=RECON Query Request**

Figure 54 illustrates the format of the output from a QUERY TYPE=RECON request. Figure 55 on page 412 is a sample DSECT describes in detail the fields of the storage blocks and their relationship to each other.

---

![Diagram](image-url)
<table>
<thead>
<tr>
<th>OFFSET</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>STRUCTURE</td>
<td>560</td>
<td>DSPAPQRC</td>
<td>Initialized with &quot;RECOVERY CONTROL DATASET&quot;</td>
</tr>
<tr>
<td>44</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQRCDATA</td>
<td>Offset to RECON data set info</td>
</tr>
<tr>
<td>48</td>
<td>SIGNED</td>
<td>4</td>
<td>*(2)</td>
<td>Reserved</td>
</tr>
<tr>
<td>56</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQRARCONINFOLEN</td>
<td>Length of each RECON dataset info element</td>
</tr>
<tr>
<td>58</td>
<td>UNSIGNED</td>
<td>1</td>
<td># of RECON dataset elements</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>CHARACTER</td>
<td>1</td>
<td>*</td>
<td>Reserved</td>
</tr>
<tr>
<td>60</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQR_FLAGS</td>
<td>Process flags...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1..</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.1..</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>..1..</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>....1...</td>
</tr>
<tr>
<td>61</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQR_FLAG2</td>
<td>More flags...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1..</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.1..</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>..1..</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>....1...</td>
</tr>
<tr>
<td>62</td>
<td>CHARACTER</td>
<td>2</td>
<td>*</td>
<td>Reserved</td>
</tr>
<tr>
<td>64</td>
<td>CHARACTER</td>
<td>136</td>
<td>APQR_CLEAN</td>
<td>Fields needed for cleanup</td>
</tr>
<tr>
<td>64</td>
<td>SIGNED</td>
<td>4</td>
<td>APQR_CSET</td>
<td>0 = updates not in progress, &gt;0 = update in progress</td>
</tr>
<tr>
<td>68</td>
<td>SIGNED</td>
<td>4</td>
<td>APQR_TYPE</td>
<td>Type of update</td>
</tr>
<tr>
<td>72</td>
<td>CHARACTER</td>
<td>32</td>
<td>APQR_OKEY</td>
<td>Key of original record</td>
</tr>
<tr>
<td>104</td>
<td>CHARACTER</td>
<td>32</td>
<td>APQR_BKEY</td>
<td>Key of base record that is in the process of being changed</td>
</tr>
<tr>
<td>136</td>
<td>CHARACTER</td>
<td>32</td>
<td>APQR_NKEY</td>
<td>Key of new record</td>
</tr>
<tr>
<td>168</td>
<td>CHARACTER</td>
<td>16</td>
<td>APQR_DBID</td>
<td>DBID of DBDS</td>
</tr>
<tr>
<td>168</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQR_DBD</td>
<td>DBD name</td>
</tr>
<tr>
<td>176</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQR_DDN</td>
<td>DD name</td>
</tr>
<tr>
<td>184</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQR_CAGRP</td>
<td>CA group name</td>
</tr>
<tr>
<td>192</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQR_DDNNEW</td>
<td>New DBDS DD name</td>
</tr>
<tr>
<td>200</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQR_DMBNO</td>
<td>DMB sequence number</td>
</tr>
<tr>
<td>202</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQR_LASTUSEDDB#</td>
<td>Last reused DMB number, valid only when apqr_DMBNO is 32767</td>
</tr>
<tr>
<td>204</td>
<td>CHARACTER</td>
<td>7</td>
<td>APQR_INITTOKEN</td>
<td>Recon init. token</td>
</tr>
<tr>
<td>211</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQR_CMDHLQ</td>
<td>Cmd auth high lvl qual</td>
</tr>
<tr>
<td>219</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQR_MVERS</td>
<td>Minimum IMS version</td>
</tr>
<tr>
<td>220</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQR_NWFLG</td>
<td>Fields needed for RECON I/O error(s)</td>
</tr>
</tbody>
</table>

Figure 55. DSECT of DSPAPQRC (Part 1 of 2)
Subsystem Query

The Subsystem query (DSPAPI FUNC=QUERY TYPE=SUBSYS) request is used to retrieve subsystem information from the RECON for a specific subsystem or all subsystems.

Syntax for the TYPE=SUBSYS Query Request

```
DSPAPI—FUNC=QUERY—TYPE=SUBSYS—TOKEN=address
```

Figure 55. DSECT of DSPAPQRC (Part 2 of 2)
**Parameters for the TYPE=SUBSYS Query Request**

The following are the possible parameters for QUERY TYPE=SUBSYS.

- **name**  
  Optional symbol you can specify. If used, begins in column 1.

- **TYPE=SUBSYS**  
  Specifies that subsystem information is requested.

- **TOKEN=symbol | (2 - 12)**  
  Required parameter that specifies the address of a 4-byte field that was returned on the FUNC=STARTDBRC request.

- **SSID=** | symbol | (2 - 12)**  
  Specifies the name of the subsystem being queried. The default (SSID=*) requests information about all of the subsystems.

- **SSYTYPE=ONLINE | BATCH | ALL**  
  Optional parameter that specifies the type of subsystem for which information is being requested. You cannot specify this parameter if you also specify a specific subsystem name for the SSID parameter or if you specify TRACKER=YES. SSType=ALL is the default.

- **TRACKER=YES | NO**  
  Optional parameter that specifies whether log information is being
QUERY TYPE=SUBSYS

requested for an RSR tracking subsystem. Specify TRACKER=YES to request the RSR tracker’s log information. Specify TRACKER=NO to request log information from a non-RSR subsystem or an active subsystem of an RSR system. TRACKER=NO is the default.

MF=S | L | M | E
Optional parameter that specifies the macro form of the request. See “Macro Forms of the DSPAPI Macro” on page 352 for the details of the parameter values.

OUTPUT=symbol | (2 - 12)
Required parameter that specifies a field to receive a pointer to the first block of a possible chain of subsystem information blocks. See “Output for the TYPE=SUBSYS Query Request” on page 417 for a detailed description of the information blocks returned.

The output address is zero if no output was built. This can occur if nothing in the RECON satisfies the request or if an error occurs before any output could be built.

The storage for the output blocks is not preallocated. DBRC acquires private storage for these blocks. The requesting application is responsible for freeing this storage by using the Buffer Release request (DSPAPI FUNC=RELBUF), specifying the returned output address.

RETCODE=symbol | (2 - 12)
Optional parameter that specifies a place in storage to receive the return code. If specified as a symbol, the symbol must be the label of a word of storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the return code is placed in register 15.

RSNCODE=symbol | (2 - 12)
Optional parameter that specifies a place in storage to receive the reason code. If specified as a symbol, the symbol must be the label of a word of storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the reason code is placed in register 0.

VERSION=1.0 | number
Optional parameter that specifies the version number of the parameter list to be generated by this request.

To use the parameters associated with a version, you must specify the number of that version or a later version. If you specify an earlier version level, the parameter is not accepted for processing and an error message is issued at assembly time. If parameters have a version dependency, the parameter descriptions with each request type identify the version number required.

The default version is 1.0.

Return and Reason Codes for the TYPE=SUBSYS Query Request

Table 26 contains most of the return and reason codes for TYPE=SUBSYS query requests. The other possible return and reason codes relate to DBRC, not the query request. For a list of these return and reason codes, see “DBRC Service Aids” in the IMS V9: Diagnosis Guide and Reference.

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X’00000000’</td>
<td>X’00000000’</td>
<td>Request completed successfully.</td>
</tr>
</tbody>
</table>
Table 26. Return and Reason Codes for TYPE=SUBSYS Query Requests  (continued)

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X’00000008’</td>
<td>X’D8600001’</td>
<td>No subsystem record of the requested type, active or tracker, exists.</td>
</tr>
<tr>
<td>X’00000008’</td>
<td>X’D8600002’</td>
<td>No batch or online subsystem records exist.</td>
</tr>
<tr>
<td><strong>Severe Error</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X’0000000C’</td>
<td>X’C9000001’</td>
<td>Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.</td>
</tr>
<tr>
<td><strong>Storage Error</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X’00000028’</td>
<td>X’D8600001’</td>
<td>Error obtaining storage for SUBSYS block.</td>
</tr>
<tr>
<td>X’00000028’</td>
<td>X’D9100001’</td>
<td>An error occurred processing the request. DBRC will release storage that was obtained up to this point. However, another error was encountered during the attempt to release storage.</td>
</tr>
<tr>
<td><strong>Internal Error</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X’0000002C’</td>
<td>all</td>
<td>DBRC internal error. Refer to &quot;Database Recovery Control Service Aids&quot; in the [IMS V9: Diagnosis Guide and Reference for further information.</td>
</tr>
<tr>
<td><strong>Parameter Error</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X’00000030’</td>
<td>X’C9000001’</td>
<td>Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.</td>
</tr>
<tr>
<td>X’00000030’</td>
<td>X’C9000002’</td>
<td>Invalid TOKEN field address. The address of the field containing the API TOKEN failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X’00000030’</td>
<td>X’C9000003’</td>
<td>Invalid RETCODE field address. The address of the field to contain the API RETCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X’00000030’</td>
<td>X’C9000004’</td>
<td>Invalid RSNCODE field address. The address of the field to contain the API RSNCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X’00000030’</td>
<td>X’C9000005’</td>
<td>Invalid OUTPUT field address. The address of the field to contain the OUTPUT address failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X’00000030’</td>
<td>X’C9000010’</td>
<td>Invalid SSID field address. The address of the field containing the SSID failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X’00000030’</td>
<td>X’D8000001’</td>
<td>Parameter error. Missing or invalid OUTPUT parameter.</td>
</tr>
<tr>
<td>X’00000030’</td>
<td>X’D8600001’</td>
<td>Parameter error. SSTYPE=ONLINE or BATCH can not be specified on requests for a specific SSID or for a tracking subsystem request (TRACKER=YES).</td>
</tr>
</tbody>
</table>
Output for the TYPE=SUBSYS Query Request

Figure 56 illustrates the format of the output from a QUERY TYPE=SUBSYS request. Figure 57 is a sample DSECT describes in detail the fields of the storage blocks and their relationship to each other.

<table>
<thead>
<tr>
<th>OFFSET</th>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>STRUCTURE</td>
<td>64</td>
<td>DSPAPQSS</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQSS_SSID</td>
<td>Subsystem identifier</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>UNSIGNED</td>
<td>4</td>
<td>APQSS_AUTHLIST</td>
<td>Offset to auth db/Area list</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>C</td>
<td>SIGNED</td>
<td>4</td>
<td>APQSS_AUTHCOUNT</td>
<td>Number of authorized DB/Areas</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>10</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQSS_AUTHLEN</td>
<td>Length of auth entry</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>12</td>
<td>CHARACTER</td>
<td>6</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>18</td>
<td>CHARACTER</td>
<td>12</td>
<td>APQSS_LOGTIME</td>
<td>Start time of log</td>
</tr>
<tr>
<td>36</td>
<td>24</td>
<td>24</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQSS_RELLEV</td>
<td>Subsystem release level '71'X=V7R1, '81'X=V8R1, '91'X=V9R1, etc.</td>
</tr>
<tr>
<td>37</td>
<td>25</td>
<td>25</td>
<td>CHARACTER</td>
<td>1</td>
<td>APQSS_COEXLVL</td>
<td>Coexistence level</td>
</tr>
<tr>
<td>38</td>
<td>26</td>
<td>26</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQSS_IRLMCT</td>
<td>IRLM status count</td>
</tr>
<tr>
<td>39</td>
<td>27</td>
<td>27</td>
<td>CHARACTER</td>
<td>1</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>28</td>
<td>28</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQSS_GSGNAME</td>
<td>Global Service Group name</td>
</tr>
<tr>
<td>48</td>
<td>30</td>
<td>30</td>
<td>CHARACTER</td>
<td>5</td>
<td>APQSS_IRLMID</td>
<td>IRLM ID of SS</td>
</tr>
<tr>
<td>53</td>
<td>35</td>
<td>35</td>
<td>CHARACTER</td>
<td>5</td>
<td>APQSS_IRLMBK</td>
<td>IRLM ID of backup SS</td>
</tr>
<tr>
<td>58</td>
<td>3A</td>
<td>3A</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQSS_FLAGS</td>
<td>Flags</td>
</tr>
<tr>
<td>59</td>
<td>3B</td>
<td>3B</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQSS_FLAGS2</td>
<td>Flags 2</td>
</tr>
<tr>
<td>60</td>
<td>3C</td>
<td>3C</td>
<td>SIGNED</td>
<td>2</td>
<td>APQSS_BCKTKN</td>
<td>Backup recovery token</td>
</tr>
<tr>
<td>62</td>
<td>3E</td>
<td>3E</td>
<td>CHARACTER</td>
<td>2</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

Figure 56. Format for QUERY TYPE=SUBSYS Output

Figure 57. DSECT of DSPAPQSS (Part 1 of 2)
### OFFSETS

<table>
<thead>
<tr>
<th>DECIMAL</th>
<th>HEX</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>NAME (DIM)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0)</td>
<td>STRUCTURE</td>
<td>32</td>
<td>APQSS_AUTHNAME</td>
<td>Names of authd DB/Areas</td>
</tr>
<tr>
<td>0</td>
<td>(0)</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQSS_DBNAME</td>
<td>DB name</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>CHARACTER</td>
<td>8</td>
<td>APQSS_AREANM</td>
<td>If FP, Area name</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQSS_SHRLVL</td>
<td>Share level</td>
</tr>
<tr>
<td>17</td>
<td>(11)</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQSS_DBACCS</td>
<td>Access intent</td>
</tr>
<tr>
<td>18</td>
<td>(12)</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQSS_DBNCOD</td>
<td>Encoded state</td>
</tr>
<tr>
<td>19</td>
<td>(13)</td>
<td>UNSIGNED</td>
<td>1</td>
<td>APQSS_DBSTAT</td>
<td>DB status flags</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>UNSIGNED</td>
<td>2</td>
<td>APQSS_DBEQCT</td>
<td>DB EQE count</td>
</tr>
<tr>
<td>22</td>
<td>(16)</td>
<td>SIGNED</td>
<td>2</td>
<td>APQSS_GLBDMB</td>
<td>Global DMB number</td>
</tr>
<tr>
<td>24</td>
<td>(18)</td>
<td>BIT(8)</td>
<td>1</td>
<td>APQSS_AUTHFLAGS</td>
<td>Flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1... ...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.1... ...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...1 ...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...1 ...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.... 1...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.... 1...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.... ...1</td>
</tr>
<tr>
<td>25</td>
<td>(19)</td>
<td>CHARACTER</td>
<td>7</td>
<td>*</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

### CONSTANTS

<table>
<thead>
<tr>
<th>LEN</th>
<th>TYPE</th>
<th>VALUE</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>DSPAPQSS</td>
<td>APQSS_EYECATCHER</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 57. DSECT of DSPAPQSS (Part 2 of 2)*
Chapter 18. DBRC Release Buffer Request

Use the DBRC release buffer request (DSPAPI FUNC=RELBUF) to release storage that was acquired as a result of a DBRC query request. The query request returns a chain of one or more blocks containing the requested information. It is the caller’s responsibility to ensure that the storage DBRC allocated for these blocks is freed.

In This Chapter:
- "RELBUF Request"

RELBUF Request

"RELBUF Usage" contains the details for the release buffer request.

RELBUF Usage

Syntax for the RELBUF Request

```
name DSPAPI—FUNC=RELBUF—TOKEN=address—BUFFER=buffer

MF=S

MF=(S,list)

MF=(L,list)

MF=(M,list)

MF=(E,list)

RETCODE=returncode

VERSION=1.0

RSNCODE=reasoncode

VERSION=number
```

Parameters for the RELBUF Request

The following are the possible parameters for RELBUF.

`name`  Optional symbol you can specify. If used, begins in column 1.

`TOKEN=symbol | (2 - 12)`
Required parameter that specifies the address of a 4-byte field that was returned on the FUNC=STARTDBRC request.

`BUFFER=symbol | (2 - 12)`
Required parameter that specifies the address of the first block of a chain of one or more information blocks to release. This storage was acquired by DBRC in order to satisfy another DBRC request.

`RETCODE=symbol | (2 - 12)`
Optional parameter that specifies a place in storage to receive the return code. If specified as a symbol, the symbol must be the label of a word of
storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the return code is placed in register 15.

RSNCODE=symbol (2 - 12)
Optional parameter that specifies a place in storage to receive the reason code. If specified as a symbol, the symbol must be the label of a word of storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the reason code is placed in register 0.

MF=S | L | M | E
Optional parameter that specifies the macro form of the request. See "Macro Forms of the DSPAPI Macro" on page 352 for the details of the parameter values.

VERSION=1.0 | number
Optional parameter that specifies the version number of the parameter list to be generated by this request.

To use the parameters associated with a version, you must specify the number of that version or a later version. If you specify an earlier version level, the parameter is not accepted for processing and an error message is issued at assembly time. If parameters have a version dependency, the parameter descriptions with each request type identify the version number required.

The default version is 1.0.

Return and Reason Codes for RELBUF
Table 27 contains the return and reason codes for the RELBUF request.

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000000'</td>
<td>X'00000000'</td>
<td>Request completed successfully.</td>
</tr>
</tbody>
</table>

Partial Success

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000004'</td>
<td>X'D9000001'</td>
<td>There was no storage to free. An address of zero was passed in the BUFFER parameter.</td>
</tr>
</tbody>
</table>

Severe Error

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0000000C'</td>
<td>X'C9000001'</td>
<td>Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.</td>
</tr>
</tbody>
</table>

Storage Error

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000028'</td>
<td>X'D9100001'</td>
<td>An error occurred attempting to release storage.</td>
</tr>
</tbody>
</table>

Parameter Error

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000002'</td>
<td>Invalid TOKEN field address. The address of the field containing the API TOKEN failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000003'</td>
<td>Invalid RETCODE field address. The address of the field to contain the API RETCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>Return Codes</td>
<td>Reason Codes</td>
<td>Meaning</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000004'</td>
<td>Invalid RSNCODE field address. The address of the field to contain the API RSNCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000007'</td>
<td>Invalid BUFFER address. The address of the BUFFER to be released failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D9100001'</td>
<td>The header of a storage block to be freed is invalid.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'D9100002'</td>
<td>The length of a storage block to be freed is not greater than zero.</td>
</tr>
</tbody>
</table>
Chapter 19. DBRC Start Request

Use the DBRC start request (STARTDBRC) to initialize the DBRC API and to start DBRC.

In This Chapter:
- "STARTDBRC Request"  

STARTDBRC Request

"STARTDBRC Usage" contains the details for the DBRC start request.

STARTDBRC Usage

Syntax for the STARTDBRC Request

`name`  
`DSPAPI FUNC=STARTDBRC`  
`SYSPRINT=ddname`  
`MF=S`  
`MF=(S,list)`  
`MF=(L,list)`  
`MF=(M,list),COMPLETE`  
`MF=(E,list),COMPLETE`  
`MF=(L,list),NOCHECK`  
`MF=(M,list),NOCHECK`  
`RETCODE=returncode`  
`RSNCODE=reasoncode`  
`VERSION=1.0`  
`VERSION=number`  

Parameters for STARTDBRC

The following are the possible parameters for STARTDBRC.

`name` Optional symbol you can specify. If used, begins in column 1.

`TOKEN=-symbol | (2 - 12)` Required parameter that specifies the address of a 4-byte field to receive the API token. This token must be included in all subsequent requests associated with this STARTDBRC request.

`SYSPRINT= symbol | (2 - 12)` Optional parameter that specifies the DDname of an output data set to be used for messages. If omitted, the default name SYSPRINT is used.

`RETCODE= symbol | (2 - 12)` Optional parameter that specifies a place in storage to receive the return code. If specified as a symbol, the symbol must be the label of a word of storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the return code is placed in register 15.
**STARTDBRC**

\[\text{RSNCODE=}&\text{symbol} \mid (2 - 12)\]

Optional parameter that specifies a place in storage to receive the reason code. If specified as a symbol, the symbol must be the label of a word of storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the reason code is placed in register 0.

\[\text{MF=S} \mid \text{L} \mid \text{M} \mid \text{E}\]

Optional parameter that specifies the macro form of the request. See “Macro Forms of the DSPAPI Macro” on page 352 for the details of the parameter values.

\[\text{VERSION=1.0} \mid \text{number}\]

Optional parameter that specifies the version number of the parameter list to be generated by this request.

To use the parameters associated with a version, you must specify the number of that version or a later version. If you specify an earlier version level, the parameter is not accepted for processing and an error message is issued at assembly time. If parameters have a version dependency, the parameter descriptions with each request type identify the version number required.

The default version is 1.0.

**Return and Reason Codes for STARTDBRC**

Table 28 contains most of the return and reason codes for the STARTDBRC request. The other possible return and reason codes relate to DBRC, not the request. For a list of these return and reason codes, see “DBRC Service Aids” in the IMS V9: Diagnosis Guide and Reference.

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000000'</td>
<td>X'00000000'</td>
<td>Request completed successfully.</td>
</tr>
</tbody>
</table>

**Severe Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000000C'</td>
<td>X'C9000002'</td>
<td>DBRC could not successfully initialize during phase 0 of DBRC initialization</td>
</tr>
<tr>
<td>X'00000000C'</td>
<td>X'C9000003'</td>
<td>DBRC could not successfully initialize during phase 1 of DBRC initialization.</td>
</tr>
<tr>
<td>X'00000000C'</td>
<td>X'C9000004'</td>
<td>A z/OS LOAD failed for the DBRC module DSPCRTR0.</td>
</tr>
<tr>
<td>X'00000000C'</td>
<td>X'C9000005'</td>
<td>STORAGE request failure. The API cannot obtain storage necessary to complete the request.</td>
</tr>
<tr>
<td>X'00000000C'</td>
<td>X'C90000012'</td>
<td>STARTDBRC has been previously issued without an intervening STOPDBRC.</td>
</tr>
</tbody>
</table>

**Parameter Error**

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000002'</td>
<td>Invalid TOKEN field address. The address of the field to contain the API TOKEN failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000003'</td>
<td>Invalid RETCODE field address. The address of the field to contain the API RETCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>Return Codes</td>
<td>Reason Codes</td>
<td>Meaning</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000004'</td>
<td>Invalid RSNCODE field address. The address of the field to contain the API RSNCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000006'</td>
<td>Invalid SYSPRINT field address. The address of the field containing the SYSPRINT DD name failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
</tbody>
</table>
Chapter 20. DBRC Stop Request

Use the DBRC stop request (STOPDBRC) to terminate the DBRC API and to stop DBRC.

In This Chapter:
* "STOPDBRC Request"

STOPDBRC Request

"STOPDBRC Usage" contains the details for the release buffer request.

STOPDBRC Usage

Syntax for the STOPDBRC Request

```
-DSPAPI-FUNC=STOPDBRC-TOKEN=address
```

Parameters for STOPDBRC

The following are the possible parameters for STOPDBRC.

- **name**  Optional symbol you can specify. If used, begins in column 1.

- **TOKEN=**  symbol  |  (2 - 12)
  Required parameter that specifies the address of a 4-byte field that was returned on the FUNC=STARTDBRC request.

- **RETCODE=**  symbol  |  (2 - 12)
  Optional parameter that specifies a place in storage to receive the return code. If specified as a symbol, the symbol must be the label of a word of storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the return code is placed in register 15.

- **RSNCODE=**  symbol  |  (2 - 12)
  Optional parameter that specifies a place in storage to receive the reason code. If specified as a symbol, the symbol must be the label of a word of storage. If specified as a register, the register must contain the address of a word of storage. If not specified, the reason code is placed in register 0.
STOPDBRC

MF=S | L | M | E
Optional parameter that specifies the macro form of the request. See “Macro Forms of the DSPAPI Macro” on page 352 for the details of the parameter values.

VERSION=1.0 | number
Optional parameter that specifies the version number of the parameter list to be generated by this request.

To use the parameters associated with a version, you must specify the number of that version or a later version. If you specify an earlier version level, the parameter is not accepted for processing and an error message is issued at assembly time. If parameters have a version dependency, the parameter descriptions with each request type identify the version number required.

The default version is 1.0.

Return and Reason Codes for STOPDBRC
Table 29 contains the return and reason codes for the STOPDBRC request. The other possible return and reason codes relate to DBRC, not the request. For a list of these return and reason codes, see “DBRC Service Aids” in the IMS V9: Diagnosis Guide and Reference.

Table 29. Return and Reason Codes for the STOPDBRC Request

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000000'</td>
<td>X'00000000'</td>
<td>Request completed successfully.</td>
</tr>
</tbody>
</table>

Severe Error

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0000000C'</td>
<td>X'C9000001'</td>
<td>Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.</td>
</tr>
</tbody>
</table>

Parameter Error

<table>
<thead>
<tr>
<th>Return Codes</th>
<th>Reason Codes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000030'</td>
<td>X'C9000001'</td>
<td>Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000002'</td>
<td>Invalid TOKEN field address. The address of the field containing the API TOKEN failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000003'</td>
<td>Invalid RETCODE field address. The address of the field to contain the API RETCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
<tr>
<td>X'00000030'</td>
<td>X'C9000004'</td>
<td>Invalid RSNCODE field address. The address of the field to contain the API RSNCODE failed validity checking. The address specifies storage not owned by the calling program.</td>
</tr>
</tbody>
</table>
Part 4. Appendixes
Appendix A. Symbolic Keywords Recognized by DBRC

Table 30 through Table 36 on page 438 show the symbolic keywords that DBRC recognizes in the IBM-supplied skeletal JCL execution members.

Table 30 shows the keywords that all supported utilities recognize.

Table 31 on page 432 through Table 36 on page 438 show the keywords that each individual utility recognizes.

These tables and the figures in “IBM-Supplied Skeletal JCL Execution Members” on page 435 are presented in alphabetical order of the GENJCL commands:

- **GENJCL.ARCHIVE**, which generates JCL for the Log Archive utility. See “Log Archive Utility (ARCHJCL)” on page 432 for a list of the symbolic keywords supported by the utility and see “GENJCL.ARCHIVE” on page 217 for details using the command.

- **GENJCL.CA**, which generates JCL for the Database Change Accumulation utility. See “Database Change Accumulation Utility (CAJCL)” on page 433 for a list of the symbolic keywords supported by the utility and see “GENJCL.CA” on page 221 for details of the command usage.

- **GENJCL.CLOSE**, which generates JCL for the Log Recovery utility. See “Log Recovery Utility (LOGCLJCL)” on page 434 for a list of the symbolic keywords supported by the utility and see “GENJCL.CLOSE” on page 225 for details of the command usage.

- **GENJCL.IC** and **GENJCL.OIC**, which generates JCL for the Database Image Copy utilities. See “Database Image Copy Utility, Database Image Copy Utility 2, and Online Database Image Copy Utility (ICJCL and OICJCL)” on page 434 for a list of the symbolic keywords supported by the utilities and see “GENJCL.IC” on page 229 and “GENJCL.OIC” on page 238 for details of the commands usage.

- **GENJCL.RECEIVE**, which generates JCL for the Database Recovery utility. See “Database Recovery Utility-Receive (ICRCVJCL)” on page 436 for a list of the symbolic keywords supported by the utility and see “GENJCL.RECEIVE” on page 243 for details of the command usage.

- **GENJCL.RECOV**, which also generates JCL for the Database Recovery utility. See “Database Recovery Utility-Recovery (RECOVJCL)” on page 438 for a list of the symbolic keywords supported by the utility and see “GENJCL.RECOV” on page 247 for details of the command usage.

All Supported Utilities

Table 30 explains the symbolic keywords recognized by all the supported utilities.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%RCNDSN1</td>
<td>Name of the RECON1 data set if RECONs are allocated by JCL. Set to null if RECONs are dynamically allocated.</td>
</tr>
<tr>
<td>%RCNDSN2</td>
<td>Name of the RECON2 data set if RECONs are allocated by JCL. Set to null if RECONs are dynamically allocated.</td>
</tr>
<tr>
<td>%RCNDSN3</td>
<td>Name of the RECON3 data set if RECONs are allocated by JCL. Set to null if RECONs are dynamically allocated.</td>
</tr>
</tbody>
</table>
Symbolic Keywords

Table 30. Symbolic Keywords for All Supported Utilities (continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%STPN0</td>
<td>The current step number. The step number is set to 0 if the JOB parameter was specified on the GENJCL command. The step number is increased by 1 when DBRC first encounters it in a skeletal JCL execution member. The step number remains at that value while the execution member is processed. As the keyword is encountered in the remaining skeletal JCL, the current value is substituted. The JCL execution member can be processed again because of a multi-step generation, or because the subsequent GENJCL command specifies NOJOB. When it is processed again the step number is increased by 1 from its current value when the keyword is first encountered in the next skeletal JCL execution member. This increase takes place before the keyword value is substituted.</td>
</tr>
<tr>
<td>%TIME</td>
<td>The time of day, in the form hhmmss.</td>
</tr>
<tr>
<td>%GRPINDX</td>
<td>The DBDS group member index. This keyword is set only when a DBDS group is specified, implicitly or explicitly, on the GENJCL command. (A DBDS group can be specified on the GENJCL.IC, GENJCL.OIC, GENJCL.RECOV, and GENJCL.USER commands.) When you specify a DBDS group, the keyword is initialized to 1. It is then increased by 1 as each successive group member is processed.</td>
</tr>
<tr>
<td>%CNTR</td>
<td>A counter controlled by DBRC. The counter is set to 0 whenever the first GENJCL command is issued or a JOB statement is reproduced from the skeletal JCL execution member JOBJCL. DBRC increases the counter by each time the keyword is encountered in a skeletal JCL execution member. The JCL execution member can be processed again because of a multi-step generation, or because the subsequent GENJCL command specifies NOJOB. If so, the counter continues to increase from its current value when the keyword is encountered in next skeletal JCL execution member. This increase takes place before the keyword value is substituted.</td>
</tr>
<tr>
<td>%DATE</td>
<td>The day of the year, in the form yyddd.</td>
</tr>
<tr>
<td>%DATE7</td>
<td>The day of the year, in the form yyyyddd.</td>
</tr>
</tbody>
</table>

Log Archive Utility (ARCHJCL)

Table 31 explains the symbolic keywords recognized by the Log Archive utility.

Table 31. Symbolic Keywords for Log Archive Utility

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%SSID</td>
<td>The subsystem ID, which is set from the SSID parameter on the GENJCL.ARCHIVE command. If the SSID parameter is not specified, the default subsystem ID is used. The default subsystem ID is set by you in the INIT, RECON or CHANGE.RECON command. If no default subsystem ID was specified, the command fails.</td>
</tr>
<tr>
<td>%DDNAMES</td>
<td>The ddnames of the OLDSs that are to be archived. If ALL is specified or used as the default on the GENJCL.ARCHIVE command, the ddnames of all unarchived OLDSs are determined from RECON. Otherwise, the ddnames specified on the command are used.</td>
</tr>
<tr>
<td>%OLDSDDN</td>
<td>The ddname of one or more specific OLDS.</td>
</tr>
<tr>
<td>%OLDSDNS</td>
<td>The data set name of the one or more OLDS.</td>
</tr>
<tr>
<td>%ARDATE</td>
<td>The date (from the open time stamp) of the first OLDS that is to be archived. The date is in the form yyddd where: yy is the year ddd is the day</td>
</tr>
</tbody>
</table>
Symbolic Keywords

### Table 31. Symbolic Keywords for Log Archive Utility (continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
</table>
| %ARDATE7  | The date (from the open time stamp) of the first OLDS that is to be archived. The date is in the form yyyysddd where:  

  - yyyy is the 4-digit year  
  - ddd is the Julian day |
| %ARTIME   | The time (from the open time stamp) of the first OLDS that is to be archived. The time is in the form hhmmssT where:  

  - hh is the hour  
  - mm is the minute  
  - ss is the second  
  - t is the tenth of a second |
| %ARVERS   | The archive version number of the first OLDS to be archived.                 |
| %ARSLDS   | Set to YES when the SLDS parameter is specified.                             |

### Database Change Accumulation Utility (CAJCL)

Table 32 explains the symbolic keywords recognized by the Database Change Accumulation utility.

### Table 32. Symbolic Keywords for Database Change Accumulation Utility

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%CAGRP</td>
<td>The CA group name.</td>
</tr>
<tr>
<td>%DSLLGTM</td>
<td>The start time for selecting input log data. If an input change accumulation data set is used, %DSLLGTM is set to the volume stop time of the last-accumulated log volume.</td>
</tr>
<tr>
<td>%CAODSN</td>
<td>The data set name of the input change accumulation data set. This keyword is set to null if no existing change accumulation data set is defined in RECON for the CA group.</td>
</tr>
<tr>
<td>%CAOUNIT</td>
<td>The unit type of the input change accumulation data set. This keyword is set to null if no existing change accumulation data set is defined in RECON for the CA group.</td>
</tr>
<tr>
<td>%CAOVOLS</td>
<td>The volume serial number list of the input change accumulation data set. This keyword is set to null if there is no existing change accumulation data set is defined in RECON for the CA group.</td>
</tr>
<tr>
<td>%CAOFSEQ</td>
<td>The file sequence number of the input change accumulation data set. This keyword is set to null if no existing change accumulation data set is defined in RECON for the CA group.</td>
</tr>
</tbody>
</table>
| %CANDSN   | The data set name of the output change accumulation data set. If REUSE is specified for the CA group, the keyword is set from information in RECON. If NOREUSE is specified, DBRC generates a data set name. The generated name is:  

  IMS.cagrpname.CA.CAhhmmss  

  where cagrpname is the CA group name, and hhmmss is the current time of day. |
| %CANUNIT  | The unit type of the output change accumulation data set. If REUSE is specified for the CA group, the keyword is set from information in RECON. If NOREUSE is specified, this keyword is set from the UNIT parameter on the GENJCL.CA command. If UNIT is not specified, the keyword is set to 3400. |
| %CANVCNT  | The number of volumes in the output change accumulation data set. If REUSE is specified for the CA group, the keyword is set from information in RECON. If NOREUSE is specified, this keyword is set from the VOLLIST parameter on the GENJCL.CA command. |
| %CANVOLS  | The volume serial number list of the output change accumulation data set. If REUSE is specified for the CA group, the keyword is set from information in RECON. If NOREUSE is specified, this keyword is set from the VOLLIST parameter on the GENJCL.CA command. |
Symbolic Keywords

Table 32. Symbolic Keywords for Database Change Accumulation Utility (continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%CABFSEQ</td>
<td>The file sequence number of the output change accumulation data set. If REUSE is specified for the CA group, the keyword is set from information in RECON. If NOREUSE is specified, this keyword is set to 1.</td>
</tr>
<tr>
<td>%LOGDSN</td>
<td>The data set name of the log data set.</td>
</tr>
<tr>
<td>%LOGUNIT</td>
<td>The unit type of the log data set.</td>
</tr>
<tr>
<td>%LOGVSEQ</td>
<td>The volume sequence number of the log data set.</td>
</tr>
<tr>
<td>%LOGVOLS</td>
<td>The volume serial numbers of the log data set.</td>
</tr>
<tr>
<td>%LOGFSEQ</td>
<td>The file sequence number of the log data set.</td>
</tr>
<tr>
<td>%LOGSEL</td>
<td>Set to YES if any log data sets were selected. Otherwise, set to NO.</td>
</tr>
<tr>
<td>%CADB0</td>
<td>This keyword generates the DB0 control statements for the Database Change Accumulation utility. Note: %CADB0 always generates DB0 control statements with the time stamps in UTC format (indicated by an offset of zeroes).</td>
</tr>
</tbody>
</table>

Log Recovery Utility (LOGCLJCL)

Table 33 explains the symbolic keywords recognized by the Log Recovery utility.

Table 33. Symbolic Keywords for Log Recovery Utility

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%SSID</td>
<td>The subsystem ID, which is set from the SSID parameter on the GENJCL.CLOSE command. If the SSID parameter is not specified, the default subsystem ID is used. The default subsystem ID is set by you in the INIT, RECON or CHANGE.RECON command. If no default subsystem ID is specified, the command fails.</td>
</tr>
<tr>
<td>%CDDNAME</td>
<td>The ddname of the OLDS to be closed. This keyword is set from the OLDS parameter on the GENJCL.CLOSE command. If GENJCL.CLOSE did not specify an OLDS, the most recent open OLDS for the specified subsystem is used.</td>
</tr>
<tr>
<td>%OLDS_TYP</td>
<td>The type of OLDS, primary or secondary (set to P or S, respectively).</td>
</tr>
<tr>
<td>%OLDSdsn</td>
<td>The data set name of the OLDS.</td>
</tr>
<tr>
<td>%WADS</td>
<td>If the OLDS to be closed is currently open, this keyword is set to YES. Otherwise, this keyword is set to NO.</td>
</tr>
<tr>
<td>%WDDNAME</td>
<td>The ddname of the ‘next OLDS’ to be used to close the OLDS. If %WADS is set to NO, this keyword is set to the ddname of the OLDS used immediately after the OLDS being closed. If %WADS is set to YES, this keyword is set to null.</td>
</tr>
<tr>
<td>%PDDNAME</td>
<td>The ddname of the immediately prior OLDS to be used to close the OLDS by providing a last block sequence number for base point information.</td>
</tr>
</tbody>
</table>

Database Image Copy Utility, Database Image Copy Utility 2, and Online Database Image Copy Utility (ICJCL and OICJCL)

Table 34 explains the symbolic keywords recognized by the Database Image Copy utilities.

Table 34. Symbolic Keywords for Database Image Copy Utilities

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%PSB</td>
<td>The PSB name, which is set from the PSB parameter on the GENJCL command. This keyword is applicable only for the Database Online Image Copy utility.</td>
</tr>
</tbody>
</table>
**Symbolic Keywords**

**Table 34. Symbolic Keywords for Database Image Copy Utilities (continued)**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%DBNAME</td>
<td>The database name, which is set from the DBD parameter on the GENJCL command.</td>
</tr>
<tr>
<td>%DBDDN</td>
<td>The DBDS ddname, which is set from the DDN parameter on the GENJCL command.</td>
</tr>
<tr>
<td>%DBDSN</td>
<td>The DBDS data set name, which is set from the DBDS record in RECON.</td>
</tr>
<tr>
<td>%DBDSAM</td>
<td>This keyword is set to VSAM for VSAM DBDS. Otherwise, it is set to null.</td>
</tr>
<tr>
<td>%DBADDSN</td>
<td>For DEDBs, the ddname of the ADS. Otherwise, set to null. This keyword is applicable only for the Database Image Copy utility.</td>
</tr>
<tr>
<td>%DBADDSAV</td>
<td>For DEDBs, set to AVAIL if RECON indicates that the ADS is available, or UNAVAIL if the ADS is unavailable. For other types of databases, this keyword is set to null. This keyword is applicable only for the Database Image Copy utility.</td>
</tr>
<tr>
<td>%COPIES</td>
<td>The number of image copy data sets to be produced. This keyword is set to 1 or 2 from the COPIES parameter on the GENJCL command.</td>
</tr>
<tr>
<td>%MDBNAME</td>
<td>The HALDB master name, if this is a DBDS of a HALDB partition. This keyword is set to NULL for non-HALDBs.</td>
</tr>
<tr>
<td>%SMS</td>
<td>Indicates whether a Database Image Copy 2 (DFSUMDT0) image copy data set is being used for the requested utility execution. If used, the keyword is set to 1; otherwise, the keyword is set to 0.</td>
</tr>
<tr>
<td>%ICDDN1, %ICDDN2, %ICDDN3, %ICDDN4</td>
<td>The DD name of the first image copy data set. If multiple image copy data sets are produced, %ICDDN1, %ICDDN2, %ICDDN3, and %ICDDN4 are similarly set.</td>
</tr>
<tr>
<td>%ICSYSIN</td>
<td>The Database Image Copy utility control statement. Columns in the statement are set as follows:</td>
</tr>
<tr>
<td>Column</td>
<td>Setting</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Number of image copy data sets to be produced (either 1 or 2)</td>
</tr>
<tr>
<td>4-11</td>
<td>Database name</td>
</tr>
<tr>
<td>13-20</td>
<td>ddname of the DBDS</td>
</tr>
<tr>
<td>22-30</td>
<td>ddname of the primary image copy data set</td>
</tr>
<tr>
<td>31-38</td>
<td>ddname of the duplicate image copy data set, if one is produced.</td>
</tr>
<tr>
<td>40-43</td>
<td>Checkpoint interval (applicable only for Online Database Image Copy utility). All other columns are set to blanks.</td>
</tr>
<tr>
<td>%ICDSN1, %ICDSN2, %ICDSN3, %ICDSN4</td>
<td>The data set name of the image copy data set is %ICDSN1.</td>
</tr>
</tbody>
</table>

If NOREUSE is specified for the DBDS, DBRC generates the following data set name:

\[
\text{IMS.dbname.ddname.IC.IChhmmss}
\]

where:
- dbname is the database name of the DBDS
- ddname is the ddname of the DBDs
- hhmss is the current time of day

If Sameds is specified for a group of DBDSs, DBRC generates the following data set name:

\[
\text{IMSVS.GROUP.grpname.IC.IChhmmss}
\]

where:
- grpname is the name specified in the GROUP or DBD parameter
- hhmms is the current time of day

If multiple image copy data sets are to be produced, %ICDSN3, or %ICDSN4 are set similarly.
Symbolic Keywords

Table 34. Symbolic Keywords for Database Image Copy Utilities (continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%ICUNIT1, %ICUNIT2, %ICUNIT3, %ICUNIT4</td>
<td>The unit type of the image copy data set. If NOREUSE is specified for the DBDS, %ICUNIT1 is set from the UNIT parameter on the command. If multiple image copy data sets are to be produced, %ICUNIT2, %ICUNIT3, or %ICUNIT4 are set similarly.</td>
</tr>
<tr>
<td>%ICFSEQ1, %ICFSEQ2, %ICFSEQ3, %ICFSEQ4</td>
<td>The file sequence number of the image copy data set. If NOREUSE is specified for the DBDS, %ICFSEQ1 is set to 1. If multiple image copy data sets are to be produced, %ICFSEQ2, %ICFSEQ3, or %ICFSEQ4 are set similarly.</td>
</tr>
<tr>
<td>%ICVOLS1, %ICVOLS2, %ICVOLS3, %ICVOLS4</td>
<td>The volume serial number of the image copy data set. If NOREUSE is specified for the DBDS, %ICVOLS1 is set from the VOLLIST parameter on the command. If multiple image copy data sets are to be produced, %ICVOLS2, %ICVOLS3, or %ICVOLS4 are set similarly.</td>
</tr>
<tr>
<td>%ICVCNT1, %ICVCNT2, %ICVCNT3, %ICVCNT4</td>
<td>The number of volumes of the image copy data set. If NOREUSE is specified for the DBDS, %ICVCNT1 is set to the number of volumes specified on the VOLLIST parameter on the command. If multiple image copy data sets are to be produced, %ICVCNT2, %ICVCNT3, or %ICVCNT4 is set similarly.</td>
</tr>
<tr>
<td>%SMGROUP</td>
<td>A numeric value indicating the status of group processing:</td>
</tr>
<tr>
<td>Column</td>
<td>Setting</td>
</tr>
<tr>
<td>0</td>
<td>One job step is being generated for each DBDS and/or area. Otherwise a single job is being generated to copy multiple DBDSs and/or areas (Image Copy 2, GROUP or DB without DDN, and ONEJOB were specified). This means that the skeletal JCL member will be processed more than once per job step generated.</td>
</tr>
<tr>
<td>1</td>
<td>Processing the only member of the group</td>
</tr>
<tr>
<td>2</td>
<td>Processing the first of multiple members</td>
</tr>
<tr>
<td>3</td>
<td>Processing an intermediate member</td>
</tr>
<tr>
<td>4</td>
<td>Processing the last member</td>
</tr>
<tr>
<td>%SMS1DS</td>
<td>A numeric value indicating whether Sameds was specified:</td>
</tr>
<tr>
<td>Column</td>
<td>Setting</td>
</tr>
<tr>
<td>0</td>
<td>Sameds was not specified</td>
</tr>
<tr>
<td>1</td>
<td>Sameds was specified</td>
</tr>
<tr>
<td>%GROUP</td>
<td>A character value containing the value of the GROUP parameter, or the value of the DBD parameter if DDn was omitted (an implied group)—otherwise null.</td>
</tr>
</tbody>
</table>

Database Recovery Utility- Receive (ICRCVJCL)

Table 35 explains the symbolic keywords recognized by the Database Recovery (Receive) utility.

Table 35. Symbolic Keywords for Database Recovery (Receive) Utility

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%DBNAME</td>
<td>The database name of the DBDS to be covered. %DBNAME is set from the DBD parameter on the GENJCL.RECEIVE command.</td>
</tr>
<tr>
<td>%DBDDN</td>
<td>The dname of the DBDS, %DBDDN is set from the DDN parameter on the GENJCL.RECEIVE command.</td>
</tr>
<tr>
<td>%DBDSN</td>
<td>The data set name of the DBDS, %DBDSN is set from the DBDS record in RECON.</td>
</tr>
<tr>
<td>%DBDSAM</td>
<td>Set to VSAM for a VSAM DBDS. Otherwise, set to null.</td>
</tr>
</tbody>
</table>
### Symbolic Keywords

<table>
<thead>
<tr>
<th><strong>Keyword</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>%DBUSID</strong></td>
<td>The update set identifier for the DBDS.</td>
</tr>
<tr>
<td><strong>%ALLUSID</strong></td>
<td>The update set identifier of the most-recent ALLOC record for the DBDS.</td>
</tr>
<tr>
<td><strong>%DBNAME</strong></td>
<td>The HALDB master name, if this is a DBDS of a HALDB partition. This keyword is set to NULL for non-HALDBs.</td>
</tr>
<tr>
<td><strong>%DSLLGTM</strong></td>
<td>The start time for selecting input log data. If an input change accumulation data set is used, %DSLLGTM is set to the volume stop time of the last-accumulated log volume. Otherwise, the keyword value is set to image copy time.</td>
</tr>
<tr>
<td><strong>%ICDSN</strong></td>
<td>The data set name of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECEIVE command. Otherwise, set from the image copy record for the DBDS.</td>
</tr>
<tr>
<td><strong>%ICUNIT</strong></td>
<td>The unit type of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECEIVE command. Otherwise, set from the image copy record for the DBDS.</td>
</tr>
<tr>
<td><strong>%ICVOLS</strong></td>
<td>The volume serial number list of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECEIVE command. Otherwise set from the image copy record for the DBDS.</td>
</tr>
<tr>
<td><strong>%ICFSEQ</strong></td>
<td>The file sequence number of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECEIVE command. Otherwise, set from the image copy record for the DBDS.</td>
</tr>
<tr>
<td><strong>%ICUSID</strong></td>
<td>The update set identifier for the image copy.</td>
</tr>
<tr>
<td><strong>%CADSNS</strong></td>
<td>The data set name of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.</td>
</tr>
<tr>
<td><strong>%CAUNIT</strong></td>
<td>The unit type of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.</td>
</tr>
<tr>
<td><strong>%CAVOLS</strong></td>
<td>The volume serial number list of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.</td>
</tr>
<tr>
<td><strong>%CAFSEQ</strong></td>
<td>The file sequence number of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.</td>
</tr>
<tr>
<td><strong>%OLDLFRID</strong></td>
<td>The log record sequence number (log record ID) of the first log record in the OLDS.</td>
</tr>
<tr>
<td><strong>%OLDLLRID</strong></td>
<td>The log record sequence number (log record ID) of the last log record in the OLDS. If the OLDS has not been closed, %OLDLLRID is set to null.</td>
</tr>
<tr>
<td><strong>%SLDFLRID</strong></td>
<td>The log record sequence number (log record ID) of the first log record in the SLDS.</td>
</tr>
<tr>
<td><strong>%SLDFSEQ</strong></td>
<td>The file sequence number of the SLDS.</td>
</tr>
<tr>
<td><strong>%SLDLRID</strong></td>
<td>The log record sequence number (log record ID) of the last log record in the SLDS. If the SLDS has not been closed, %SLDLRID is set to null.</td>
</tr>
<tr>
<td><strong>%SLDREMOT</strong></td>
<td>Set to YES if the SLDS data was created by an active IMS subsystem at a tracking site. That is, the SLDS was received and written locally by the log router. %SLDREMOT is set to null if the SLDS was created locally by an active IMS subsystem.</td>
</tr>
<tr>
<td><strong>%SLDUNIT</strong></td>
<td>Set to null if the SLDS data was created by an active IMS subsystem at a tracking site. SLDSs received from an active site are always cataloged.</td>
</tr>
<tr>
<td><strong>%SLDVOLS</strong></td>
<td>Set to null if the SLDS data was created by an active IMS subsystem at a tracking site. SLDSs received from an active site are always cataloged.</td>
</tr>
<tr>
<td><strong>%LOGDSN</strong></td>
<td>The data set name of the log data set.</td>
</tr>
<tr>
<td><strong>%LOGUNIT</strong></td>
<td>The unit type of the log data set. Set to null if the RLDS data was created by an active IMS subsystem at a tracking site. RLDSs received from an active site are always cataloged.</td>
</tr>
<tr>
<td><strong>%LOGVSEQ</strong></td>
<td>The volume sequence number of the log data set.</td>
</tr>
<tr>
<td><strong>%LOGVOLS</strong></td>
<td>The volume serial numbers of the log data set. Set to null if the RLDS data was created by an active IMS subsystem at a tracking site. RLDSs received from an active site are always cataloged.</td>
</tr>
</tbody>
</table>
Symbolic Keywords

Table 35. Symbolic Keywords for Database Recovery (Receive) Utility (continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%LOGFSEQ</td>
<td>The file sequence number of the log data set.</td>
</tr>
<tr>
<td>%LOGSEL</td>
<td>Set to YES if any log data sets are selected by the select group; in this case, the delete group following the select group is deleted. Otherwise, the %LOGSEL keyword is set to NO, and a DD dummy statement is generated.</td>
</tr>
<tr>
<td>%LOGFLRID</td>
<td>The log record sequence number (log record ID) of the first log record in the RLDS.</td>
</tr>
<tr>
<td>%LOGLLRID</td>
<td>The log record sequence number (log record ID) of the last log record in the RLDS. If the RLDS has not been closed, %LOGLLRID is set to null.</td>
</tr>
<tr>
<td>%LOGREMO</td>
<td>Set to YES if the RLDS data was created by an active IMS subsystem at a tracking site. That is, the RLDS was received and written locally by the log router. %LOGREMO is set to null if the RLDS was created locally by an active IMS subsystem.</td>
</tr>
<tr>
<td>%RVSYSIN</td>
<td>The Database Recovery utility control statement. Columns in the statement are set as follows:</td>
</tr>
<tr>
<td>Column</td>
<td>Setting</td>
</tr>
<tr>
<td>1</td>
<td>$</td>
</tr>
<tr>
<td>4-11</td>
<td>Database name</td>
</tr>
<tr>
<td>13-20</td>
<td>Data set or area ddname</td>
</tr>
<tr>
<td>22-29</td>
<td>DFSUDDUMP</td>
</tr>
<tr>
<td>All other columns are set to blanks.</td>
<td></td>
</tr>
</tbody>
</table>

Database Recovery Utility-Recovery (RECOVJCL)

Table 36 explains the symbolic keywords recognized by the Database Recovery (Receive) utility.

Table 36. Symbolic Keywords for Database Recovery Utility - Recovery (RECOVJCL)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%DBNAME</td>
<td>The database name of the DBDS to be recovered. %DBNAME is set from the DBD parameter on the GENJCL.RECOV command.</td>
</tr>
<tr>
<td>%DBDIN</td>
<td>The ddname of the DBDS; %DBDIN is set from the DDN parameter on the GENJCL.RECOV command.</td>
</tr>
<tr>
<td>%DBDSN</td>
<td>The data set name of the DBDS; %DBDSN is set from the DBDS record of the DBDSs.</td>
</tr>
<tr>
<td>%DBDSAM</td>
<td>Set to VSAM for a VSAM DBDS. Otherwise, set to null.</td>
</tr>
<tr>
<td>%MDBNAME</td>
<td>The HALDB master name, if this is a DBDS of a HALDB partition. This keyword is set to NULL for non-HALDBs.</td>
</tr>
<tr>
<td>%DSLLGTM</td>
<td>The start time for selecting input log data. If an input change accumulation data set is used, %DSLLGTM is set to the volume stop time of the last-accumulated log volume. Otherwise, the keyword value is set to image-copy time.</td>
</tr>
<tr>
<td>%SMS</td>
<td>Indicates whether or not an Image Copy 2 image copy data set is being used for the requested utility execution. Set to 1 if yes; otherwise, set to 0.</td>
</tr>
<tr>
<td>%ICDSN</td>
<td>The data set name of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECOV command. Otherwise, set from the image copy record for the DBDS.</td>
</tr>
<tr>
<td>%ICUNIT</td>
<td>The unit type of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECOV command. Otherwise, set from the image copy record for the DBDS.</td>
</tr>
<tr>
<td>%ICVOLS</td>
<td>The volume serial number list of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECOV command. Otherwise, set from the image copy record for the DBDS.</td>
</tr>
<tr>
<td>%ICFSEQ</td>
<td>The file sequence number of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECOV command. Otherwise, set from the image copy record for the DBDS.</td>
</tr>
</tbody>
</table>
Symbolic Keywords

Table 36. Symbolic Keywords for Database Recovery Utility - Recovery (RECOVJCL) (continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%CADSN</td>
<td>The data set name of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.</td>
</tr>
<tr>
<td>%CAUNIT</td>
<td>The unit type of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.</td>
</tr>
<tr>
<td>%CAVOLS</td>
<td>The volume serial number list of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.</td>
</tr>
<tr>
<td>%CAFSEQ</td>
<td>The file sequence number of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.</td>
</tr>
<tr>
<td>%LOGDSN</td>
<td>The data set name of the log data set.</td>
</tr>
<tr>
<td>%LOGUNIT</td>
<td>The unit type of the log data set.</td>
</tr>
<tr>
<td>%LOGVSEQ</td>
<td>The volume sequence number of the log data set.</td>
</tr>
<tr>
<td>%LOGVOLS</td>
<td>The volume serial numbers of the log data set.</td>
</tr>
<tr>
<td>%LOGFSEQ</td>
<td>The file sequence number of the log data set.</td>
</tr>
<tr>
<td>%LOGSEL</td>
<td>Set to YES if any log data sets are selected by the select group; in this case, the delete group following the select group is deleted. Otherwise, the %LOGSEL keyword is set to NO, and a DD DUMMY statement is generated.</td>
</tr>
<tr>
<td>%RCVFULL</td>
<td>Indicates whether full recoveries are to be generated. When set to YES, full recoveries are generated. If the RCVTIME parameter was specified on the GENJCL.RECOV command, %RCVFULL is set to NO.</td>
</tr>
<tr>
<td></td>
<td>The Database Recovery utility control statement. Columns in the statement are set as follows:</td>
</tr>
<tr>
<td><strong>Column</strong></td>
<td><strong>Setting</strong></td>
</tr>
<tr>
<td>1</td>
<td>$</td>
</tr>
<tr>
<td>4-11</td>
<td>Database name</td>
</tr>
<tr>
<td>13-20</td>
<td>Data set ddname</td>
</tr>
<tr>
<td>31-42</td>
<td>The specified time stamp if the RCVTIME parameter was specified on the GENJCL.RECOV command. Otherwise, blank.</td>
</tr>
<tr>
<td>44</td>
<td>C, if the USEDBDS parameter was specified on the GENJCL.RECOV command. Otherwise, blank.</td>
</tr>
<tr>
<td>All other columns are set to blanks.</td>
<td></td>
</tr>
</tbody>
</table>

IBM-Supplied Skeletal JCL Execution Members

This section lists and describes each of the skeletal JCL execution members that are provided by IBM. This skeletal JCL generates executable JCL for running the applicable utilities:

- “The JOB Statement” on page 440
- “Log Archive Utility JCL (ARCHJCL)” on page 440
- “Database Change Accumulation Utility JCL (CAJCL)” on page 446
- “Log Recovery Utility JCL (LOGCLJCL)” on page 449
- “Database Image Copy Utility JCL (ICJCL)” on page 451
- “Online Database Image Copy Utility JCL (OICJCL)” on page 457
- “Database Recovery Utility JCL-Image Copy Receive-Tracking Site (ICRCVJCL)” on page 459
- “Database Recovery Utility JCL (RECOVJCL)” on page 461
IBM-Supplied Skeletal JCL

- “Sample JCL for HALDB INDEX/ILDS Rebuild Utility (DSPUPJCL)” on page 466

**Related Reading:** Instructions on what you must do before using the skeletal JCL execution members are in “Using IBM-Supplied Skeletal JCL” on page 84

### The JOB Statement

The IBM-supplied skeletal JCL execution member for the JOB statement is named JOBJCL. JOBJCL is invoked when any GENJCL command is issued.

JOBJCL consists of a single statement, as follows:

```
//JT%TIME JOB
```

You need to modify JOBJCL to add job accounting information that is required by your installation. In addition, you can add J0BLIB, STEPLIB, and JES control statements to JOBJCL. The default job name can be modified. If you use this supplied JOB statement, the job name is generated as JThhmmss, where hhmmss is the time (hour, minute, second) that the JCL was generated.

### Log Archive Utility JCL (ARCHJCL)

The IBM-supplied skeletal JCL execution member for the Log Archive utility is named ARCHJCL. ARCHJCL is used when the GENJCL.ARCHIVE command is issued.

Figure 58 on page 441 is a listing of ARCHJCL. A description of the statements in ARCHJCL follows. Figure 58 on page 441
Note: The following is the OLDS archive EXEC statement.
%DELETE (%ARCSLDS EQ 'YES')
//AR%STPNO EXEC PGM=DFSUARC0,PARM='SSID'
%ENDDEL

Note: The following is the SLDS archive EXEC statement.
%DELETE (%ARCSLDS EQ 'NO')
//AR%STPNO EXEC PGM=DFSUARC0,PARM='DBRC=Y'
%ENDDEL

Note: The following lines are used to archive OLDS.
%DELETE (%ARCSLDS EQ 'YES')
%SELECT OLDS(SSID,(ddnames))
//@OLDSDSN DD DSN=%OLDSDSN,DISP=SHR
%ENDSEL

//DFSSLOGP DD DSN=IMS.SLDSP.%SSID.%ARDATE.%ARTIME.%ARVERS,
//    UNIT=3400,Vol=,,99,
//    DISP=(NEW,KEEP),LABEL=(1,SL)
//DFSSLOGS DD DSN=IMS.SLDSS.%SSID.%ARDATE.%ARTIME.%ARVERS,
//    UNIT=3400,Vol=,,99,
//    DISP=(NEW,KEEP),LABEL=(1,SL)
//RLDSDD1 DD DSN=IMS.RLDSP.%SSID.%ARDATE.%ARTIME.%ARVERS,
//    UNIT=3400,Vol=,,99,
//    DISP=(NEW,KEEP),LABEL=(1,SL)
//RLDSDD2 DD DSN=IMS.RLDSS.%SSID.%ARDATE.%ARTIME.%ARVERS,
//    UNIT=3400,Vol=,,99,
//    DISP=(NEW,KEEP),LABEL=(1,SL)
%ENDDEL

Figure 58. IBM-Supplied Skeletal JCL for the Log Archive Utility (Part 1 of 3)
Note: The following lines are used to archive primary SLDSs.

```
%DELETE (%ARCSLDS EQ 'NO')
%SELECT SLDS(%SSID,ALL)
//DFSSLDSP DD DSN=%SLDSdsn,DISP=(OLD,PASS)
%ENDSEL
%ENDDEL
%DELETE (%ARCSLDS EQ 'NO' | %SLDSSEL EQ 'NO')
//DFSSLOGP DD DSN=IMSVS.ARCH1.%SSID.DATE.T%ARTIME,
// UNIT=3400,VOL=(,,99),
// DISP=(NEW,PASS),LABEL=(1,SL)
//RLDSDD1 DD DSN=IMSVS.RLDS1.%SSID.DATE.T%ARTIME,
// UNIT=3400,VOL=(,,99),
// DISP=(NEW,PASS),LABEL=(1,SL)
%ENDDEL
```

Note: The following lines are used to archive secondary SLDSs.

```
%DELETE (%ARCSLDS EQ 'NO')
%SELECT SSLDS(%SSID,ALL)
//DFSSLDSS DD DSN=%SLDSdsn,DISP=(OLD,PASS)
%ENDSEL
%ENDDEL
%DELETE (%ARCSLDS EQ 'NO' | %SLDSSEL EQ 'NO')
//DFSSLOGS DD DSN=IMSVS.ARCH2.%SSID.DATE.T%ARTIME,
// UNIT=3400,VOL=(,,99),
// DISP=(NEW,PASS),LABEL=(1,SL)
//RLDSDD2 DD DSN=IMSVS.RLDS2.%SSID.DATE.T%ARTIME,
// UNIT=3400,VOL=(,,99),
// DISP=(NEW,PASS),LABEL=(1,SL)
%ENDDEL
```

Note: The following lines are common to both processes.

```
//SYSIN DD *
SLDS FEOV(08000)
COPY DONOUT1(RLDSDD1) DONOUT2(RLDSDD2) DBRECOV
/*
```

Figure 58. IBM-Supplied Skeletal JCL for the Log Archive Utility (Part 2 of 3)
You can modify this JCL to suit your needs. It is important to maintain the position of the output DD statements (DFSSLOGP and RLDSDD1) or (DFSSLOGS and RLDSDD2) with respect to the correct %SELECT group. So, the DD statements for the primary output data sets (DFSSLOGP and RLDSDD1) must follow the %SELECT SLDS(%SSID,ALL) select group and precede the %SELECT SSLDS(%SSID,ALL) select group.

Restrictions:

- The %ARVERS keyword is not supported for the SLDS archive process and must not be used.
- Do not use the FREE=CLOSE JCL parameter on the DFSSLOGP DD statement. The data set is dynamically deallocated and using FREE=CLOSE will produce unpredictable results.

EXEC Statement

The %STPNO keyword is replaced with the current step number; then the current step number is increased by 1. The %SSID keyword is replaced with the ID of the IMS subsystem that created the OLDSs.

The DD Statements

STEPLIB DD Statement
DBRC makes no changes to this statement.

SYSPRINT DD Statement
DBRC makes no changes to this statement.

RECONn DD Statements
The RECON DD statements identify the RECONs.
Each of these statements is within a delete group that is controlled by a
%RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names
that are used when the GENJCL command is executed.

- If RECONs are allocated dynamically, the %RCNDSN keywords are set to null,
  and the RECONn DD statements are deleted.
- If RECONs are allocated with JCL, the %RCNDSN keywords are set to the name
  of the corresponding RECON in the GENJCL command.
- If a RECON is not used when the GENJCL command is executed (for
  example, no spare RECON exists), the keyword is set to null, and the DD
  statement is deleted.

OLDS DD Statements
The DD statements for the OLDSs that are to be archived are generated with a
select group. The %SSID keyword identifies the subsystem ID. The %DDNAMES
keyword identifies the OLDSs. A DD statement is generated for each specified
OLDS. The OLDS ddname replaces the %OLDSDDN keyword. The data set name
replaces the %OLDSDDSN keyword.

DFSSLOGP DD Statement
This DD statement defines the primary SLDS to be created. The subsystem ID
replaces the %SSID keyword. The %ARDATE and %ARTIME keywords are replaced
with the date (yyddd) and time (hhmmsst) from the open time stamp of
the oldest OLDS being archived. The %ARVERS keyword is replaced with the archive
version number (nn) of the oldest OLDS being archived.

If you are not using dual SLDS logging, delete this DD statement from the
skeletal JCL execution member.

DFSSLOGS DD Statement
This DD statement defines the secondary SLDS that is to be created. The
subsystem ID replaces the %SSID keyword. The %ARDATE and %ARTIME keywords are
replaced with the date (yyddd) and time (hhmmsst) from the open time
stamp of the oldest OLDS that is being archived. The %ARVERS keyword is
replaced with the archive version number (nn) of the oldest OLDS that is being
archived.

If you are not using an RLDS, delete this statement and the RLDSDD2 DD
statement from the execution member. If this statement is deleted, the utility
control COPY statement must be deleted from the SYSIN data. DBRC does not
verify that the SYSIN data matches the DD statements.

RLDSDD1 DD Statement
This DD statement defines the primary RLDS that is to be created. The
subsystem ID replaces the %SSID keyword. The %ARDATE and %ARTIME keywords are
replaced with the date (yyddd) and time (hhmmsst) from the open time
stamp of the oldest OLDS that is being archived. The %ARVERS keyword is
replaced with the archive version number (nn) of the oldest OLDS that is being
archived.

If you are not using RLDS logging, delete this statement from the execution
member. If this statement is deleted, the DDOUT2(RLDSDD2) parameter must be
deleted from the utility control COPY statement in the SYSIN data. DBRC does not verify that the SYSIN data matches the DD statements.

SYSIN DD Statement
DBRC makes no changes to the SYSIN DD statement or to the utility control statements in the SYSIN data.

DFSSLDSP DD Statements
The DD statements for the primary SLDSs that are to be archived are generated with a select group. The %SSID keyword identifies the subsystem ID. A DD statement is generated for each unarchived SLDS. The SLDS data set name replaces the %SLDS0SN keyword.

DFSSLDSS DD Statements
The DD statements for the secondary SLDSs that are to be archived are generated with a select group. The %SSID keyword identifies the subsystem ID. A DD statement is generated for each unarchived SLDS. The SLDS name replaces the %SLDS0SN keyword.

DFSSLOGP DD Statement
This DD statement defines the primary SLDS that is to be created. The subsystem ID replaces the %SSID keyword. The %ARDATE and %ARTIME keywords are replaced with the date (yyddd) and time (hmmmss) from the open time stamp of the oldest OLDS or SLDS that is being archived. The %ARVERS keyword is replaced with the archive version number (nn) of the oldest OLDS being archived.

If you are not using dual SLDS logging, delete these DD statements and the DD2 DD statements from the skeletal JCL execution member.

RLDSDD1 DD Statement
This DD statement defines the primary RLDS that is to be created. The subsystem ID replaces the %SSID keyword. The %ARDATE and %ARTIME keywords are replaced with the date (yyddd) and time (hmmmss) from the open time stamp of the oldest OLDS or SLDS being archived. The %ARVERS keyword is replaced with the archive version number (nn) of the oldest OLDS being archived.

If you are not using an RLDS, delete these statements, the RLDSDD2 DD statements, and the DD3 and DD4 DD statements from the execution member. If these statements are deleted, the utility control COPY statement must be deleted from the SYSIN data. DBRC does not verify that the SYSIN data matches the DD statements.

RLDSDD2 DD Statement
This DD statement defines the secondary RLDS that is to be created. The subsystem ID replaces the %SSID keyword. The %ARDATE and %ARTIME keywords are replaced with the date (yyddd) and time (hmmmss) from the open time stamp of the oldest OLDS or SLDS that is being archived. The %ARVERS keyword is replaced with the archive version number (nn) of the oldest OLDS that is being archived.
If you are not using dual logging, delete these statements and the DD4 DD statements from the execution member. If these statements are deleted, the DDNOUT2(RLDSD2) parameter must be deleted from the utility control COPY statement in the SYSIN data. DBRC does not verify that the SYSIN data matches the DD statements.

**Database Change Accumulation Utility JCL (CAJCL)**

The IBM-supplied skeletal JCL execution member for the Database Change Accumulation utility is named CAJCL. CAJCL is used when the GENJCL.CA command is issued. You can specify an execution member other than CAJCL by using the CAJCL parameter on the INIT.CAGRP or CHANGE.CAGRP commands.

Figure 59 on page 447 is a listing of CAJCL. A description of the statements in CAJCL follows Figure 59 on page 447.
EXEC Statement

The %STPNO keyword is replaced with the current step number; then the current step number is increased by 1.

The DD Statements

STEPLIB DD Statement

DBRC makes no changes to this statement.

Figure 59. IBM-Supplied Skeletal JCL for the Database Change Accumulation Utility
SYSPRINT DD Statement
DBRC makes no changes to this statement.

RECONn DD Statements
The RECON DD statements identify the RECONs.

Each of these statements is within a delete group that is controlled by a
%RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names
that are used when the GENJCL command is executed.
- If RECONs are allocated dynamically, the %RCNDSN keywords are set to null,
  and the RECONn DD statements are deleted.
- If RECONs are allocated with JCL, the %RCNDSN keywords are set to the name
  of the corresponding RECON in the GENJCL command.
- If a RECON is not used when the GENJCL command is executed (for example, no spare RECON exists), the keyword is set to null, and the DD
  statement is deleted.

IMS DD Statement
DBRC makes no changes to this statement.

SYSOUT DD Statement
DBRC makes no changes to this statement.

SORTLIB DD Statement
DBRC makes no changes to this statement.

SORTWKn DD Statements
DBRC makes no changes to these statements.

DFSUCUMO DD Statement
This statement identifies a previously created change accumulation data set
that is used as input.

Two delete groups are used to generate this DD statement. If no existing
change accumulation data set is defined in RECON for the CA group, the
value of the %CAODSN keyword is null. Thus, the first delete group is deleted,
and the DFSUCUMO DD statement is generated as DUMMY,DCB=BLKSIZE=100.

If an input change accumulation data set is defined in RECON, the %CAODSN
keyword is set to the data set name. Thus, the second delete group is deleted,
and the DFSUCUMO DD statement identifies the input data set. Other keywords
relating to the output data set are replaced as follows:

%CAODSN Data set name
%CAOUNIT Unit type
%CAOVOLS Volume serial number list
%CAOFSEQ File sequence number

DFSUCUMN DD Statement
This DD statement identifies the output change accumulation data set. Other
keywords relating to the output data set are replaced as follows:

%CANDSN Data set name
%CANUNIT Unit type
%CANVCNT Volume count
%CANVOLS Volume serial numbers
%CANFSEQ File sequence number
DFSULOG DD Statement
This DD statement identifies the IMS log data sets that are to be used as input to the Database Change Accumulation utility. A select group selects the required log data sets. %CAGRP identifies the CA group for which log data sets are to be selected. All log volumes that are not previously processed for the CA group are selected. Other keywords for the selected data sets are replaced as follows:

%LOGDSN Data set name
%LOGUNIT Unit type
%LOGVSEQ Volume sequence number
%LOGVOLS Volume serial numbers
%LOGFSEQ File sequence numbers

If any log data sets are selected by the select group, the value of the %LOGSEL keyword in the next delete group is YES; this causes the delete group to be deleted. Otherwise, the %LOGSEL keyword is set to NO, and a DD DUMMY statement is generated.

DFSUDD1 DD Statement
DBRC makes no changes to this statement.
The DFSUDD1 DD statement identifies the optional output log data set that is produced by the Database Change Accumulation utility. DBRC does not record the optional output log data set; therefore, the skeletal JCL execution member specifies the DFSUDD1 DD statement as DUMMY.

SYSIN DD Statement
DBRC makes no changes to this statement.

DB0 Control Statements
A DB0 control statement is generated for each DBDS in the CA group.

For a detailed description of the DB0 control statement, see “DB0 Statement” under “Database Change Accumulation Utility (DFSUCUM0)” in the "Utilities Reference: Database and Transaction Manager".

Log Recovery Utility JCL (LOGCLJCL)
The IBM-supplied skeletal JCL execution member for the Log Recovery utility is named LOGCLJCL. The LOGCLJCL member is used to generate the JCL necessary that allows the Log Recovery utility (DFSULTR0) to close an OLDS from the WADS. For all other uses of the Log Recovery utility, see the "IMS V9: Utilities Reference: System"

LOGCLJCL is used when the GENJCL.CLOSE command is issued.

"Figure 60 on page 450" is a listing of LOGCLJCL. A description of the statements in LOGCLJCL follows "Figure 60 on page 450".
EXEC Statement

The %STPNO keyword is replaced with the current step number; then the current step number is increased by 1. The %SSID keyword is replaced with the ID of the IMS subsystem that created the OLDS that is to be closed.

STEPLIB DD Statement

DBRC makes no changes to this statement.

SYSPRINT DD Statement

DBRC makes no changes to this statement.

RECONn DD Statements

The RECON DD statements identify the RECONs.

Each of these statements is within a delete group that is controlled by a %RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names that are used when the GENJCL command is executed.

- If RECONs are allocated dynamically, the %RCNDSN keywords are set to null, and the RECONn DD statements are deleted.
- If RECONs are allocated with JCL, the %RCNDSN keywords are set to the name of the corresponding RECON in the GENJCL command.

---

Figure 60. IBM-Supplied Skeletal JCL for the Log Recovery Utility

EXEC Statement

The %STPNO keyword is replaced with the current step number; then the current step number is increased by 1. The %SSID keyword is replaced with the ID of the IMS subsystem that created the OLDS that is to be closed.

STEPLIB DD Statement

DBRC makes no changes to this statement.

SYSPRINT DD Statement

DBRC makes no changes to this statement.

RECONn DD Statements

The RECON DD statements identify the RECONs.

Each of these statements is within a delete group that is controlled by a %RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names that are used when the GENJCL command is executed.

- If RECONs are allocated dynamically, the %RCNDSN keywords are set to null, and the RECONn DD statements are deleted.
- If RECONs are allocated with JCL, the %RCNDSN keywords are set to the name of the corresponding RECON in the GENJCL command.
• If a RECON is not used when the GENJCL command is executed (for example, no spare RECON exists), the keyword is set to null, and the DD statement is deleted.

**OLDS DD Statement**
This DD statement identifies the OLDS that is to be closed. A select group is used to select the OLDS. The %SSID keyword identifies the subsystem ID, and the %DDNAME identifies the OLDS by its DD name. The OLDS type, primary or secondary, replaces the %OLDSTYP keyword. The resulting ddname is DFSO LP or DFSOLS. The %OLDS0SN keyword is replaced with the data set name of the OLDS.

**WADS DD Statement**
This statement is provided only as a model. You must change it before using the skeletal JCL execution member.

The supplied DFSWADS0 DD statement must be replaced with DD statements DFSWADS0 through DFSWADSn, n+1 is the number of WADSs that the online IMS control region uses. The WADS DD statements are contained in a select group that is controlled by the keyword %WADS. The GENJCL.CLOSE command processor sets the value of the %WADS keyword to YES if the OLDS is to be closed using the WADS. The command processor sets the value to NO if the OLDS is to be closed using the next OLDS. The WADS DD statements are, therefore, deleted if the OLDS is to be closed using the next OLDS.

**Next OLDS DD Statements**
If the OLDS is to be closed using the next OLDS, these DD statements identify the next OLDSs. These statements are contained in a delete group that is controlled by the %WADS keyword. Thus, if the OLDS is to be closed using the WADS, these statements are deleted. A select group is used in order to select the next OLDSs. The %SSID keyword identifies the subsystem ID. The %DDNAME keyword identifies the next OLDS by ddname. The OLDS type, primary or secondary, replaces the %OLDSTYP keyword. The resulting ddname is DFSNOLP or DFSNOLS. The %OLDS0SN keyword is replaced with the data set name of the OLDS.

**Prior OLDS DD Statements**
If an immediately prior OLDS exists, the corresponding DD statement identifies the immediately prior OLDS. These statements are contained in a delete group that is controlled by the %DDNAME keyword. If its value is not null, a select group is used in order to select the immediately prior OLDS. The resulting ddname is DFSPOLP or DFSPOLLS. Processing of the other keywords is as described under Next OLDS DD Statements, above.

**SYSIN DD Statement**
DBRC makes no changes to the SYSIN DD statement or to the utility control statements in the SYSIN data.

**Database Image Copy Utility JCL (ICJCL)**
The IBM-supplied skeletal JCL execution member for the Database Image Copy and Database Image Copy 2 utility is named ICJCL.

ICJCL is used when the GENJCL.IC command is issued. The ICJCL may be processed more than once for each job step in the output JCL. You can specify an execution member other than ICJCL by using the ICJCL parameter on the INIT.DBDS or CHANGE.DBDS commands.

The JCL is divided into three parts:
IBM-Supplied Skeletal JCL

First part (from the EXEC statement through the IMS DD statement)
Contained in delete groups controlled by keyword %SMSGRP so that they are processed only once per job.

Second part (from the %DBADDN DD statement through the %ICDDN4 DD statement)
Processed once for each DBDS or area to be copied in a job step. The second set of %ICDDNn DD statements are deleted if Sameds was specified on the GENJCL command.

Third part (SYSIN DD statement and the %ICSYSIN statement)
Contained in a delete group controlled by keyword %SMSGRP so that they are processed only for the last (or only) DBDS or area to be copied in a job step.

Figure 61 on page 453 is a listing of ICJCL. A description of the statements in ICJCL follows.
%DELETE (%SMGRP NE ')
//------- These comments are useful for modifying the skeletal member, but are confusing when generated in the output JCL. Force them to be deleted.
//-------
//------- New variables for GENJCL.IC:
//------- One DBDS per step:
//-------  %SMGRP  =  0 - Not SMS or not Group
//-------       %SMSIDS  =  0 - Not SMS or not Group or not ONEDS
//------- Multiple DBDSs per step:
//-------  %SMGRP  =  1 - SMS + Group + only member
//-------       %SMGRP  =  2 - SMS + Group + first member
//-------       %SMGRP  =  3 - SMS + Group + not first or last
//-------       %SMGRP  =  4 - SMS + Group + last member
//-------       %SMSIDS  =  1 - %SMGRP > 0 + ONEDS
//-----------------------------------------------------------------
//-------| Section 1 is used only once per step
//-----------------------------------------------------------------
%ENDDEL
%DELETE (%SMGRP GT '2')
//IC%STPNO EXEC PGM=%PGMIC,REGION=800K,
//      PARM='%PARMX'
//*
//*
// THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
//*
// KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
//*
// THE IMS/ESA DATA BASE RECOVERY CONTROL FEATURE.
//*
//*
// JCL FOR IMAGE COPY.
//*
//*/
//STPLIB DD DSN=IMSVS.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
%ENDDEL
%DELETE (%RCNDSN1 EQ '' | %SMGRP GT '2')
//RECON1 DD DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE (%RCNDSN2 EQ '' | %SMGRP GT '2')
//RECON2 DD DSN=%RCNDSN2,DISP=SHR
%ENDDEL
%DELETE (%RCNDSN3 EQ '' | %SMGRP GT '2')
//RECON3 DD DSN=%RCNDSN3,DISP=SHR
%ENDDEL
%DELETE (%SMGRP GT '2')
//IMS DD DSN=IMSVS.DBDLIB,DISP=SHR
%ENDDEL
%DELETE (%SMGRP NE '')

Figure 61. IBM-Supplied Skeletal JCL for the Database Image Copy Utility and the Database Image Copy 2 Utility (Part 1 of 3)
Figure 61. IBM-Supplied Skeletal JCL for the Database Image Copy Utility and the Database Image Copy 2 Utility (Part 2 of 3)
This member may be processed more than once for each job step in the output JCL.

All statements in the first part of the JCL, from the EXEC statement through the IMS DD statement, are in delete groups controlled by keyword %SMSGRP so that they are processed only once per job step.

The first set of %ICDDn DD statements identify the single output data set for a group of image copies. It is in a delete group controlled by the %SMSIDS keyword so that it is output only when SameDS was specified in the GENJCL command.

The second part of the JCL, from the %DBADDn DD statement through the %ICDDn4 DD statement, is processed once for each DBDS or area to be copied in a job step. The %ICDDn DD statements are deleted if SameDS was specified on the GENJCL command.
IBM-Supplied Skeletal JCL

- The third part of the JCL, the SYSIN DD statement and the %ICSYSIN statement, are in a delete group controlled by keyword %SMSGRP so that they are processed only for the last (or only) DBDS or area to be copied in a job step.

**EXEC Statement**
The %STPN0 keyword is replaced with the current step number; then the current step number is increased by 1.

**STEPLIB DD Statement**
DBRC makes no changes to this statement.

**SYSPRINT DD Statement**
DBRC makes no changes to this statement.

**RECONn DD Statements**
The RECON DD statements identify the RECONs.

Each of these statements is within a delete group that is controlled by a %RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names that are used when the GENJCL command is executed.

- If RECONs are allocated dynamically, the %RCNDSN keywords are set to null, and the RECONn DD statements are deleted.
- If RECONs are allocated with JCL, the %RCNDSN keywords are set to the name of the corresponding RECON in the GENJCL command.
- If a RECON is not used when the GENJCL command is executed (for example, no spare RECON exists), the keyword is set to null, and the DD statement is deleted.

**IMS DD Statement**
DBRC makes no changes to this statement.

**%DBADDN DD Statement**
This statement identifies the available ADS that is to be used. The %DBADDN keyword is replaced with the ddname of the ADS. The %DBDSN keyword is replaced with the ADS name.

**%DBDNN DD Statement**
This statement identifies the DBDS that is to be copied. The %DBDNN keyword is replaced with the ddname of the DBDS. The %DBDSN keyword is replaced by the data set name of the DBDS.

**%ICDDN1 DD Statement**
These statements identify the first image copy data set that is produced by an Image Copy utility. Other keywords relating to the image copy data set are replaced as follows:

- %ICDSN1 Data set name
- %ICVCNT1 Volume count
- %ICVOLS1 Volume serial number list
- %ICUNIT1 Unit type
- %ICFSEQ1 File sequence number

**%ICDDN2 | 3 | 4 DD Statement**
These statements identify the subsequent images that are produced by the Image Copy utility. This DD statement is within a delete group that is controlled by the %COPIES keyword. The %COPIES keyword is set to 1 if a single image copy data set is to be produced or to a 2, 3, or 4 if multiple image copy data sets are to be produced. If %COPIES is 1, the group is deleted.
The %ICDNx, %ICVCNTx, %ICVOLSx, %ICUNITx, and %ICFSEQx keywords are replaced with the same type of information as is shown under the %D1OU0 DD statement just preceding. x can be either 2, 3 or 4.

**SYSIN DD Statement**

DBRC makes no changes to this statement.

**%ICSYSIN Statement**

The Image Copy utility control statement replaces the %ICSYSIN keyword.

The %ICSYSIN statement is required. If the %ICSYSIN statement is deleted, the GENJCL.IC command fails. Multiple control statements are generated if you are processing a group.

### Online Database Image Copy Utility JCL (OICJCL)

The IBM-supplied skeletal JCL execution member for the Online Image Copy utility is named OICJCL. OICJCL is used when the GENJCL.IC command is issued. You can specify an execution member other than OICJCL by using the OICJCL parameter on the INIT.DBDs or CHANGE.DBDs commands.

Figure 62 is a listing of OICJCL. A description of the statements in OICJCL follows.

```
//OIC%STPNO EXEC PGM=DFSRRRC00,PARM='BMP,DFSUICP0,%PSB,,MASTER',
// REGION=700K
/* THIS JCL ORIGIANATES FROM THE USER'S 'JCLPDS' LIBRARY.
/* KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
/* THE IMS/VS DATA BASE RECOVERY CONTROL FEATURE.
/*
/* JCL FOR ONLINE IMAGE COPY.
/*
//STEPLIB DD DSN=IMSVS.DFSRESL,DISP=SHR
//SYSPRINT DD SYSPOUT=A
%DELETE (%RCNDSN1 EQ '')
//RECON1 DD DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE (%RCNDSN2 EQ '')
//RECON2 DD DSN=%RCNDSN2,DISP=SHR
%ENDDEL
%DELETE (%RCNDSN3 EQ '')
//RECON3 DD DSN=%RCNDSN3,DISP=SHR
%ENDDEL
//IMS DD DSN=IMSVS.DBDBLIB,DISP=SHR
//%ICDDN1 DD DSN=%ICDSN1,UNIT=%ICUNIT1,
// VOL=(PRIVATE,%,ICVCNT1,SER=(%ICVOLS1)),
// LABEL=(%ICFSEQ1,SL),
// DISP=(NEW,KEEP)
%DELETE (%COPIES EQ '1')
//%ICDDN2 DD DSN=%ICDSN2,UNIT=%ICUNIT2,
// VOL=(PRIVATE,%,ICVCNT2,SER=(%ICVOLS2)),
// LABEL=(%ICFSEQ2,SL),
// DISP=(NEW,KEEP)
%ENDDEL
//DFSUCKPT DD DSN=IMSVS.%DBNAME.%DBDDN.CHECKPT.IC%TIME,
// UNIT=SYSDA,SPACE=(TRK,1),DISP=(NEW,CATLG)
//SYSIN DD *
%1CSYSIN
/*
```

*Figure 62. IBM-Supplied Skeletal JCL for the Online Database Image Copy Utility*

**EXEC Statement**

The %STPNO keyword is replaced with the current step number; then the current
step number is increased by 1. The PSB name that is specified on the GENJCL.01C command replaces the %PSB keyword.

**STEPLIB DD Statement**
DBRC makes no changes to this statement.

**SYSPRINT DD Statement**
DBRC makes no changes to this statement.

**RECONn DD Statements**
The RECON DD statements identify the RECONs.
Each of these statements is within a delete group that is controlled by a %RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names that are used when the GENJCL command is executed.

- If RECONs are allocated dynamically, the %RCNDSN keywords are set to null, and the RECONn DD statements are deleted.
- If RECONs are allocated with JCL, the %RCNDSN keywords are set to the name of the corresponding RECON in the GENJCL command.
- If a RECON is not used when the GENJCL command is executed (for example, no spare RECON exists), the keyword is set to null, and the DD statement is deleted.

**IMS DD Statement**
DBRC makes no changes to this statement.

**%ICDDN1 DD Statement**
This statement identifies the first image copy data set that is produced by the Image Copy utility. Other keywords relating to the online image copy data set are replaced as follows:

- %ICDSN1: Data set name
- %ICVCNT1: Volume count
- %ICVOLS1: Volume serial number list
- %ICUNIT1: Unit type
- %ICFSEQ1: File sequence number

**%ICDDN2 DD Statement**
This statement identifies the duplicate image copy data set that produced by the Image Copy utility. This DD statement is within a delete group controlled by the %COPIES keyword. The %COPIES keyword is set to 1 if a single image copy data set is to be produced or to 2 if duplicate image copy data sets are to be produced. If %COPIES is 1, the group is deleted.

The %ICDSN2, %ICVCNT2, %ICVOLS2, %ICUNIT2, and %ICFSEQ1 keywords are replaced with the same type of information as is shown under the %D1OUO DD statement just preceding.

**DFSUCKPT DD Statement**
The DFSUCKPT DD statement identifies the optional online image copy checkpoint data set. Keywords relating to this optional data set are replaced as follows:

- %DBNAME: Database name
- %DBDDN: ddname
- %TIME: Current time of day (in the form hhmmss)
The volume serial number and device type for the checkpoint data set are not specified in the IBM-supplied skeletal JCL. You must supply these if checkpoint data sets are to be used.

The DFSU CKPT DD statement is optional. If checkpoint data sets are not to be used by the Online Image Copy utility, the statement can be deleted.

**SYSIN DD Statement**
- DBRC makes no changes to this statement.

**%ICSYSIN Statement**
- The Image Copy utility control statement replaces the %ICSYSIN keyword.

**Database Recovery Utility JCL-Image Copy Receive-Tracking Site (ICRCVJCL)**

The IBM-supplied skeletal JCL execution member for the Database Recovery utility (as used at the tracking site to receive the image copy) is named ICRCVJCL. ICRCVJCL is used when the GENJCL.RECEIVE command is issued. You can specify an execution member other than ICRCVJCL by using the RECVJCL parameter on the INIT.DBDS or CHANGE.DBDS commands.

Figure 63 on page 460 is a listing of ICRCVJCL. A description of the statements in ICRCVJCL follows Figure 63 on page 460.
IBM-Supplied Skeletal JCL

//RCV%STPNO EXEC PGM=DFSRRC00,REGION=1300K,
//      PARM='UDR,DFSURDB0,%DBNAME',.......................,%GSGNAME'
//*
//  THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
//  KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
//  THE IMS/VS DATA BASE RECOVERY CONTROL FEATURE.
//*
//  JCL FOR IMAGE COPY RECEIVE
//*
//STEPLIB DD DSN=IMSVS.RESLIB,DISP=SHR
// SYSPRINT DD SYSOUT=A
%DELETE (%RCNDSN1 EQ '')
//RECON1 DD DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE (%RCNDSN2 EQ '')
//RECON2 DD DSN=%RCNDSN2,DISP=SHR
%ENDDEL
%DELETE (%RCNDSN3 EQ '')
//RECON3 DD DSN=%RCNDSN3,DISP=SHR
%ENDDEL
//IMS DD DSN=IMSVS.DDBLIB,DISP=SHR
//%OBDSN DD DSN=%OBDSN,
%DELETE (%DBDSAM EQ 'VSAM')
// UNIT=SYSDA,
// VOL=SER=VOLSER,
// SPACE=(CYL,(20,2)),
// DISP=(NEW,KEEP),
// DCB=BUFNO=10
%ENDDEL
%DELETE (%DBDSAM NE 'VSAM')
// DISP=OLD
%ENDDEL
//DFSUDUMP DD DSN=%ICDSN,UNIT=%ICUNIT,
// VOL=(PRIVATE,,,SER=(%ICVOLS)),
// LABEL=(%ICFSEQ,SL),
// DISP=(OLD,KEEP),DCB=BUFNO=10
%DELETE (%LOGSEL EQ 'YES')
//DFSULOG DD DUMMY
%ENDDEL
%DELETE (%CADSN NE '')
//DFSUCUM DD DUMMY
%ENDDEL
//DFSVSAMP DD *
1024,4
4096,4
//SYSIN DD *
%RVSYSIN
*/

Figure 63. IBM-Supplied Skeletal JCL for the Database Recovery Utility - Tracking Site

EXEC Statement
The %STPNO keyword is replaced with the current step number; then the current step number is increased by 1. The %DBNAME keyword is replaced with the database name of the DBDS or area that is being received.

The DD Statements

STEPLIB DD Statement
DBRC makes no changes to this statement.

SYSPRINT DD Statement
DBRC makes no changes to this statement.

RECONn DD Statements
The RECON DD statements identify the RECONs.
Each of these statements is within a delete group that is controlled by a %RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names that are used when the GENJCL command is executed.

- If RECONs are allocated dynamically, the %RCNDSN keywords are set to null, and the RECON DD statements are deleted.
- If RECONs are allocated with JCL, the %RCNDSN keywords are set to the name of the corresponding RECON in the GENJCL command.
- If a RECON is not used when the GENJCL command is executed (for example, no spare RECON exists), the keyword is set to null, and the DD statement is deleted.

**IMS DD Statement**
DBRC makes no changes to this statement.

**%DBDDN DD Statement**
The %DBDDN keyword is replaced by the ddname of the DBDS that is being received. The %DBDSN keyword is replaced by the data set name of the DBDS or area.

Delete groups control the remainder of the %DBDDN DD statement. The access method of the DBDS controls the content of the delete groups. If the access method is VSAM, DISP=OLD is generated. Otherwise, the UNIT, VOL, SPACE, DISP, and DCB parameters are generated.

**DFSUDUMP DD Statement**
This DD statement identifies the image copy data set that is to be received. The %ICDSN, %ICUNIT, %ICVOLS, and %ICFSEQ keywords are set from the appropriate fields in the image copy RECON record.

**DFSUCUM DD Statement**
This DD statement is always listed as DUMMY at the tracking site.

**DFSULOG DD Statement**
This DD statement is always listed as DUMMY at the tracking site.

**DFSVSAMDD DD Statement**
The DFSVSAMP DD statement identifies information that is required by the DL/I buffer handler. DBRC makes no changes to these statements.

**SYSIN DD Statement**
This DD statement contains database recovery statements that control the processing.

**%RVSYSIN Statement**
DBRC replaces the %RVSYSIN keyword.

**Database Recovery Utility JCL (RECOVJCL)**
The IBM-supplied skeletal JCL execution member for the Database Recovery utility is named RECOVJCL. RECOVJCL is used when the GENJCL.RECOV command is issued. You can specify an execution member other than RECOVJCL by using the RECOVJCL parameter on the INIT.DBDS or CHANGE.DBDS commands.

[Figure 64 on page 462] is a listing of RECOVJCL. A description of the statements in RECOVJCL follows [Figure 64 on page 462]
IBM-Supplied Skeletal JCL

```jcl
//RCV%STPNO EXEC PGM=DFSRRC00,REGION=1300K, // PARM='UDR,DFSURDB0,%DBNAME,................,%GSGNAME'
%ENDDEL
%DELETE (%MDBNAME NE '')
//RCV%STPNO EXEC PGM=DFSRRC00,REGION=1300K, // PARM='UDR,DFSURDB0,%MDBNAME,................,%GSGNAME'
%ENDDEL
%DELETE (%MDBNAME EQ '')
//RCV%STPNO EXEC PGM=DFSRRC00,REGION=1300K, // PARM='UDR,DFSURDB0,%MDBNAME,................,%GSGNAME'
%ENDDEL

//* THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
//* KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
//* THE IMS/VS DATA BASE RECOVERY CONTROL FEATURE.
//* JCL FOR RECOVERY.

//STPLIB DD DSN=IMSVS.SDFSRESL,DISP=SHR
//SYSPRINT DD SYSOUT=A
%DELETE (%RCNDSN1 EQ '')
//RECON1 DD DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE (%RCNDSN2 EQ '')
//RECON2 DD DSN=%RCNDSN2,DISP=SHR
%ENDDEL
%DELETE (%RCNDSN3 EQ '')
//RECON3 DD DSN=%RCNDSN3,DISP=SHR
%ENDDEL
//IMS DD DSN=IMSVS.DBDLIB,DISP=SHR
/%OBDSN DD DSN=%OBDSN,
%DELETE (%OBDSN EQ 'VSAM' | %SMS EQ '1')
  //UNIT=SYSDA,
  //VOL=SER=VOLSER,
  //SPACE=(CYL,(20,2)),
  //DISP=(NEW,KEEP),
  //DCB=BUFNO=10
%ENDDEL
%DELETE (%OBDSN EQ 'VSAM' | %SMS EQ '0')
  //UNIT=SYSDA,
  //VOL=SER=VOLSER,
  //SPACE=(CYL,(20,2)),
  //DISP=(NEW,KEEP)
%ENDDEL
%DELETE (%OBDSAM NE 'VSAM')
  //DISP=OLD
%ENDDEL
%DELETE (%ICDNS EQ '')
//DFSUDUMP DD DSN=%ICDNS,
%ENDDEL
```

Figure 64. IBM-Supplied Skeletal JCL for the Database Recovery Utility (Part 1 of 2)
EXEC Statement

The %STPNO keyword is replaced with the current step number; then the current step number is increased by 1. For HALDB, the HALDB master name, %MDBNAME, of the DBDs being recovered is used in the EXEC statement. For all others, the database name, %DBNAME, is used.

The DD Statements

STEPLIB DD Statement

DBRC makes no changes to this statement.

SYSPRINT DD Statement

DBRC makes no changes to this statement.
RECONn DD Statements
The RECON DD statements identify the RECONs.

Each of these statements is within a delete group that is controlled by a
%RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names
that are used when the GENJCL command is executed.

- If RECONs are allocated dynamically, the %RCNDSN keywords are set to null,
  and the RECONn DD statements are deleted.
- If RECONs are allocated with JCL, the %RCNDSN keywords are set to the name
  of the corresponding RECON in the GENJCL command.
- If a RECON is not used when the GENJCL command is executed (for
  example, no spare RECON exists), the keyword is set to null, and the DD
  statement is deleted.

IMS DD Statement
DBRC makes no changes to this statement.

%DBDDN DD Statement
The %DBDDN keyword is replaced by the ddname of the DBDS that is being
recovered. The %DBDSN keyword is replaced by the data set name of the DBDS.

Delete groups control the remainder of the %DBDDN DD statement. The access
method of the DBDS controls the content of the delete groups. If the access
method is VSAM, DISP=OLD is generated. Otherwise, the UNIT, VOL, SPACE, DISP,
and DCB parameters are generated.

DFSUDUMP DD Statement
This DD statement identifies the image copy data set, if any, that is to be used
for recovery. Delete groups, which are controlled by the %ICDSN keyword, are
used to generate this DD statement.

If the USEIC parameter was specified or if it was the default, on the
GENJCL.RECOV command, the %ICDSN keyword is set to its data set name. Thus,
the first delete group for DFSUDUMP is used, and the second delete group is
deleted. Other keywords within the first delete group are unchanged.

If the USEDBDS or USEAREA keyword was specified on the GENJCL.RECOV
command, the DFSUDUMP DD statement is generated as DUMMY.

DFSVDUMP DD Statement
The DFSVDUMP DD statement is always generated as DUMMY.

DFSUCUM DD Statement
This DD statement identifies the change accumulation data set, if any, to be
used as input to recovery. Delete groups, which are controlled by the %CADSN
keyword, are used to generate the DFSUCUM DD statement. If the DBDS belongs
to a CA group, the %CADSN keyword is set to the data set name of the
most-recent change accumulation data set. If the DBDS does not belong to a
CA group or if no usable change accumulation data set exists, the %CADSN
keyword is set to null.

- If the %CADSN keyword is null, the DFSUCUM DD statement is generated as
  DUMMY.
- If the %CADSN keyword is not null, the DFSUCUM DD statement identifies the
  change accumulation data set.

Other keywords relating to the change accumulation data set are replaced as
follows:

%CAVOLS Volume serial number list
%CAUNIT Unit type
DFSULOG DD Statement
This DD statement identifies the log data sets that are to be used as input to the Database Recovery utility.

A select group selects the required log data sets. The %DBNAME and %DBDDN keywords identify the DBDS for which log data sets are to be selected. All log volumes that contain change records for the DBDS, that are not included in the change accumulation data sets, are selected. Other keywords relating to the log data sets are replaced as follows:

%LOGDSN Log data set name
%LOGUNIT Unit type
%LOGVSEQ Volume sequence number
%LOGVOLS Volume serial numbers
%LOGFSEQ File sequence number

If any log data sets are selected, the value of the %LOGSEL keyword is YES, and the following delete group is deleted. Otherwise, the %LOGSEL keyword is NO, and a DD DUMMY statement is generated.

DFSTRCV DD Statement
The DFSTRCV DD statement identifies the DBDS for which one or more tracks is being recovered. If the TRACK parameter was not specified on the GENJCL command, this statement does not appear in the generated JCL.

You must modify the DFSTRCV DD statement to include in it the appropriate data set name and unit information. You can modify it in either the skeletal JCL or generated JCL.

DFSVSAMP DD Statement
The DFSVSAMP DD statement identifies information required by the DL/I buffer handler. DBRC makes no changes to these statements.

SYSIN DD Statement
This DD statement contains database recovery statements that control the processing.

%RCSYSIN Statement
DBRC replaces the %RCSYSIN keyword.

%RCVFULL
The %RCVFULL keyword indicates what type of recovery is being generated. It is set to NO when the RCVTIME parameter (time-stamp recovery) is specified on the GENJCL.RECOV command. It is set to YES to indicate full recoveries.

This keyword is useful if, for example, you want to turn ON the IC-NEEDED flag in the DBDS record following a time-stamp recovery. You could add the following JCL to the end of your RECOVJCL skeletal JCL member to accomplish this.

```plaintext
%DELETE (%RCVFULL 'YES')
//RCV%STPNO EXEC PGM=DSPURX00
//STEPLIB DD DSN=IMS.RESLIB,DISP=SHR
%ENDDEL
%DELETE (%RCVFULL 'YES' | %RCNDSN1 '')
//RECON1 DD DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE (%RCVFULL 'YES' | %RCNDSN2 '')
//RECON2 DD DSN=%RCNDSN2,DISP=SHR
```

Appendix A. Symbolic Keywords Recognized by DBRC 465
Sample JCL for HALDB INDEX/ILDS Rebuild Utility (DSPUPJCL)

DFSPREC0 is used to rebuild the ILDS or prime index data sets (or both) of a HALDB partition.

The Database Recovery utility and the image copy utilities cannot be used on ILDS and prime index data sets. The GENJCL.RECOV, GENJCL.RECEIVE, GENJCL.IC, and GENJCL.OIC commands fail when attempted specifically for ILDS and prime index data sets. GENJCL commands on groups, either explicit (using the GROUP keyword) or implicit (DBD with no DDN keyword), do not generate any JCL for ILDS or prime index data sets. These data sets are ignored. GENJCL.USER can specify ILDS and prime index data sets. The ILDS and prime index data sets are not ignored for this command.

After a time-stamp recovery of a HALDB partition (the data DBDSs of the HALDB partition), the applicable ILDS or prime index data sets (or both) must be rebuilt. There is no specific GENJCL support for these data sets, but the GENJCL.USER command can be used. The IBM-supplied skeletal JCL execution member is named DSPUPJCL. Here is a suggestion for implementation:

GENJCL.USER MEMBER (DSPUPJCL) - USERKEYS ((%MDBNAME, 'DBHDOJ01'),(%DBDNAME,'PART1'), - (%RCVTYP, 'ILE'))

Figure 65 on page 467 is a listing of DSPUPJCL. A description of the statements in DSPUPJCL follows Figure 65 on page 467.
EXEC Statement

The `%STPNO` keyword is replaced with the current step number; then the current step number is increased by 1. The `%MDBNAME` keyword is replaced with the HALDB master name of the DBDS that is being recovered.

The DD Statements

SYSPRINT DD Statement

DBRC makes no changes to this statement.

RECONn DD Statements

The RECON DD statements identify the RECONs.

Each of these statements is within a delete group that is controlled by a `%RCNDSN` keyword. The `%RCNDSN` keyword values are set from the RECON names that are used when the GENJCL command is executed.

- If RECONs are allocated dynamically, the `%RCNDSN` keywords are set to null, and the RECONn DD statements are deleted.
- If RECONs are allocated with JCL, the `%RCNDSN` keywords are set to the name of the corresponding RECON in the GENJCL command.
- If a RECON is not used when the GENJCL command is executed (for example, no spare RECON exists), the keyword is set to null, and the DD statement is deleted.

IMS DD Statement

DBRC makes no changes to this statement.

DFSVSAMP DD Statement

The DFSVSAMP DD statement identifies information required by the DL/I buffer handler. DBRC makes no changes to these statements.

SYSGIN DD Statement

This DD statement contains database recovery statements that control the processing.

Figure 65. IBM-Supplied Skeletal JCL for the HALDB Index/ILDS Rebuild Utility

//UPREC%STPNO EXEC PGM=DFSSR00,REGION=1300K,
//                  PARM='ULU,DFSPREC0,%MDBNAME,..................,Y,N'
//*
.TextChanged.RecoveryStatement

//*****************************************************************************
//*****************************************************************************
//*****************************************************************************
//**          THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
//**          KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION.
//**          User JCL for rebuilding the Index and/or ILDS data
//**          sets for a Partition of a HALDB.
//**
//SYSPRINT DD SYSOUT=A
%DELETE (%RCNDSN1 EQ ' ')
//RECON1 DD DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE (%RCNDSN2 EQ ' ')
//RECON2 DD DSN=%RCNDSN2,DISP=SHR
%ENDDEL
%DELETE (%RCNDSN3 EQ ' ')
//RECON3 DD DSN=%RCNDSN3,DISP=SHR
%ENDDEL
//IMS DD DSN=IMS.DBDLIB,DISP=SHR
//DFSVSAMP DD DSN=IMS.VSAM.PARM(OPTIONS),DISP=SHR
//SYSIN DD *
PARTITION=%DBNAME,RECOVTYP=%RCVTYP
/*
PARTITION=%DBNAME
This statement indicates the HALDB partition name of the ILDS or prime index data sets being rebuilt. For a description of the %DBNAME keyword values, see the [IMS V9: Utilities Reference: Database and Transaction Manager]

RECOVTYPE=%RCVTYP
This statement indicates which type of data sets (ILDS or prime index) are being rebuilt. For a description of the %RCVTYP keyword values, see the [IMS V9: Utilities Reference: Database and Transaction Manager]
Appendix B. Sample Listings from the RECON Data Set

The following listings show the format and content of various records from an example RECON data set, as they were listed by LIST.RECON commands.

In This Appendix:

- “Sample Listing of LIST.HISTORY Output” on page 470
- “Sample Listing of a RECON at the Active Site” on page 475
- “Sample Listing of a RECON at the Tracking Site” on page 504
- “Fields Present in a Listing of a RECON by Record Type” on page 528

The LIST command causes the RECON to be read.

The sample listings illustrate:

- These records: PRILOG, PRISLD, PRIOLD, LOGALL, GSG, SSYS, and BACKOUT
- DBDS group records including DBDSs, CAGRP records and DEDB areas
- DB records showing various share levels and authorization states
- DBDS records, including area data sets that support DEDB areas
- Information corresponding to the database activity regarding:
  - Image copy data set information
  - Change accumulation information
  - Reorganization information
  - Recovery information
  - And, depending upon database activity:
    - ALLOC records
    - IMAGE records
    - CA records
    - REORG records
    - RECOV records

To find the DSECTS defining the formats of the RECON records in DBRC in ADFSMAC and SDFSMAC, run the generate job with SDFSMAC=ALL. The member names are:

<table>
<thead>
<tr>
<th>Member Name</th>
<th>RECON Record DSECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSPALLRC</td>
<td>ALLOC record</td>
</tr>
<tr>
<td>DSPBKORC</td>
<td>BACKOUT record</td>
</tr>
<tr>
<td>DSPCAGRC</td>
<td>CAGRP record</td>
</tr>
<tr>
<td>DSPCHGRC</td>
<td>CA record</td>
</tr>
<tr>
<td>DSPDBHRC</td>
<td>DB record</td>
</tr>
<tr>
<td>DSPDGRC</td>
<td>DBDSGRP record</td>
</tr>
<tr>
<td>DSPDSHRC</td>
<td>DBDS record</td>
</tr>
<tr>
<td>DSPGSGRC</td>
<td>Global Service Group record</td>
</tr>
</tbody>
</table>
Sample RECON Listing Active Site

DSPIMGRC IC record
DSPLGARC LOGALL record
DSPLOGRC PRILOG / SECLOG/PRISLD / SECSLD / PRITSLDS / SECTSLDS record
DSPOLDRC PRIOLD / SECOLD record
DSPPTNRC HALDB Partition record
DSPRCNRC HEADER record
DSPRCR1 HEADEREXT record
DSPRCVRC RECOV record
DSPRRGRRC REORG record
DSPRDTRC RECON DMB Table record
DSPSSRC SUBSYS record

Sample Listing of LIST.HISTORY Output

You use the LIST.HISTORY command to produce a history of activity for DBDSs or DEDB areas. You can produce a RECON listing using LIST.HISTORY from either an active or a remote site. [Figure 66 on page 471] is an example of LIST.HISTORY output that includes a time line.
LIST.HISTORY Sample

IMS VERSION 9 RELEASE 1 DATA BASE RECOVERY CONTROL

LIST.HISTORY DBD(DIVNTZ02) DDN(DBHVSAM1)
04.194 12:54:37.9 LISTING OF RECON

DB
DBD=DIVNTZ02 IRLMID=**NULL** DMB#=1 TYPE=IMS
SHARE LEVEL=3 GSGNAME=**NULL** USID=0000000005
AUTHORIZED USID=0000000005 RECEIVE USID=0000000005 HARD USID=0000000005
RECEIVE NEEDED USID=0000000000 DBCVGRP=**NULL**

FLAGS:
  BACKOUT NEEDED =OFF
  RECOVERY NEEDED COUNT =0
  READ ONLY =OFF
  IMAGE COPY NEEDED COUNT =0
  PROHIBIT AUTHORIZATION=OFF
  AUTHORIZED SUBSYSTEMS =0
  RECOVERABLE =YES
  EEQE COUNT =0
  TRACKING SUSPENDED =NO
  RECEIVE REQUIRED COUNT =0
  OFR REQUIRED =NO

DBDS
DSN=DIVNTZ02.FJXXS01K TYPE=IMS
DBD=DIVNTZ02 DDN=DBHVSAM1 DSID=001 DBORG=HISAM DSORG=VSAM

CAGRP=**NULL** ICGMAX=2 IC AVAIL=0 IC USED=1 DSSN=00000004
NOREUSE RECOVPD=0 OTHER DDN=**NULL**

DEFLTJCL=**NULL** ICJCL=ICJCL OICJCL=OICJCL RECJCL=RECJCL

FLAGS:
  IC NEEDED =OFF
  RECOV NEEDED =OFF
  RECEIVE NEEDED =OFF
  EEQE COUNT =0

ALLOC
ALLOC =04.194 12:49:55.0 * ALLOC LRID =0000000000000000
DSSN=0000000001 USID=0000000002 START =04.194 12:49:23.7
DEALLOC =04.194 12:50:19.8 DEALLOC LRID =0000000000000000

IMAGE
RUN =04.194 12:50:08.4 * RECORD COUNT =183
STOP =04.194 12:50:10.5 ONLINE USID=0000000002

IC1
DSN=IMSVS.DIVNTZ02.DBHVSAM1.IC.IC125005 FILE SEQ=0001
UNIT=SYSDA VOLS DEF=0001 VOLS USED=0001 VOLSER=222222
ALLOC
ALLOC =04.194 12:51:10.5 * ALLOC LRID =0000000000000000
DSSN=0000000002 USID=0000000003 START =04.194 12:49:23.7
DEALLOC =04.194 12:51:30.6 DEALLOC LRID =0000000000000000

ALLOC
04.194 12:54:37.9 LISTING OF RECON
ALLOC =04.194 12:52:21.5 * ALLOC LRID =0000000000000000
DSSN=0000000003 USID=0000000004 START =04.194 12:49:34.3
DEALLOC =04.194 12:52:27.7 DEALLOC LRID =0000000000000000

ALLOC
ALLOC =04.194 12:53:17.6 * ALLOC LRID =0000000000000000
DSSN=0000000004 USID=0000000005 START =04.194 12:49:34.3
DSP0181I NO CAGRP RECORD FOUND
DSP0181I NO CA RECORD FOUND
DSP0181I NO REORG RECORD FOUND
DSP0181I NO RECOV RECORD FOUND

Figure 66. Sample LIST.HISTORY Output (Part 1 of 5)
Figure 66. Sample LIST.HISTORY Output (Part 2 of 5)
LIST.HISTORY Sample

Figure 66. Sample LIST.HISTORY Output (Part 3 of 5)
LIST.HISTORY Sample

04.194 12:54:37.9 LISTING OF RECON PAGE 0005

-----------------------------------------------------------------------------
PRILOG
RECORD SIZE = 944
START = 04.194 12:49:34.3
STOP = 04.194 12:54:28.1
SSID=IMS2
VERSION=9.1
GSGNAME=**NULL**
FIRST RECORD ID= 0000000000000001
PRILOG TOKEN= 0
EARLIEST CHECKPOINT = 04.194 12:49:36.9

DSN=IMSVS.RLDSP.IMS2.D04194.T1249343.V00
UNIT=SYSDA
START = 04.194 12:49:34.3
STOP = 04.194 12:54:28.1
FIRST DS LSN= 0000000000000001
LAST DS LSN= 00000000000000010C
FILE SEQ=0001
#VOLUMES=0001
LOCK SEQUENCE#= 0

DSN=IMSVS.RLDSP.IMS2.D04194.T1250197.V00
UNIT=SYSDA
START = 04.194 12:50:19.7
STOP = 04.194 12:54:28.1
FIRST DS LSN= 000000000000010D
LAST DS LSN= 000000000000025E
FILE SEQ=0001
#VOLUMES=0001
LOCK SEQUENCE#= 0

DSN=IMSVS.RLDSP.IMS2.D04194.T1251305.V00
UNIT=SYSDA
START = 04.194 12:51:30.5
STOP = 04.194 12:54:27.5
FIRST DS LSN= 000000000000025F
LAST DS LSN= 0000000000000551
FILE SEQ=0001
#VOLUMES=0001
LOCK SEQUENCE#= BB81AFA0F956

DSN=IMSVS.RLDSP.IMS2.D04194.T1252282.V00
UNIT=SYSDA
START = 04.194 12:52:28.2
STOP = 04.194 12:54:27.5
FIRST DS LSN= 0000000000000552
LAST DS LSN= 0000000000000AA7
FILE SEQ=0001
#VOLUMES=0001
LOCK SEQUENCE#= BB81AFFFB0E

DSN=IMSVS.RLDSP.IMS2.D04194.T1252282.V01
UNIT=SYSDA
START = 04.194 12:52:28.2
STOP = 04.194 12:54:28.1
FIRST DS LSN= 0000000000000AA8
LAST DS LSN= 0000000000000B58
FILE SEQ=0001
#VOLUMES=0001
LOCK SEQUENCE#= BB81AFFFB0E

-----------------------------------------------------------------------------

Figure 66. Sample LIST.HISTORY Output (Part 4 of 5)
Sample Listing of a RECON at the Active Site

The following sections contain RECON record listings from an active site in an RSR environment. “Fields Present in a Listing of a RECON by Record Type” on page 528 describes the fields that can be present in a listing of the RECON data set.

In This Section:

- “RECON Status Record” on page 476
- “Log Records” on page 477
- “GSG Record” on page 486
- “SSYS Record” on page 487
- “BACKOUT Record” on page 487
- “CAGRP and CA Records” on page 488
- “DBGRP, DBDSGRP, and RECOVGRP Records” on page 489
- “DB (IMS) and Related Records” on page 490
Sample RECON Listing Active Site

- “DB (HALDB and PART) and Related Records” on page 492
- “DB (FP) and Related Records” on page 503

---

### RECON Status Record

<table>
<thead>
<tr>
<th>LIST.RECON STATUS</th>
<th>LISTING OF RECON</th>
<th>PAGE 0002</th>
</tr>
</thead>
<tbody>
<tr>
<td>04.197 11:04:46.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
RECON
RECOVERY CONTROL DATA SET, IMS V9R1
DMB#=9                INIT TOKEN=04196F1616381F
NOFORCER LOG DSN CHECK=CHECK17 STARTNEW=NO
TAPE UNIT=3400        DASD UNIT=SYSDA TRACEOFF SSID***NULL**
LIST DLOG=YES         CA/IC/LOG DATA SETS CATALOGED=NO
MINIMUM VERSION = 7.1  CROSS DBRC SERVICE LEVEL ID= 00001
LOG RETENTION PERIOD=00.000 00:15:00.0
COMMAND AUTH=NONE    HLQ=**NULL**
SIZALERT DSNUM=15    VOLNUM=16 PERCENT= 95
LOGALERT DSNUM=3     VOLNUM=16
REORG NUMBER VERIFICATION=NO
TIME STAMP INFORMATION:

  TIMEZIN = ¾SYS     -LABEL- -OFFSET-
  PDT    -07:00
  PST    -08:00

OUTPUT FORMAT: DEFAULT = LOCORG NONE PUNC YY
               CURRENT = LOCORG NONE PUNC YY

IMSPLEX = ** NONE **

- -DDNAME- -STATUS- -DATA SET NAME-
  RECON1  COPY1  IMSTESTL.IMS.RECON1
  RECON2  COPY2  IMSTESTL.IMS.RECON2
  RECON3  SPARE  IMSTESTL.IMS.RECON3
```

---

*Figure 67. Sample Listing of a RECON at the Active Site - RECON Status*
Sample RECON Listing Active Site

Log Records

LIST.LOG
04.197 11:04:43.8  LISTING OF RECON  PAGE 0002

PRILOG   RECORD SIZE= 464
START = 04.196 09:21:02.7  SSID=SYS3  VERSION=9.1
STOP = 04.196 09:28:46.7  #DSN=2
GSNAME=IMSGSG1
FIRST RECORD ID= 0000000000000001  PRILOG TOKEN= 1
EARLIEST CHECKPOINT = 04.196 09:21:04.5

DSN=IMS.V.RLDSP.SYS3.D04196.T0921027.V00  UNIT=SYSDA
START = 04.196 09:21:02.7  FIRST DS LSN= 0000000000000001
STOP = 04.196 09:24:54.6  LAST DS LSN= 0000000000000000
FILE SEQ=0001  #VOLUMES=0001
VOLSER=000000 STOPTIME = 04.196 09:24:54.6
CKPTCT=1  CHKPT ID = 04.196 09:21:04.5
LOCK SEQUENCE#= 000000000000

DSN=IMS.V.RLDSP.SYS3.D04196.T0924546.V00  UNIT=SYSDA
START = 04.196 09:24:54.6  FIRST DS LSN= 000000000000004D
STOP = 04.196 09:28:46.7  LAST DS LSN= 00000000000006B9
FILE SEQ=0001  #VOLUMES=0001
VOLSER=000000 STOPTIME = 04.196 09:28:46.7
CKPTCT=2  CHKPT ID = 04.196 09:28:46.3
LOCK SEQUENCE#= 000000000000

LOGALL
START = 04.196 09:21:02.7  *
EARLIEST ALLOC TIME = 04.196 09:22:45.2
DBDS ALLOC=11
-DBD-  -DDN-  -ALLOC-
DEDBJN21 DB21AR1  1
DEDBJN21 DB21AR2  1
DEDBJN21 DB21AR3  1
DEDBJN21 DB21AR4  1
DEDBJN21 DB21AR5  1
DEDBJN21 DB21AR6  1
DEDBJN21 DB21AR7  1
DEDBJN21 DB21AR8  1
DEDBJN21 DB21AR9  1
DEDBJN21 DB21AR10 1
DEDBJN21 DB21AR11 1

Figure 68. Sample Listing of a RECON at the Active Site - Log Records (Part 1 of 10)
Sample RECON Listing Active Site

04.197 11:04:43.8 LISTING OF RECON PAGE 0003

PRISLD RECORD SIZE= 464
START = 04.196 09:21:02.7 *SSID=SYS3 VERSION=9.1
STOP = 04.196 09:28:46.7 #DSN=2
GSGNAME=IMSGSG1
FIRST RECORD ID= 0000000000000001 PRILOG TOKEN= 1

DSN=IMSVS.SLDSP.SYS3.D04196.T0921027.V00 UNIT=SYSDA
START = 04.196 09:21:02.7 FIRST DS LSN= 0000000000000001
STOP = 04.196 09:24:54.6 LAST DS LSN= 000000000000004C
FILE SEQ=0001 #VOLUMES=0001
CHECKPOINT TYPES=80: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=N

VOLSER=000000 STOPTIME = 04.196 09:24:54.6
CKPTCT=1 CHKPT ID = 04.196 09:21:04.5
LOCK SEQUENCE=# 000000000000

DSN=IMSVS.SLDSP.SYS3.D04196.T0924546.V00 UNIT=SYSDA
START = 04.196 09:24:54.6 FIRST DS LSN= 000000000000004D
STOP = 04.196 09:28:46.7 LAST DS LSN= 0000000000000689
FILE SEQ=0001 #VOLUMES=0001
CHECKPOINT TYPES=88: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=Y

VOLSER=000000 STOPTIME = 04.196 09:28:46.7
CKPTCT=2 CHKPT ID = 04.196 09:28:46.3
LOCK SEQUENCE=# 000000000000

Figure 68. Sample Listing of a RECON at the Active Site - Log Records (Part 2 of 10)
Figure 68. Sample Listing of a RECON at the Active Site - Log Records (Part 3 of 10)
<table>
<thead>
<tr>
<th>PRISLD</th>
<th>RECORD SIZE=</th>
<th>START = 04.196 09:29:46.6</th>
<th>STOP = 04.196 09:47:18.5</th>
<th>SSID=SYS3</th>
<th>VERSION=9.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSGNAME=IMSGSG1</td>
<td>FIRST RECORD ID= 0000000000000001</td>
<td>PRILOG TOKEN= 2</td>
<td>DSN=**** COMPRESSED DATA SET ****</td>
<td>UNIT=</td>
<td>CHECKPOINT TYPES=80: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=N</td>
</tr>
<tr>
<td>DSN=IMSVS.SLDSP.SYS3.D04196.T0930286.V01</td>
<td>START = 04.196 09:30:28.6</td>
<td>STOP = 04.196 09:31:00.8</td>
<td>LAST DS LSN= 00000000000004D9</td>
<td>FILE SEQ=0001</td>
<td>#VOLUMES=0001</td>
</tr>
<tr>
<td>CHECKPOINT TYPES=40: SIMPLE=N SNAPQ=Y DUMPQ=N PURGE=N FREEZE=N</td>
<td>VOLSER=000000 STOPTIME = 04.196 09:31:01.8 LOCK SEQUENCE#= 000000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSN=IMSVS.SLDSP.SYS3.D04196.T0931008.V00</td>
<td>VOLSER=000000 STOPTIME = 04.196 09:31:21.2 LOCK SEQUENCE#= 000000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSN=IMSVS.SLDSP.SYS3.D04196.T0931212.V01</td>
<td>VOLSER=000000 STOPTIME = 04.196 09:47:18.5 LOCK SEQUENCE#= 000000000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 68. Sample Listing of a RECON at the Active Site - Log Records (Part 4 of 10)
Sample RECON Listing Active Site

04.197 11:04:43.8 LISTING OF RECON PAGE 0006

PRILOG
RECORD SIZE= 304
START = 04.196 09:47:38.3 * SSID=BATCH1 VERSION=9.1
STOP = 04.196 09:47:39.5 #DSN=1
GSGNAME=IMSGSG1
FIRST RECORD ID= 0000000000000001 PRILOG TOKEN= 3

DSN=BATCH1.UPDATEF.LOG
UNIT=SYSDA
START = 04.196 09:47:38.3 FIRST DS LSN= 0000000000000001
STOP = 04.196 09:47:39.5 LAST DS LSN= 000000000000009C
FILE SEQ=0001 #VOLUMES=0001
VOLSER=000000 STOPTIME = 04.196 09:47:39.5
CKPTCT=0 CHKPT ID = 00.000 00:00:00.0
LOCK SEQUENCE#= 000000000000

LOGALL
START = 04.196 09:47:38.3 *
EARLIEST ALLOC TIME = 04.196 09:47:39.0
DBDS ALLOC=3 DBD- DDN- ALLOC-
DIVNTZ02 HIDAM 1
DXVNTZ02 XDLBT04I 1
DIVNTZ02 DBHVSAM1 1

04.197 11:04:43.8 LISTING OF RECON PAGE 0007

SECLOG
RECORD SIZE= 304
START = 04.196 09:47:38.3 * SSID=BATCH1 VERSION=9.1
STOP = 04.196 09:47:39.5 #DSN=1
GSGNAME=IMSGSG1
FIRST RECORD ID= 0000000000000001 PRILOG TOKEN= 3

DSN=IMSTESTL.IMS01.OLDSP1
UNIT=SYSDA
START = 04.196 09:47:38.3 FIRST DS LSN= 0000000000000001
STOP = 04.196 09:47:39.5 LAST DS LSN= 000000000000009C
FILE SEQ=0001 #VOLUMES=0001
VOLSER=USER03 STOPTIME = 04.196 09:47:39.5
CKPTCT=0 CHKPT ID = 00.000 00:00:00.0
LOCK SEQUENCE#= 000000000000

Figure 68. Sample Listing of a RECON at the Active Site - Log Records (Part 5 of 10)
Figure 68. Sample Listing of a RECON at the Active Site - Log Records (Part 6 of 10)
Sample RECON Listing Active Site

04.197 11:04:43.8 LISTING OF RECON PAGE 0010

PRILOG
RECORD SIZE= 304
START = 04.196 09:48:49.8 * SSID=SYS3 VERSION=9.1
STOP = 04.196 08:51:28.4 #DSN=1
GSNAME=IMSGSG1
FIRST RECORD ID= 0000000000000001 PRILOG TOKEN= 4
EARLIEST CHECKPOINT = 04.196 09:31:14.9

DSN=IMSVS.RLDSP.SYS3.D04196.T0948498.V02 UNIT=SYSDA
START = 04.196 09:48:49.8 FIRST DS LSN= 0000000000000001
STOP = 04.196 08:51:28.4 LAST DS LSN= 00000000000002B9
FILE SEQ=0001 #VOLUMES=0001

VOLSER=000000 STOPTIME = 04.196 08:51:28.4
CKPTCT=1 CHKPT ID = 04.196 09:48:51.5
LOCK SEQUENCE#= 000000000000

LOGALL
START = 04.196 09:48:49.8 *
EARLIEST ALLOC TIME = 04.196 09:48:51.8
DBDS ALLOC=7 -DBD- -DDN- -ALLOC-
DEDBDD01 DD01AR0 1
DHVNTZ02 HIDAM 1
DXVNTZ02 XDLBT04I 1
DIVNTZ02 DBHVSAM1 1
DBOHID5 CKOHIG10 1
DXVHID5 CKVHIIXK 1
DBVHDJ05 CJVHDG1E 1

04.197 11:04:43.8 LISTING OF RECON PAGE 0011

PRILOG
RECORD SIZE= 304
START = 04.196 09:48:49.8 * SSID=SYS3 VERSION=9.1
STOP = 04.196 08:51:28.4 #DSN=1
GSNAME=IMSGSG1
FIRST RECORD ID= 0000000000000001 PRILOG TOKEN= 4

DSN=IMSVS.SLDSP.SYS3.D04196.T0948498.V02 UNIT=SYSDA
START = 04.196 09:48:49.8 FIRST DS LSN= 0000000000000001
STOP = 04.196 08:51:28.4 LAST DS LSN= 00000000000002B9
FILE SEQ=0001 #VOLUMES=0001

CHECKPOINT TYPES=80: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=N

VOLSER=000000 STOPTIME = 04.196 08:51:28.4
CKPTCT=1 CHKPT ID = 04.196 09:48:51.5
LOCK SEQUENCE#= 000000000000

Figure 68. Sample Listing of a RECON at the Active Site - Log Records (Part 7 of 10)
LISTING OF RECON PAGE 0012

PRILOG
START = 04.196 08:51:28.5  SSID=SYS3  VERSION=9.1
STOP  = 04.196 08:56:33.9  #DSN=2
GSNAME=IMSGSG1
FIRST RECORD ID= 00000000000002BA  PRILOG TOKEN= 5
EARLIEST CHECKPOINT = 04.196 08:51:30.4

DSN=IMSVS.RLDSP.SYS3.D04196.T0851285.V01
UNIT=SYSDA
START = 04.196 08:51:28.5  FIRST DS LSN= 00000000000002BA
STOP  = 04.196 08:53:43.2  LAST DS LSN= 0000000000000444
FILE SEQ=0001  #VOLUMES=0001

VOLSER=000000  STOP TIME = 04.196 08:53:43.2
CKPTCT=2  CHKPT ID = 04.196 08:51:30.4
LOCK SEQUENCE#= 000000000000

LOGALL
START = 04.196 08:51:28.5
EARLIEST ALLOC TIME = 04.196 08:51:31.6
DBDS ALLOC=1  -DBD-  -DDN-  -ALLOC-
DEDDB001 DD01AR0 1

Figure 68. Sample Listing of a RECON at the Active Site - Log Records (Part 8 of 10)
Figure 68. Sample Listing of a RECON at the Active Site - Log Records (Part 9 of 10)
Sample RECON Listing Active Site

04.197 11:04:43.8  LISTING OF RECON  PAGE 0017

PRIOLD
SSID=SYS3  # DD ENTRIES=4
EARLIEST CHECKPOINT = 04.196 08:51:30.4

DDNAME=DFSOLP3  DSN=IMSTESTL.IMS01.OLDSP3
START = 04.196 09:31:21.2  FIRST DS LSN= 00000000000005F8
STOP = 04.196 09:47:18.1  LAST DS LSN= 000000000000097AB
LOCK SEQUENCE# = 000000000000
STATUS=ARC COMPLT  FE OV=NO  A VAIL
PRILOG TIME=04.196 09:29:46.6  ARCHIVE JOB NAME=JT085128
VERSION=9.1

DDNAME=DFSOLP01  DSN=IMSTESTL.IMS01.OLDSP1
START = 04.196 09:48:49.8  FIRST DS LSN= 0000000000000001
STOP = 04.196 08:51:28.5  LAST DS LSN= 0000000000000444
LOCK SEQUENCE# = 000000000000
STATUS=ARC COMPLT  FE OV=NO  A VAIL
PRILOG TIME=04.196 09:48:49.8  ARCHIVE JOB NAME=JT085343
VERSION=9.1

DDNAME=DFSOLP02  DSN=IMSTESTL.IMS01.OLDSP2
START = 04.196 08:51:28.4  FIRST DS LSN= 000000000000002B9
STOP = 04.196 08:53:43.2  LAST DS LSN= 0000000000000444
LOCK SEQUENCE# = 000000000000
STATUS=ARC COMPLT  FE OV=NO  A VAIL
PRILOG TIME=04.196 08:51:28.5  ARCHIVE JOB NAME=JT085633
VERSION=9.1

DDNAME=DFSOLP00  DSN=IMSTESTL.IMS01.OLDSP0
START = 04.196 08:51:28.4  FIRST DS LSN= 000000000000002B9
STOP = 04.196 08:56:33.9  LAST DS LSN= 00000000000006F3
LOCK SEQUENCE# = 000000000000
STATUS=ARC COMPLT  FE OV=NO  A VAIL
PRILOG TIME=04.196 08:51:28.5  ARCHIVE JOB NAME=JT085633
VERSION=9.1

Figure 68. Sample Listing of a RECON at the Active Site - Log Records (Part 10 of 10)

GSG Record

04.197 11:04:25.7  LISTING OF RECON  PAGE 0002

GSG
GSGNAME=IMSGSG1  #SGS=2  -SGNAME-  -ROLE-
STLSITE1  TRACING  LOCAL
STLSITE2  ACTIVE

CURRENT PRILOG TOKEN = 6  TAKEOVER TOKEN = 0
MINIMUM PRILOG TOKEN = 1  DSN SEQ NUMBER = 0
START TIME OF CURRENT LOG = 04.196 08:55:03.1
HIGHEST ACTIVE SITE TIME = 00.000 00:00:00.0
TRACKING SUBSYSTEM ID = **NULL**
TAKEOVER IN PROGRESS
DSP0180I  NUMBER OF RECORDS LISTED IS  1

Figure 69. Sample Listing of a RECON at the Active Site - GSG Record
SSYS Record

04.201 07:41:46.5  LISTING OF RECON  PAGE 0017

SSYS
SSID=SYS3  LOG START=04.201 07:39:28.9
SSTYPE=ONLINE  ABNORMAL TERM=OFF  RECOVERY STARTED=NO  BACKUP=NO
TRACKED=NO  TRACKER TERM=OFF  SHARING COVERED DBS=NO
IRLMID=**NULL**  IRLM STATUS=NORMAL  GSGNAME=IMSGSG1

AUTHORIZED DATA BASES/AREAS=1  VERSION=9.1  XRF CAPABLE=NO
ENCORED

-DBD-  -AREA-  -LEVEL-  -ACCESS INTENT-  -STATE-
DEDBDD01  DD01AR0  1  UPDATE  6

Figure 70. Sample Listing of a RECON at the Active Site - SSYS Record

BACKOUT Record

04.197 11:04:33.4  LISTING OF RECON  PAGE 0002

BACKOUT
SSID=SYS3  #UORS=2

RECOVERY TOKEN=E2EBE2F3404040400000003000000002
TIME=04.196 09:49:19.0  PSB=PLVAPZ12
INFLT  BMP  COLDEND
ASSOCIATED DATA BASES=3

BACKED  DYN BKOUT
-DBD-  -OUT-  -FAILURE-
DHVNTZ02  NO  NO
DXVNTZ02  NO  NO
DIVNTZ02  NO  NO

RECOVERY TOKEN=E2EBE2F34040404000000030000000000
TIME=04.196 09:49:19.1  PSB=PSBEJK05
INFLT  BMP  COLDEND
ASSOCIATED DATA BASES=3

BACKED  DYN BKOUT
-DBD-  -OUT-  -FAILURE-
DBOHIDK5  NO  NO
DXVHIDK5  NO  NO
DBVHDJ05  NO  NO

Figure 71. Sample Listing of a RECON at the Active Site - BACKOUT Record
### CAGRP and CA Records

<table>
<thead>
<tr>
<th>04.197 11:04:30.2</th>
<th>LISTING OF RECON</th>
<th>PAGE 0002</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CAGRP</th>
<th>GRPNAME=CAGRP1</th>
<th>GRPMAX=3</th>
<th>CA AVAIL=0</th>
<th>CA USED=1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>NOREUSE</th>
<th>CAJCL=CAJCL</th>
<th>DEFLTJCL=<strong>NULL</strong></th>
<th>#MEMBERS=4</th>
<th>-DBD-</th>
<th>-DDN-</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DEDBJN21</th>
<th>DB21AR1</th>
<th>DEDBJN21</th>
<th>DB21AR3</th>
<th>DEDBJN21</th>
<th>DB21AR6</th>
<th>DEDBJN21</th>
<th>DB21AR7</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CA DSN=IMSVS.CAGRP1.CA.CA165501</th>
<th>FILE SEQ=1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CAGRP=CAGRP1</th>
<th>STOP = 04.196 09:24:54.6</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>UNIT=SYSDA</th>
<th>VOLS DEF=1</th>
<th>VOLS USED=1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>RUN = 04.196 08:55:39.9</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DBD=DEDBJN21</th>
<th>DDN=DB21AR1</th>
<th>PURGETIME = 04.196 09:22:45.2</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CHANGES ACCUMULATED=YES</th>
<th>COMPLETE CA=YES</th>
<th>INDOUBT EEQES=NO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LSN = 0000000000000000</th>
<th>DSSN = 0000000001</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LRID = 00000000000000003E</th>
<th>USID = 0000000002</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DBD=DEDBJN21</th>
<th>DDN=DB21AR3</th>
<th>PURGETIME = 04.196 09:22:46.0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CHANGES ACCUMULATED=YES</th>
<th>COMPLETE CA=YES</th>
<th>INDOUBT EEQES=NO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LSN = 0000000000000000</th>
<th>DSSN = 0000000001</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LRID = 00000000000000003E</th>
<th>USID = 0000000002</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DBD=DEDBJN21</th>
<th>DDN=DB21AR6</th>
<th>PURGETIME = 04.196 09:22:47.2</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CHANGES ACCUMULATED=YES</th>
<th>COMPLETE CA=YES</th>
<th>INDOUBT EEQES=NO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LSN = 0000000000000000</th>
<th>DSSN = 0000000001</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LRID = 000000000000000041</th>
<th>USID = 0000000002</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DBD=DEDBJN21</th>
<th>DDN=DB21AR7</th>
<th>PURGETIME = 04.196 09:22:47.7</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CHANGES ACCUMULATED=YES</th>
<th>COMPLETE CA=YES</th>
<th>INDOUBT EEQES=NO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LSN = 0000000000000000</th>
<th>DSSN = 0000000001</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LRID = 000000000000000042</th>
<th>USID = 0000000002</th>
</tr>
</thead>
</table>

*Figure 72. Sample Listing of a RECON at the Active Site - CAGRP and CA Records*
### DBGRP, DBDSGRP, and RECOVGRP Records

<table>
<thead>
<tr>
<th>04.197 12:14:19.7</th>
<th>LISTING OF RECON</th>
<th>PAGE 0002</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DBGPRP</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRPNAME=DBGRP1</td>
<td>#MEMBERS=6</td>
<td></td>
</tr>
<tr>
<td>-DBD/AREA-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIVNTZ02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DHVNTZ02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DXVNTZ02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB21AR0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB21AR1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB21AR2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| **DBDSGRP**       |                  |            |
| GRPNAME=FJKGRP    | #MEMBERS=5       |            |
| -DBD-             |                  |            |
| DIVNTZ02          |                  |            |
| HIDEAM            |                  |            |
| DIVNTZ02          |                  |            |
| HIDAM             |                  |            |
| DXVNTZ02          |                  |            |
| XDLB7041          |                  |            |

| **RECOVGRP**      |                  |            |
| GRPNAME=RCVGRP1   | #MEMBERS=5       |            |
| -DBD-             |                  |            |
| DIVNTZ02          |                  |            |
| DHVNTZ02          |                  |            |
| DEDBJN21          |                  |            |
| DEDBJN21          |                  |            |

*Figure 73. Sample Listing of a RECON at the Active Site - DBGRP, DBDSGRP, and RECOVGRP Records*
Sample RECON Listing Active Site

DB (IMS) and Related Records

Figure 74. Sample Listing of a RECON at the Active Site - DB (IMS) and Related Records (Part 1 of 2)
04.197 11:13:15.7  LISTING OF RECON  PAGE 0005

---

DB
DBD=DBVHDJ05  IRLMID=*NULL  DMB#=2  TYPE=IMS
SHARE LEVEL=3  GSGNAME=*NULL**  USID=0000000006
AUTHORIZED USID=0000000006  RECEIVE USID=0000000006  HARD USID=0000000006
RECEIVE NEEDED USID=0000000000
DBRCVGRP=**NULL**
FLAGS:  COUNTERS:
    BACKOUT NEEDED =ON  RECOVERY NEEDED COUNT =0
    READ ONLY =OFF  IMAGE COPY NEEDED COUNT =0
    PROHIBIT AUTHORIZATION=OFF  AUTHORIZED SUBSYSTEMS =1
    RECOVERABLE =YES  HELD AUTHORIZATION STATE=0
    EEQE COUNT =0
    TRACKING SUSPENDED =NO  RECEIVE REQUIRED COUNT =0
    OFR REQUIRED =NO
ASSOCIATED SUBSYSTEM INFORMATION:

---

04.197 11:13:15.7  LISTING OF RECON  PAGE 0006

---

DBDS
DSN=DBVHDJ05.CJXXD01E  TYPE=IMS
DBD=DBVHDJ05  DDN=CJVHDG1E  DSID=001  DBORG=HDAM  DSORG=VSAM
CAGRP=CAGRP2  GENMAX=2  IC AVAIL=0  IC USED=2  DSN=00000005
NOREUSE  RECOVPD=0  OTHER DDN=**NULL**
DEFLTJCL=**NULL**  ICJCL=ICJCL  OICJCL=OICJCL  RECOVJCL=RECOVJCL
RECVJCL=ICRCVJCL
FLAGS:  COUNTERS:
    IC NEEDED =OFF
    RECOV NEEDED =OFF
    RECEIVE NEEDED =OFF
    EEQE COUNT =0

---

ALLOC
ALLOC =04.196 09:30:57.0  * ALLOC LRID =0000000000000000
DSSN=0000000000 USID=0000000005 START = 04.196 09:29:46.6
DEALLOC =04.196 09:31:21.1  DEALLOC LRID =0000000000000000

ALLOC
ALLOC =04.196 08:49:43.0  * ALLOC LRID =0000000000000000
DSSN=0000000000 USID=0000000006 START = 04.196 09:48:49.8

IMAGE
RUN = 04.196 09:30:45.8  * RECORD COUNT=33
STOP = 00.000 00:00:00.0  BATCH USID=0000000004

IC1
DSN=IMSVS.DBVHDJ05.CJXHDGIE.IC.IC093043  FILE SEQ=0001
UNIT=SYSDA  VOLS DEF=0001 VOLS USED=0001
VOLSER=22222

IMAGE
RUN = 04.196 09:31:42.8  * RECORD COUNT=33
STOP = 04.196 09:31:43.0  CONCUR USID=0000000005

IC1
DSN=IMSVS.DBVHDJ05.CJVHDG1E.IC.IC093140  FILE SEQ=0001
UNIT=SYSDA  VOLS DEF=0001 VOLS USED=0001
VOLSER=222222

---

Figure 74. Sample Listing of a RECON at the Active Site - DB (IMS) and Related Records (Part 2 of 2)
Sample RECON Listing Active Site

DB (HALDB and PART) and Related Records

Figure 75. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 1 of 11)
Sample RECON Listing Active Site

04.194 12:30:54.1 LISTING OF RECON PAGE 0004

DBDS
DSN=IMSTESTS.DBOHIDK5.A00001 TYPE=PART
DBD=POHIDKA  DDN=POHIDKAA DSID=001 DBORG=HIDAM DSORG=OSAM
CAGRP=**NULL** GENMAX=2 IC AVAIL=0 IC USED=0 DSSN=00000004
NOREUSE RECOVPD=0 OTHER DDN=POHIDKAM
DEFLTJCL=**NULL** ICJCL=PICJCL OICJCL=POICJCL RECOVJCL=PRECOJCL
RECVJCL=PRECVJCL
FLAGS: 
COUNTERS:
IC NEEDED =OFF
IC RECOMMENDED =ON
RECOV NEEDED =OFF
RECEIVE NEEDED =OFF EEQE COUNT =0

ALLOC
ALLOC =04.194 11:03:22.4 * ALLOC LRID =0000000000000000
DSSN=0000000001 USID=0000000002 START = 04.194 11:03:20.4
ALLOC
ALLOC =04.194 11:04:10.1 * ALLOC LRID =0000000000000000
DSSN=0000000002 USID=0000000003 START = 04.194 11:04:06.6
ALLOC
ALLOC =04.194 11:09:03.6 * ALLOC LRID =0000000000000000
DSSN=0000000003 USID=0000000004 START = 04.194 11:07:30.7
DEALLOC =04.194 11:13:06.4 DEALLOC LRID =0000000000000000
ALLOC
ALLOC =04.194 11:19:31.3 * ALLOC LRID =0000000000000000
DSSN=0000000004 USID=0000000004 START = 04.194 11:07:30.7
DEALLOC =04.194 11:24:45.5 DEALLOC LRID =0000000000000000
REORG
RUN = 04.194 11:02:28.6 * USID = 0000000001
REORG#=00000
REORG
RUN = 04.194 11:03:02.2 * USID = 0000000001
REORG#=00000
REORG
RUN = 04.194 11:12:58.8 * USID = 0000000004
STOP = 04.194 11:13:06.2 ONLINE RECOV = NO
REORG
RUN = 04.194 11:19:31.3 * USID = 0000000004
REORG#=00000
STOP = 04.194 11:19:32.5 ONLINE RECOV = YES
REORG
RUN = 04.194 11:24:37.3 * USID = 0000000004
REORG#=00000
STOP = 04.194 11:24:45.3 ONLINE RECOV = NO

Figure 75. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 2 of 11)
Sample RECON Listing Active Site

04.194 12:30:54.1 LISTING OF RECON PAGE 0005

DBDS
DSN=IMSTESTS.DBOHIDK5.B00001
DBD=POHIDKA  DDN=POHIDKAB DSID=006 DBORG=HIDAM DSORG=OSAM
CAGR=**NULL**  GENMAX=2  IC AVAIL=0  IC USED=0  DSNN=00000004
NOREUSE RECOVPD=0  OTHER DDN=POHIDKAN
DEFLTJCL=**NULL**  ICJCL=ICJCL  OICJCL=POICJCL  RECOVJCL=PRECOJCL
RECVJCL=PRECVJCL
FLAGS:
COUNTERS:
IC NEEDED =OFF
IC RECOMMENDED =ON
RECOV NEEDED =OFF
RECEIVE NEEDED =OFF
EEQE COUNT =0

ALLOC
ALLOC =04.194 11:03:22.3  * ALLOC LRID =0000000000000000
DSN=0000000001 USID=0000000002 START = 04.194 11:03:20.4
ALLOC =04.194 11:04:10.0  * ALLOC LRID =0000000000000000
DSN=0000000002 USID=0000000003 START = 04.194 11:04:06.6
ALLOC =04.194 11:09:03.7  * ALLOC LRID =0000000000000000
DSN=0000000003 USID=0000000004 START = 04.194 11:07:30.7
DEALLOC =04.194 11:13:06.5  DEALLOC LRID =0000000000000000
ALLOC =04.194 11:19:31.8  * ALLOC LRID =0000000000000000
DSN=0000000004 USID=0000000005 START = 04.194 11:07:30.7
DEALLOC =04.194 11:24:45.7  DEALLOC LRID =0000000000000000
REORG
RUN = 04.194 11:02:28.6  * USID = 0000000001
REORG
RUN = 04.194 11:03:02.2  * USID = 0000000001
REORG
RUN = 04.194 11:12:58.8  * USID = 0000000004
STOP = 04.194 11:13:06.2  ONLINE RECOV = NO
REORG
RUN = 04.194 11:19:31.3  * USID = 0000000004
STOP = 04.194 11:19:32.5  ONLINE RECOV = YES
REORG
RUN = 04.194 11:24:37.3  * USID = 0000000004
STOP = 04.194 11:24:45.3  ONLINE RECOV = NO

Figure 75. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 3 of 11)
Sample RECON Listing Active Site

04.194 12:30:54.1 LISTING OF RECON PAGE 0006

DBDS
DSN=IMSTESTS.DBOHIDK5.L00001 TYPE=PART
DBD=POHIDKA  DDN=POHIDKAL DSID=003 DBORG=INDEX DSORG=VSAM
OTHER DDN=**NULL**
FLAGS:
RECOV NEEDED =OFF
EEQE COUNTERS:
EEQE COUNT =0

04.194 12:30:54.1 LISTING OF RECON PAGE 0007

DBDS
DSN=IMSTESTS.DBOHIDK5.M00001 TYPE=PART
DBD=POHIDKA  DDN=POHIDKAM DSID=001 DBORG=HIDAM DSORG=OSAM
CAGRP=**NULL** GENMAX=2 IC AVAIL=0 IC USID=0 DSSN=00000005
NOREUSE RECOVDP=0 OTHER DDN=POHIDKAA
DEFLTJCL=**NULL** ICJCL=PICJCL OICJCL=POICJCL RECOVJCL=PRECOJCL
RECVJCL=PRECVJCL
FLAGS:
IC NEEDED =OFF
IC RECOMMENDED =ON
RECOV NEEDED =OFF
RECEIVE NEEDED =OFF EEQE COUNTERS:
EEQE COUNT =0

ALLOC
ALLOC =04.194 11:12:58.8 * ALLOC LRID =0000000000000000
DSSN=0000000004 USID=0000000004 START = 04.194 11:07:30.7
DEALLOC =04.194 11:19:32.7 DEALLOC LRID =0000000000000000

ALLOC
ALLOC =04.194 11:24:37.3 * ALLOC LRID =0000000000000000
DSSN=0000000005 USID=0000000004 START = 04.194 11:07:30.7

REORG
RUN = 04.194 11:12:58.8 * USID = 0000000004
STOP = 04.194 11:13:06.2 ONLINE RECOV = YES

REORG
RUN = 04.194 11:19:31.3 * USID = 0000000004
STOP = 04.194 11:19:32.5 ONLINE RECOV = NO

REORG
RUN = 04.194 11:24:37.3 * USID = 0000000004
STOP = 04.194 11:24:45.3 ONLINE RECOV = YES

Figure 75. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 4 of 11)
Figure 75. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 5 of 11)
Sample RECON Listing Active Site

04.194 12:30:54.1 LISTING OF RECON PAGE 0011

DB
DBD=POHIDKB MASTER DB=BOHIDK5 IRLMID=**NULL** CHANGE#=3 TYPE=PART
USID=0000000004 AUTHORIZED USID=0000000004 HARD USID=0000000004
RECEIVE USID=0000000004 RECEIVE NEEDED USID=0000000000
DBRCVGRPS=**NULL**
DSN PREFIX=IMSTESTS.BOHIDK5 PARTITION ID=00002
PREVIOUS PARTITION=POHIDKA NEXT PARTITION=POHIDKC
OLRIMSID=**NULL** ACTIVE DBDS=A-J

FREE SPACE:
FREE BLOCK FREQ FACTOR=0 FREE SPACE PERCENTAGE=0

PARTITION HIGH KEY/STRING (CHAR): (LENGTH=5 ) K4000
PARTITION HIGH KEY/STRING (HEX):
D2F4F0F0F0404040404040404040404040404040404040404040404040404040

OSAM BLOCK SIZE:
A = 4096
B = 4096

FLAGS: COUNTERS:
BACKOUT NEEDED =OFF RECOVERY NEEDED COUNT =0
READ ONLY =OFF IMAGE COPY NEEDED COUNT =0
PROHIBIT AUTHORIZATION=OFF AUTHORIZED SUBSYSTEMS =0
TRACKING SUSPENDED =NO RECEIVE NEEDED COUNT =0
OREQUIRED =NO OLR ACTIVE HARD COUNT =0
PARTITION INIT NEEDED =NO OLR INACTIVE HARD COUNT =0
OLREORG CURSOR ACTIVE =NO
PARTITION DISABLED =NO
ONLINE REORG CAPABLE =YES
REORG INTENT =OFF

04.194 12:30:54.1 LISTING OF RECON PAGE 0012

DBDS
DSN=IMSTESTS.BOHIDK5.A00002 TYPE=PART
DBD=POHIDKB DDN=POHIDKBA DSID=001 DBORG=HIDAM DSORG=OSAM
CAGRP=**NULL** GENMAX=2 IC AVAIL=0 IC USED=0 DSSN=00000003
NOREUSE RECVPD=0 OTHER DSN=**NULL**
DEFINJCL=**NULL** IJCJCL=PICJCL OJCJCL=POICJCL RECVJCL=PRECVJCL
RECVJCL=PRECVJCL
FLAGS: COUNTERS:
IC NEEDED =OFF
RECOV NEEDED =OFF
RECEIVE NEEDED =OFF EEQE COUNT =0

Figure 75. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 6 of 11)
Sample RECON Listing Active Site

ALLOC
ALLOC = 04.194 11:03:25.4  *  ALLOC LRID = 0000000000000000
DSSN=0000000001 USID=0000000002 START = 04.194 11:03:20.4

ALLOC
ALLOC = 04.194 11:04:10.8  *  ALLOC LRID = 0000000000000000
DSSN=0000000002 USID=0000000003 START = 04.194 11:04:06.6

ALLOC
ALLOC = 04.194 11:06:34.8  *  ALLOC LRID = 0000000000000000
DSSN=0000000003 USID=0000000004 START = 04.194 11:06:30.7

REORG
RUN = 04.194 11:02:29.0  *  USID = 0000000001

REORG
RUN = 04.194 11:03:02.6  *  USID = 0000000001
04.194 12:30:54.1 LISTING OF RECON PAGE 0013

DBDS
DBS=IMSTESTS.DBOHIDK5.B00002  TYPE=PART
DBD=POHIDKB  DDN=POHIDKBB DSID=006 DBORG=HIDAM  DSORG=OSAM
CAGRP=**NULL**  GENMAX=2  IC AVAIL=0  IC USED=0  DSSN=00000000
NOREUSE  RECOVPD=0  OTHER DDN=**NULL**
DEFLTJCL=**NULL**  ICJCL=PICJCL  OICJCL=POICJCL  RECOVJCL=PRECOJCL
RECVJCL=PRECVJCL
FLAGS:  COUNTERS:
   IC NEEDED = OFF
   RECOV NEEDED = OFF
   RECEIVE NEEDED = OFF  EEQE COUNT = 0

ALLOC
ALLOC = 04.194 11:03:25.5  *  ALLOC LRID = 0000000000000000
DSSN=0000000001 USID=0000000002 START = 04.194 11:03:20.4

ALLOC
ALLOC = 04.194 11:04:13.5  *  ALLOC LRID = 0000000000000000
DSSN=0000000002 USID=0000000003 START = 04.194 11:04:06.6

ALLOC
ALLOC = 04.194 11:08:39.3  *  ALLOC LRID = 0000000000000000
DSSN=0000000003 USID=0000000004 START = 04.194 11:07:30.7

REORG
RUN = 04.194 11:02:29.0  *  USID = 0000000001

REORG
RUN = 04.194 11:03:02.6  *  USID = 0000000001

Figure 75. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 7 of 11)
Figure 75. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 8 of 11)
### Sample RECON Listing Active Site

**04.194 12:30:54.1 LISTING OF RECON PAGE 0017**

<table>
<thead>
<tr>
<th>DBDS</th>
<th>DSN=IMSTESTS.DBOHIDK5.A00003 TYPE=PART</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DBD=POHIDKC DDN=POHIDKCA DSID=001 DBORG=HIDAM DSORG=OSAM</td>
</tr>
<tr>
<td></td>
<td>CAGR=<strong>NULL</strong> GENMAX=2 IC AVAIL=0 IC USED=0 DSSN=000000001</td>
</tr>
<tr>
<td></td>
<td>NOREUSE RECOVPD=0 OTHER DDN=<strong>NULL</strong></td>
</tr>
<tr>
<td></td>
<td>DEFJCL=<strong>NULL</strong> ICJCL=PICJCL OICJCL=POICJCL RECOVJCL=PRECOJCL</td>
</tr>
<tr>
<td></td>
<td>RECJCL=PRECVJCL</td>
</tr>
<tr>
<td></td>
<td>FLAGS: COUNTERS:</td>
</tr>
<tr>
<td></td>
<td>IC NEEDED =OFF</td>
</tr>
<tr>
<td></td>
<td>RECOV NEEDED =OFF</td>
</tr>
<tr>
<td></td>
<td>RECEIVE NEEDED =OFF EEQE COUNT =0</td>
</tr>
</tbody>
</table>

**ALLOC**

<table>
<thead>
<tr>
<th>ALLOC</th>
<th>04.194 11:06:14.8 * ALLOC LRID =0000000000000000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DSSN=000000001 USID=0000000002 START = 04.194 11:07:30.7</td>
</tr>
</tbody>
</table>

**REORG**

<table>
<thead>
<tr>
<th>REORG</th>
<th>04.194 12:30:54.1 LISTING OF RECON PAGE 0018</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DBDS</td>
</tr>
<tr>
<td></td>
<td>DBD=POHIDKC DDN=POHIDKCB DSID=006 DBORG=HIDAM DSORG=OSAM</td>
</tr>
<tr>
<td></td>
<td>CAGR=<strong>NULL</strong> GENMAX=2 IC AVAIL=0 IC USED=0 DSSN=000000001</td>
</tr>
<tr>
<td></td>
<td>NOREUSE RECOVPD=0 OTHER DDN=<strong>NULL</strong></td>
</tr>
<tr>
<td></td>
<td>DEFJCL=<strong>NULL</strong> ICJCL=PICJCL OICJCL=POICJCL RECOVJCL=PRECOJCL</td>
</tr>
<tr>
<td></td>
<td>RECJCL=PRECVJCL</td>
</tr>
<tr>
<td></td>
<td>FLAGS: COUNTERS:</td>
</tr>
<tr>
<td></td>
<td>IC NEEDED =OFF</td>
</tr>
<tr>
<td></td>
<td>RECOV NEEDED =OFF</td>
</tr>
<tr>
<td></td>
<td>RECEIVE NEEDED =OFF EEQE COUNT =0</td>
</tr>
</tbody>
</table>

**ALLOC**

<table>
<thead>
<tr>
<th>ALLOC</th>
<th>04.194 11:06:14.8 * ALLOC LRID =0000000000000000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DSSN=000000001 USID=0000000002 START = 04.194 11:07:30.7</td>
</tr>
</tbody>
</table>

**REORG**

<table>
<thead>
<tr>
<th>REORG</th>
<th>04.194 12:30:54.1 LISTING OF RECON PAGE 0019</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DBDS</td>
</tr>
<tr>
<td></td>
<td>DBD=POHIDKC DDN=POHIDKCL DSID=003 DBORG=INDEX DSORG=VSAM</td>
</tr>
<tr>
<td></td>
<td>OTHER DDN=<strong>NULL</strong></td>
</tr>
<tr>
<td></td>
<td>FLAGS: COUNTERS:</td>
</tr>
<tr>
<td></td>
<td>RECOV NEEDED =OFF EEQE COUNT =0</td>
</tr>
</tbody>
</table>

---

**Figure 75.** Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 9 of 11)
Figure 75. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 10 of 11)
### Sample RECON Listing Active Site

<table>
<thead>
<tr>
<th>Time</th>
<th>Page</th>
<th>DBDS</th>
<th>DSN</th>
<th>DBD</th>
<th>DDN</th>
<th>DSID</th>
<th>DBORG</th>
<th>DSORG</th>
<th>CAGRP</th>
<th>GENMAX</th>
<th>IC AVAIL</th>
<th>IC USED</th>
<th>DSSN</th>
<th>NOREUSE</th>
<th>DEFLTJCL</th>
<th>OTHER DDN</th>
<th>ICJCL</th>
<th>OICJCL</th>
<th>RECOVJCL</th>
<th>RECVJCL</th>
<th>FLAGS:</th>
<th>COUNTERS:</th>
<th>EEQE COUNT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>04.194 12:30:54.1</td>
<td>0022</td>
<td>DBDS</td>
<td>DSN=IMSTESTS.DBOHIDK5.A00004</td>
<td>DBD=POHIDKD</td>
<td>DDN=POHIDKDA</td>
<td>DSID=001</td>
<td>DBORG=HIDAM</td>
<td>DSORG=OSAM</td>
<td>CAGRP=<strong>NULL</strong></td>
<td>GENMAX=2</td>
<td>IC AVAIL=0</td>
<td>IC USED=0</td>
<td>DSSN=00000001</td>
<td>NOREUSE</td>
<td>DEFLTJCL=<strong>NULL</strong></td>
<td>OTHER DDN=<strong>NULL</strong></td>
<td>ICJCL=PICJCL</td>
<td>OICJCL=POICJCL</td>
<td>RECOVJCL=PRECOJCL</td>
<td>RECVJCL=PRECVJCL</td>
<td>FLAGS:</td>
<td>COUNTERS:</td>
<td>IC NEEDED=OFF</td>
<td>RECOV NEEDED=OFF</td>
</tr>
<tr>
<td>04.194 11:07:52.2</td>
<td>0023</td>
<td>ALLOC</td>
<td>ALLOC=04.194 11:07:52.2</td>
<td>ALLOC LRID=0000000000000000</td>
<td>DSN=0000000001</td>
<td>USID=0000000002</td>
<td>START=04.194 11:07:30.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04.194 12:30:54.1</td>
<td>0024</td>
<td>REORG</td>
<td>RUN=04.194 11:02:29.9</td>
<td>USID=0000000001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04.194 12:30:54.1</td>
<td>0025</td>
<td>DBDS</td>
<td>DSN=IMSTESTS.DBOHIDK5.A00004</td>
<td>DBD=POHIDKD</td>
<td>DDN=POHIDKDB</td>
<td>DSID=006</td>
<td>DBORG=HIDAM</td>
<td>DSORG=OSAM</td>
<td>CAGRP=<strong>NULL</strong></td>
<td>GENMAX=2</td>
<td>IC AVAIL=0</td>
<td>IC USED=0</td>
<td>DSSN=00000001</td>
<td>NOREUSE</td>
<td>DEFLTJCL=<strong>NULL</strong></td>
<td>OTHER DDN=<strong>NULL</strong></td>
<td>ICJCL=PICJCL</td>
<td>OICJCL=POICJCL</td>
<td>RECOVJCL=PRECOJCL</td>
<td>RECVJCL=PRECVJCL</td>
<td>FLAGS:</td>
<td>COUNTERS:</td>
<td>IC NEEDED=OFF</td>
<td>RECOV NEEDED=OFF</td>
</tr>
<tr>
<td>04.194 11:07:52.3</td>
<td>0026</td>
<td>ALLOC</td>
<td>ALLOC=04.194 11:07:52.3</td>
<td>ALLOC LRID=0000000000000000</td>
<td>DSN=0000000001</td>
<td>USID=0000000002</td>
<td>START=04.194 11:07:30.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04.194 12:30:54.1</td>
<td>0027</td>
<td>REORG</td>
<td>RUN=04.194 11:02:29.9</td>
<td>USID=0000000001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04.194 12:30:54.1</td>
<td>0028</td>
<td>DBDS</td>
<td>DSN=IMSTESTS.DBOHIDK5.B00004</td>
<td>DBD=POHIDKD</td>
<td>DDN=POHIDKDB</td>
<td>DSID=006</td>
<td>DBORG=HIDAM</td>
<td>DSORG=OSAM</td>
<td>CAGRP=<strong>NULL</strong></td>
<td>GENMAX=2</td>
<td>IC AVAIL=0</td>
<td>IC USED=0</td>
<td>DSSN=00000001</td>
<td>NOREUSE</td>
<td>DEFLTJCL=<strong>NULL</strong></td>
<td>OTHER DDN=<strong>NULL</strong></td>
<td>ICJCL=PICJCL</td>
<td>OICJCL=POICJCL</td>
<td>RECOVJCL=PRECOJCL</td>
<td>RECVJCL=PRECVJCL</td>
<td>FLAGS:</td>
<td>COUNTERS:</td>
<td>IC NEEDED=OFF</td>
<td>RECOV NEEDED=OFF</td>
</tr>
<tr>
<td>04.194 11:07:52.3</td>
<td>0029</td>
<td>ALLOC</td>
<td>ALLOC=04.194 11:07:52.3</td>
<td>ALLOC LRID=0000000000000000</td>
<td>DSN=0000000001</td>
<td>USID=0000000002</td>
<td>START=04.194 11:07:30.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04.194 12:30:54.1</td>
<td>0030</td>
<td>REORG</td>
<td>RUN=04.194 11:02:29.9</td>
<td>USID=0000000001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04.194 12:30:54.1</td>
<td>0031</td>
<td>DBDS</td>
<td>DSN=IMSTESTS.DBOHIDK5.C00004</td>
<td>DBD=POHIDKD</td>
<td>DDN=POHIDKDC</td>
<td>DSID=003</td>
<td>DBORG=INDEX</td>
<td>DSORG=VSAM</td>
<td>OTHER DDN=<strong>NULL</strong></td>
<td>IC NEEDED=OFF</td>
<td>RECOV NEEDED=OFF</td>
<td>RECEIVE NEEDED=OFF</td>
<td>EEQE COUNT=0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 75. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 11 of 11)*
Sample RECON Listing Active Site

DB (FP) and Related Records

04.197 11:04:40.0 LISTING OF RECON PAGE 0002

-------------------------------------------------------------------------------
DB
DBD=DEDBDD01 DMB#=8 TYPE=FP
SHARE LEVEL=1
FLAGS:
  COUNTERS:
    RECOVERY NEEDED COUNT =0
    IMAGE COPY NEEDED COUNT =0
PROHIBIT AUTHORIZATION=OFF
RECOVERABLE =YES EEQE COUNT =0
-------------------------------------------------------------------------------

04.197 11:04:40.0 LISTING OF RECON PAGE 0003

-------------------------------------------------------------------------------
DBDS
DBD=DEDBDD01 AREA=DD01AR0 TYPE=FP
SHARE LEVEL=1
DSID=00001 DBORG=DEDB DSORG=VSAM
GSGNAME=IMSGSG1 USID=0000000003
AUTHORIZED USID=0000000003 RECEIVE USID=0000000003 HARD USID=0000000003
RECEIVE NEEDED USID=0000000000 CAGRP=**NULL**
NOREUSE RECOVPD=0 NOVSO PREOPEN NOPRELOAD
DSNAME=IMSVS.DEDBDD01.SMSCIC DSN=IMSVS.DEDBDD01.SMSCIC.DSN1
FILE SEQ=0001
UNIT=SYSDA VOLS DEF=0001 VOLS USED=0001 VOLSER=IMSCC1

Figure 76. Sample Listing of a RECON at the Active Site - DB (FP) and Related Records
Sample Listing of a RECON at the Tracking Site

Beginning with Figure 77, the following figures comprise a listing of a RECON from a tracking site in an RSR environment. “Fields Present in a Listing of a RECON by Record Type” on page 528 describes the fields that can be present in a listing of the RECON.

In This Section:
- “RECON Status Record”
- “Log Records” on page 505
- “GSG Record” on page 511
- “SSYS Record” on page 512
- “BACKOUT Record” on page 512
- “CAGRP and CA Records” on page 513
- “DBDSGRP Records” on page 514
- “DB (IMS) and Related Records” on page 515
- “DB (HALDB and PART) and Related Records” on page 517
- “DB (FP) and Related Records” on page 528

RECON Status Record

04.197 11:48:57.0 LISTING OF RECON PAGE 0002

RECON
RECOVERY CONTROL DATA SET, IMS V9R1
DMBS=8 INIT TOKEN=04196F1616485F
NOFORCER LOG DSN CHECK=CHECK17 STARTNEW=NO
TAPE UNIT=3400 DASD UNIT=SYSDA TRACEOFF SSID***NULL**
LIST DLOG=YES CA/IC/LOG DATA SETS CATALOGED=NO
MINIMUM VERSION = 7.1 CROSS DBRC SERVICE LEVEL ID= 000001
LOG RETENTION PERIOD=00.000 00:15:00.0
COMMAND AUTH=NONE HLQ=**NULL**
SIZALERT DSNUM=15 VOLNUM=16 PERCENT= 95
LOGALERT DSNUM=3 VOLNUM=16
REORG NUMBER VERIFICATION=YES
TIME STAMP INFORMATION:

TIMEZIN = %SYS -LABEL- -OFFSET-
PDT -07:00
PST -08:00

OUTPUT FORMAT: DEFAULT = LOCORG NONE PUNC YY
CURRENT = LOCORG NONE PUNC YY

IMSPLEX = ** NONE **

-DDNAME- -STATUS- -DATA SET NAME-
RECON1 COPY1 IMSTESTL.IMS.RECON1
RECON2 COPY2 IMSTESTL.IMS.RECON2
RECON3 SPARE IMSTESTL.IMS.RECON3

Figure 77. Sample Listing of a RECON at the Tracking Site - RECON Status Record
Log Records

04.197 11:48:55.3 LISTING OF RECON PAGE 0002

PRILOG
START = 04.196 09:21:02.7 * SSID=SYS3 VERSION=9.1
STOP = 04.196 09:28:46.6
GSGNAME=IMSGSG1 TRACKING
FIRST RECORD ID= 0000000000000001 PRILOG TOKEN= 1
EARLIEST CHECKPOINT = 04.196 09:21:04.5

DSN=IMSTESTL.RSR.RLDS1.N0000005
START = 04.196 09:21:02.7 FIRST DS LSN= 0000000000000001
STOP = 04.196 09:24:54.6 LAST DS LSN= 000000000000004C
#DS CHECKPOINTS= 1 CHKPT ID = 00.000 00:00:00.0

DSN=IMSTESTL.RSR.RLDS1.N0000008
START = 04.196 09:24:54.6 FIRST DS LSN= 0000000000000044D
STOP = 04.196 09:28:46.4 LAST DS LSN= 00000000000000624
#DS CHECKPOINTS= 2 CHKPT ID = 04.196 09:24:54.7

DSN=IMSTESTL.RSR.RLDS1.N0000009
START = 04.196 09:28:46.4 FIRST DS LSN= 00000000000000625
STOP = 04.196 09:28:46.6 LAST DS LSN= 00000000000000689
#DS CHECKPOINTS= 0 CHKPT ID = 04.196 09:28:46.3

LOGALL
START = 04.196 09:21:02.7 *
EARLIEST ALLOC TIME = 04.196 09:22:45.2

DBDS ALLOC=5
-DBD- -DDN- -ALLOC-
DEDBJN21 DB21AR1 1
DEDBJN21 DB21AR2 1
DEDBJN21 DB21AR3 1
DEDBJN21 DB21AR4 1
DEDBJN21 DB21AR5 1

04.197 11:48:55.3 LISTING OF RECON PAGE 0003

PRISLD
START = 04.196 09:21:02.7 * SSID=SYS3 VERSION=9.1
STOP = 04.196 09:28:46.6
GSGNAME=IMSGSG1 TRACKING
FIRST RECORD ID= 0000000000000001 PRILOG TOKEN= 1

DSN=IMSTESTL.RSR.ARCH1.N0000004
START = 04.196 09:21:02.7 FIRST DS LSN= 0000000000000001
STOP = 04.196 09:24:54.6 LAST DS LSN= 0000000000000044C
#DS CHECKPOINTS= 1 CHKPT ID = 00.000 00:00:00.0
CHECKPOINT TYPES=80: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=N

DSN=IMSTESTL.RSR.ARCH1.N0000007
START = 04.196 09:24:54.6 FIRST DS LSN= 0000000000000044D
STOP = 04.196 09:28:46.4 LAST DS LSN= 00000000000000624
#DS CHECKPOINTS= 2 CHKPT ID = 04.196 09:24:54.7
CHECKPOINT TYPES=88: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=Y

DSN=IMSTESTL.RSR.ARCH1.N0000009
START = 04.196 09:28:46.4 FIRST DS LSN= 00000000000000625
STOP = 04.196 09:28:46.6 LAST DS LSN= 00000000000000689
#DS CHECKPOINTS= 0 CHKPT ID = 04.196 09:28:46.3

Figure 78. Sample Listing of a RECON at the Tracking Site - Log Records (Part 1 of 7)
Sample RECON Listing Tracking Site

CHECKPOINT TYPES=00: SIMPLE=N SNAPQ=N DUMPQ=N PURGE=N FREEZE=N
04.197 11:48:55.3 LISTING OF RECON PAGE 0004

-------------------------------------------------------------------------------
PRILOG RECORD SIZE= 744
START = 04.196 09:29:46.6 * SSID=SYS3 VERSION=9.1
STOP = 04.196 09:47:18.3 #DSN=4
GSGNAME=IMSGSG1 TRACKING
FIRST RECORD ID= 0000000000000001 PRILOG TOKEN= 2
EARLIEST CHECKPOINT = 04.196 09:31:14.9

DSN***** COMPRESSED DATA SET ****
START = 04.196 09:29:46.6 FIRST DS LSN= 0000000000000001
STOP = 04.196 09:31:06.2 LAST DS LSN= 0000000000000409
#DS CHECKPOINTS= 1

DSN=IMSTESTL.RSR.RLDS1.N0000023
START = 04.196 09:31:06.2 FIRST DS LSN= 0000000000000040DA
STOP = 04.196 09:31:21.2 LAST DS LSN= 00000000000005F7
#DS CHECKPOINTS= 1 CHKPT ID = 04.196 09:31:14.9

DSN=IMSTESTL.RSR.RLDS1.N0000026
START = 04.196 09:31:21.2 FIRST DS LSN= 00000000000005F8
STOP = 04.196 09:47:18.1 LAST DS LSN= 00000000000007BD
#DS CHECKPOINTS= 2 CHKPT ID = 00.000 00:00:00.0

DSN=IMSTESTL.RSR.RLDS1.N0000028
START = 04.196 09:47:18.1 FIRST DS LSN= 00000000000007BE
STOP = 04.196 09:47:18.3 LAST DS LSN= 0000000000000828
#DS CHECKPOINTS= 0 CHKPT ID = 04.196 09:47:18.1

LOGALL
START = 04.196 09:29:46.6 *
EARLIEST ALLOC TIME = 00.000 00:00:00.0
DBDS ALLOC=0
04.197 11:48:55.3 LISTING OF RECON PAGE 0005
-------------------------------------------------------------------------------
PRISLD RECORD SIZE= 744
START = 04.196 09:29:46.6 * SSID=SYS3 VERSION=9.1
STOP = 04.196 09:47:18.3 #DSN=4
GSGNAME=IMSGSG1 TRACKING
FIRST RECORD ID= 0000000000000001 PRILOG TOKEN= 2

DSN***** COMPRESSED DATA SET ****
START = 04.196 09:29:46.6 FIRST DS LSN= 0000000000000001
STOP = 04.196 09:31:06.2 LAST DS LSN= 0000000000000409
#DS CHECKPOINTS= 1

CHECKPOINT TYPES=80: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=N

DSN=IMSTESTL.RSR.ARCH1.N0000022
START = 04.196 09:31:06.2 FIRST DS LSN= 000000000000040DA
STOP = 04.196 09:31:21.2 LAST DS LSN= 00000000000005F7
#DS CHECKPOINTS= 1 CHKPT ID = 04.196 09:31:14.9
CHECKPOINT TYPES=40: SIMPLE=N SNAPQ=Y DUMPQ=N PURGE=N FREEZE=N

DSN=IMSTESTL.RSR.ARCH1.N0000025
START = 04.196 09:31:21.2 FIRST DS LSN= 00000000000005F8
STOP = 04.196 09:47:18.1 LAST DS LSN= 00000000000007BD
#DS CHECKPOINTS= 2 CHKPT ID = 00.000 00:00:00.0
CHECKPOINT TYPES=88: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=Y

DSN=IMSTESTL.RSR.ARCH1.N0000027
START = 04.196 09:47:18.1 FIRST DS LSN= 00000000000007BE
STOP = 04.196 09:47:18.3 LAST DS LSN= 0000000000000828
#DS CHECKPOINTS= 0 CHKPT ID = 04.196 09:47:18.1
CHECKPOINT TYPES=00: SIMPLE=N SNAPQ=N DUMPQ=N PURGE=N FREEZE=N

Figure 78. Sample Listing of a RECON at the Tracking Site - Log Records (Part 2 of 7)
### Sample RECON Listing Tracking Site

#### PRILOG
- **Record Size:** 304
- **Start:** 04.196 09:47:38.3
- **Stop:** 04.196 09:47:39.4
- **SSID:** BATCH1
- **Version:** 9.1
- **DSN:** 1
- **GSGName:** IMSGSG1
- **Tracking
- **First Record ID:** 0000000000000001
- **PRILOG Token:** 3

#### PRILOG
- **Record Size:** 464
- **Start:** 04.196 09:48:49.8
- **Stop:** 04.196 08:51:28.4
- **SSID:** SYS3
- **Version:** 9.1
- **DSN:** 2
- **GSGName:** IMSGSG1
- **Tracking
- **First Record ID:** 0000000000000001
- **PRILOG Token:** 4

#### PRILOG
- **Record Size:** 464
- **Start:** 04.196 09:48:49.8
- **Stop:** 04.196 08:51:28.4
- **SSID:** SYS3
- **Version:** 9.1
- **DSN:** 2
- **GSGName:** IMSGSG1
- **Tracking
- **First Record ID:** 0000000000000001
- **PRILOG Token:** 4

---

**Figure 78. Sample Listing of a RECON at the Tracking Site - Log Records (Part 3 of 7)**
Sample RECON Listing Tracking Site

DSN=IMSTESTL.RSR.ARCH1.N0000036
START = 04.196 08:49:42.8 FIRST DS LSN= 00000000000000236
STOP = 04.196 08:51:28.4 LAST DS LSN= 00000000000000289
#DS CHECKPOINTS= 0 CHKPT ID = 00.000 00:00:00.0
CHECKPOINT TYPES=00: SIMPLE=N SNAPQ=N DUMPQ=N PURGE=N FREEZE=N
04.197 11:48:55.3 LISTING OF RECON PAGE 0009
--------------------------------------------------------------------------
PRILOG RECORD SIZE= 944
START = 04.196 08:51:28.5 * SSID=SYS3 VERSION=9.1
STOP = 04.196 08:56:33.8 #DSN=5
GSGNAME=IMSGSG1 TRACKING
FIRST RECORD ID= 0000000000000236 PRILOG TOKEN= 5
EARLIEST CHECKPOINT = 04.196 08:51:30.4

DSN=IMSTESTL.RSR.RLDS1.N0000039
START = 04.196 08:51:28.5 FIRST DS LSN= 00000000000000236
STOP = 04.196 08:51:30.6 LAST DS LSN= 00000000000000378
#DS CHECKPOINTS= 1 CHKPT ID = 04.196 09:48:51.5

DSN=IMSTESTL.RSR.SLDS1.N0000047
START = 04.196 08:51:30.6 FIRST DS LSN= 00000000000000379
STOP = 04.196 08:53:43.2 LAST DS LSN= 00000000000000444
#DS CHECKPOINTS= 1 CHKPT ID = 04.196 08:51:30.4

DSN=IMSTESTL.RSR.SLDS1.N0000052
START = 04.196 08:53:43.2 FIRST DS LSN= 00000000000000445
STOP = 04.196 08:53:43.3 LAST DS LSN= 000000000000004C2
#DS CHECKPOINTS= 0 CHKPT ID = 04.196 08:51:30.4

DSN=IMSTESTL.RSR.RLDS1.N0000049
START = 04.196 08:53:43.3 FIRST DS LSN= 000000000000004C3
STOP = 04.196 08:53:43.7 LAST DS LSN= 00000000000000692
#DS CHECKPOINTS= 2 CHKPT ID = 04.196 08:53:52.8

DSN=IMSTESTL.RSR.RLDS1.N0000051
START = 04.196 08:56:33.7 FIRST DS LSN= 00000000000000693
STOP = 04.196 08:56:33.8 LAST DS LSN= 000000000000006F3
#DS CHECKPOINTS= 0 CHKPT ID = 04.196 08:56:33.6

LOGALL
START = 04.196 08:51:28.5 *
EARLIEST ALLOC TIME = 04.196 08:51:31.6
DBDS ALLOC=1 -DBD- -DDN- -ALLOC- DEDBDD01 DD01AR0 1

Figure 78. Sample Listing of a RECON at the Tracking Site - Log Records (Part 4 of 7)
Figure 78. Sample Listing of a RECON at the Tracking Site - Log Records (Part 5 of 7)
Sample RECON Listing Tracking Site

LOGALL
START = 04.196 08:55:03.1
EARLIEST ALLOC TIME = 00.000 00:00:00.0
DBDS ALLOC=0
04.197 11:48:55.3 LISTING OF RECON PAGE 0012

-------------------------------------------------------------------------------
PRISLD
RECORD SIZE= 464
START = 04.196 08:55:03.1
SSID=IMS2
VERSION=9.1

STOP = 04.196 08:55:33.3
GSGNAME=IMSGSG1 TRACKING
FIRST RECORD ID= 0000000000000001
PRILOG TOKEN= 6

DSN=IMSTESTL.RSR.ARCH1.N0000043
START = 04.196 08:55:03.1
FIRST DS LSN= 0000000000000001
STOP = 04.196 08:55:33.2
LAST DS LSN= 00000000000001A4
#DS CHECKPOINTS= 2 CHKPT ID = 04.196 08:55:04.9
CHECKPOINT TYPES=88: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=Y

DSN=IMSTESTL.RSR.ARCH1.N0000045
START = 04.196 08:55:33.2
FIRST DS LSN= 00000000000001A5
STOP = 04.196 08:55:33.3
LAST DS LSN= 0000000000000205
#DS CHECKPOINTS= 0 CHKPT ID = 04.196 08:55:33.1
CHECKPOINT TYPES=00: SIMPLE=N SNAPQ=N DUMPQ=N PURGE=N FREEZE=N
04.197 11:48:55.3 LISTING OF RECON PAGE 0013

-------------------------------------------------------------------------------
PRIOLD
SSID=SYS3
# DD ENTRIES=2
EARLIEST CHECKPOINT = 04.196 09:20:23.6

DDNAME=DFSOLP00 DSN=IMSTESTL.IMS01.OLDSP0
START = 04.196 09:20:22.9
FIRST DS LSN= 0000000000000001
STOP = 04.196 08:53:56.8
LAST DS LSN= 0000000000000051E
LOCK SEQUENCE# = 000000000000
STATUS=ARC COMPLT FEOV=YES AVAIL
PRILOG TIME=04.196 09:20:22.9
ARCHIVE JOB NAME=JT085357
VERSION=9.1

DDNAME=DFSOLP01 DSN=IMSTESTL.IMS01.OLDSP1
START = 04.196 08:53:56.8
FIRST DS LSN= 0000000000000051F
STOP = 04.196 08:57:01.5
LAST DS LSN= 000000000000007AF
LOCK SEQUENCE# = 000000000000
STATUS=ARC COMPLT FEOV=NO AVAIL
PRILOG TIME=04.196 09:20:22.9
ARCHIVE JOB NAME=JT085701
VERSION=9.1

Figure 78. Sample Listing of a RECON at the Tracking Site - Log Records (Part 6 of 7)
### Sample RECON Listing Tracking Site

<table>
<thead>
<tr>
<th>04.197 11:48:55.3</th>
<th>LISTING OF RECON</th>
<th>PAGE 0014</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRITSLDS RECORD SIZE= 464</td>
<td></td>
<td></td>
</tr>
<tr>
<td>START = 04.196 09:20:22.9 * SSID=SYS3 VERSION=9.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOP = 04.196 08:57:01.5 #DSN=2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSGNAME=IMSGSG1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIRST RECORD ID= 0000000000000001 PRILOG TOKEN= 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSN=IMSVS.SLDSP.SYS3.D04196.T0920229.V00 UNIT=SYSDA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>START = 04.196 09:20:22.9 FIRST DS LSN= 0000000000000001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOP = 04.196 08:53:56.8 LAST DS LSN= 000000000000051E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILE SEQ=0001 #VOLUMES=0001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VOLSER=000000 STOPTIME = 04.196 08:53:56.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CKPTCT=1 CHKPT ID = 04.196 09:20:23.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOCK SEQUENCE# = 000000000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSN=IMVS.SLDSP.SYS3.D04196.T0853568.V00 UNIT=SYSDA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>START = 04.196 08:53:56.8 FIRST DS LSN= 000000000000051F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOP = 04.196 08:57:01.5 LAST DS LSN= 00000000000007AF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILE SEQ=0001 #VOLUMES=0001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VOLSER=000000 STOPTIME = 04.196 08:57:01.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CKPTCT=1 CHKPT ID = 04.196 08:57:01.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOCK SEQUENCE# = 000000000000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 78. Sample Listing of a RECON at the Tracking Site - Log Records (Part 7 of 7)*

### GSG Record

<table>
<thead>
<tr>
<th>04.197 11:48:28.1</th>
<th>LISTING OF RECON</th>
<th>PAGE 0002</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSGNAME=IMSGSG1 #SGS=2</td>
<td>-SGNAME- -ROLE-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STLSITE1 TRACKING</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STLSITE2 ACTIVE LOCAL</td>
<td></td>
</tr>
<tr>
<td>CURRENT PRILOG TOKEN = 6 TAKEOVER TOKEN = 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MINIMUM PRILOG TOKEN = 1 DSN SEQ NUMBER = 52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>START TIME OF CURRENT LOG = 04.196 08:55:03.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIGHEST ACTIVE SITE TIME = 04.196 08:56:33.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRACKING SUBSYSTEM ID = SYS3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAKEOVER IN PROGRESS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 79. Sample Listing of a RECON at the Tracking Site - GSG Record*
Sample RECON Listing Tracking Site

SSYS Record

04.201 07:41:52.9 LISTING OF RECON PAGE 0015

<table>
<thead>
<tr>
<th>SSYS</th>
<th>SSID=SYS3</th>
<th>LOG START=04.201 08:08:18.6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SSTYPE=TRACKER</td>
<td>ABNORMAL TERM=OFF</td>
</tr>
<tr>
<td></td>
<td>TRACKED=NO</td>
<td>TRACKER TERM=OFF</td>
</tr>
<tr>
<td></td>
<td>GSGNAME=IMSGSG1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AUTHORIZED DATA BASES/AREAS=3</td>
<td>VERSION=9.1</td>
</tr>
<tr>
<td></td>
<td>-DBD-</td>
<td>AREA-</td>
</tr>
<tr>
<td></td>
<td>DHVNTZ02</td>
<td><strong>NULL</strong></td>
</tr>
<tr>
<td></td>
<td>DXVNTZ02</td>
<td><strong>NULL</strong></td>
</tr>
<tr>
<td></td>
<td>DIVNTZ02</td>
<td><strong>NULL</strong></td>
</tr>
</tbody>
</table>

04.201 07:41:52.9 LISTING OF RECON PAGE 0016

<table>
<thead>
<tr>
<th>SSYS</th>
<th>SSID=SYS3</th>
<th>LOG START=04.201 08:36:48.4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SSTYPE=ONLINE</td>
<td>ABNORMAL TERM=ON</td>
</tr>
<tr>
<td></td>
<td>TRACKED=NO</td>
<td>TRACKER TERM=OFF</td>
</tr>
<tr>
<td></td>
<td>IRLMID=<strong>NULL</strong></td>
<td>IRLM STATUS=NORMAL</td>
</tr>
<tr>
<td></td>
<td>AUTHORIZED DATA BASES/AREAS=4</td>
<td>VERSION=9.1</td>
</tr>
<tr>
<td></td>
<td>-DBD-</td>
<td>AREA-</td>
</tr>
<tr>
<td></td>
<td>DEBDD01</td>
<td>DD01AR0</td>
</tr>
<tr>
<td></td>
<td>DHVNTZ02</td>
<td><strong>NULL</strong></td>
</tr>
<tr>
<td></td>
<td>DXVNTZ02</td>
<td><strong>NULL</strong></td>
</tr>
<tr>
<td></td>
<td>DIVNTZ02</td>
<td><strong>NULL</strong></td>
</tr>
</tbody>
</table>

Figure 80. Sample Listing of a RECON at the Tracking Site - SSYS Record

BACKOUT Record

04.197 11:48:49.2 LISTING OF RECON PAGE 0002

<table>
<thead>
<tr>
<th>BACKOUT</th>
<th>SSID=SYS3</th>
<th>#UORS=1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RECOVERY TOKEN=E28E2F340404040404000000030000000002</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TIME=04.196 09:49:19.0</td>
<td>PSB=PLVAPZ12</td>
</tr>
<tr>
<td></td>
<td>INFLT BMP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ASSOCIATED DATA BASES=3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-DBD-</td>
<td>OUT-</td>
</tr>
<tr>
<td></td>
<td>DHVNTZ02</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>DXVNTZ02</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>DIVNTZ02</td>
<td>NO</td>
</tr>
</tbody>
</table>

Figure 81. Sample Listing of a RECON at the Tracking Site - BACKOUT Record
CAGRP and CA Records

04.197 11:48:44.0  LISTING OF RECON  PAGE 0002

CAGRP
GRPNAME=CAGRP1  GRPMAX=3  CA AVAIL=0  CA USED=1
NOREUSE  CAJCL=CAJCL  DEFLTJCL=NULL**  #MEMBERS=4

-DBD-  -DDN-
DEDBJN21  DB21AR1
DEDBJN21  DB21AR3
DEDBJN21  DB21AR6
DEDBJN21  DB21AR7

CA
DSN=IMVS.CAGRP1.CA.CA165501  FILE SEQ=1
CAGRP=CAGRP1  STOP = 04.196 09:24:54.6
UNIT=SYSDA  VOLS DEF=1  VOLS USED=1
VOLSER=222222
RUN = 04.196 08:55:47.2
DBD=DEDBJN21  DDN=DB21AR1  PURGETIME = 04.196 09:18:36.0
CHANGES ACCUMULATED=YES  COMPLETE CA=YES INDUBT EEQES=NO
LSN = 000000000001  DSSN = 0000000001
LRID = 000000000003E4  USID = 0000000002
DBD=DEDBJN21  DDN=DB21AR3  PURGETIME = 04.196 09:18:36.0
CHANGES ACCUMULATED=YES  COMPLETE CA=YES INDUBT EEQES=NO
LSN = 000000000001  DSSN = 0000000001
LRID = 000000000003FE  USID = 0000000002
DBD=DEDBJN21  DDN=DB21AR6  PURGETIME = 04.196 09:18:36.0
CHANGES ACCUMULATED=NO  COMPLETE CA=YES INDUBT EEQES=NO
LSN = 000000000000  DSSN = 0000000000
LRID = 0000000000000000  USID = 0000000000
DBD=DEDBJN21  DDN=DB21AR7  PURGETIME = 04.196 09:18:36.0
CHANGES ACCUMULATED=NO  COMPLETE CA=YES INDUBT EEQES=NO
LSN = 000000000000  DSSN = 0000000000
LRID = 0000000000000000  USID = 0000000000

Figure 82. Sample Listing of a RECON at the Tracking Site - CAGRP and CA Records
Sample RECON Listing Tracking Site

DBDSGRP Records

<table>
<thead>
<tr>
<th>DBGRP</th>
<th>GRPNAME=DBGRP1</th>
<th>#MEMBERS=6</th>
<th>-DBD/AREA-</th>
<th>DIVNTZ02</th>
<th>DHVNTZ02</th>
<th>DXVNTZ02</th>
<th>DB21AR0</th>
<th>DB21AR1</th>
<th>DB21AR2</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DBDSGRP</th>
<th>GRPNAME=FJKGRP</th>
<th>#MEMBERS=5</th>
<th>-DBD/AREA-</th>
<th>DIVNTZ02</th>
<th>DHVNTZ02</th>
<th>DXVNTZ02</th>
<th>DBHVSAM1</th>
<th>DBHVSAM2</th>
<th>HIDAM</th>
<th>HIDAM2</th>
<th>XDLBT04I</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>RECOVGRP</th>
<th>GRPNAME=RCVGRP1</th>
<th>#MEMBERS=5</th>
<th>-DBD/AREA-</th>
<th>DIVNTZ02</th>
<th>DHVNTZ02</th>
<th>DXVNTZ02</th>
<th>DEDBJN21</th>
<th>DB21AR0</th>
<th>DB21AR1</th>
</tr>
</thead>
</table>

Figure 83. Sample Listing of a RECON at the Tracking Site - DBDSGRP Records
Sample RECON Listing Tracking Site

DB (IMS) and Related Records

04.198 08:57:55.5  LISTING OF RECON  PAGE 0002

DB
DBD=DBOHIDK5  IRLMID=*NULL  DMB#=2  TYPE=IMS
SHARE LEVEL=3  GSGNAME=**NULL**  USID=0000000001
AUTHORIZED USID=0000000000  RECEIVE USID=0000000000  HARD USID=0000000000
RECEIVE NEEDED USID=0000000000
DBRCVGRP=**NULL**
FLAGS:  COUNTERS:
BACKOUT NEEDED =OFF  RECOVERY NEEDED COUNT =0
READ ONLY =OFF  IMAGE COPY NEEDED COUNT =0
PROHIBIT AUTHORIZATION=OFF  AUTHORIZED SUBSYSTEMS =0
RECOVERABLE =YES  HELD AUTHORIZATION STATE=0
EEQE COUNT =0
TRACKING SUSPENDED =NO  RECEIVE REQUIRED COUNT =0
OFR REQUIRED =NO

04.198 08:57:55.5  LISTING OF RECON  PAGE 0003

DBDS
DSN=DBOHIDK5.CKXXI01O  TYPE=IMS
DBD=DBOHIDK5  DDN=CKOHIG1O  DSID=001  DBORG=HIDAM  DSORG=OSAM
CAGRP=CAGRP2  GENMAX=2  IC AVAIL=0  IC USED=0  DSN=00000000
NOREUSE  RECOVPD=0  OTHER DDN=**NULL**
DEFAULT=**NULL**  ICJCL=ICJCL  OICJCL=OICJCL  RECOVJCL=RECOVJCL
RECVJCL=ICRCVJCL
FLAGS:  COUNTERS:
IC NEEDED =OFF
IC RECOMMENDED =ON
RECOV NEEDED =OFF
RECEIVE NEEDED =OFF  EEQE COUNT =0

DSP0180I NUMBER OF RECORDS LISTED IS  2
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I COMMAND COMPLETION TIME 04.198 08:57:56.2
IMS VERSION 9 RELEASE 1  DATA BASE RECOVERY CONTROL  PAGE 0004
LIST.DB DBD(DHVNTZ02) DBDS

04.198 08:57:55.5  LISTING OF RECON  PAGE 0005

DB
DBD=DHVNTZ02  IRLMID=*NULL  DMB#=5  TYPE=IMS
SHARE LEVEL=3  GSGNAME=IMSGSG1  USID=0000000003
AUTHORIZED USID=0000000000  RECEIVE USID=0000000001  HARD USID=0000000000
RECEIVE NEEDED USID=0000000000
DBRCVGRP=RCVGRP1
FLAGS:  COUNTERS:
BACKOUT NEEDED =ON  RECOVERY NEEDED COUNT =0
READ ONLY =OFF  IMAGE COPY NEEDED COUNT =0
PROHIBIT AUTHORIZATION=OFF  AUTHORIZED SUBSYSTEMS =1
RECOVERABLE =YES  HELD AUTHORIZATION STATE=0
DATABASE LEVEL TRACK =YES  EEQE COUNT =0
TRACKING SUSPENDED =NO  RECEIVE REQUIRED COUNT =0
OFR REQUIRED =NO

ASSOCIATED SUBSYSTEM INFORMATION:
ENCODER  B/O NEEDED
-SSID-  -ACCESS INTENT-  -STATE-  -COUNT-  -SS ROLE-
SYS3  EXCLUSIVE  7  1  TRACKING

Figure 84. Sample Listing of a RECON at the Tracking Site - DB (IMS) and Related Records (Part 1 of 2)
Figure 84. Sample Listing of a RECON at the Tracking Site - DB (IMS) and Related Records (Part 2 of 2)
Sample RECON Listing Tracking Site

DB (HALDB and PART) and Related Records

03.287 13:23:51.7  LISTING OF RECON  PAGE 0013

<table>
<thead>
<tr>
<th>DB</th>
<th>DBD=DBHIDJ03</th>
<th>DMB#=6</th>
<th>CHANGE#=5</th>
<th>TYPE=HALDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHARE LEVEL=3</td>
<td>GSGNAME=IMSGSG1</td>
<td>PSNAME=<strong>NULL</strong></td>
<td>DBORG=PHIDAM</td>
<td>DSORG=OSAM</td>
</tr>
<tr>
<td>FLAGS:</td>
<td>COUNTERS:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECOVERABLE</td>
<td>=YES</td>
<td>PARTITIONS</td>
<td>=4</td>
<td></td>
</tr>
<tr>
<td>ONLINE REORG CAPABLE</td>
<td>=YES</td>
<td>DATA SET GROUP MEMBERS</td>
<td>=2</td>
<td></td>
</tr>
<tr>
<td>REORG INTENT</td>
<td>=OFF</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

03.287 13:23:51.7  LISTING OF RECON  PAGE 0014

<table>
<thead>
<tr>
<th>DB</th>
<th>DBD=POHIDJA</th>
<th>MASTER DB=DBHIDJ03</th>
<th>IRLMID=<strong>NULL</strong></th>
<th>CHANGE#=2</th>
<th>TYPE=PART</th>
</tr>
</thead>
<tbody>
<tr>
<td>USID=0000000003</td>
<td>AUTHORIZED USID=0000000003</td>
<td>HARD USID=0000000003</td>
<td>RECEIVE USID=0000000003</td>
<td>RECEIVE NEEDED USID=0000000000</td>
<td></td>
</tr>
<tr>
<td>DBRCVGRP=<strong>NULL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSN PREFIX=IMSTESTS.DBHIDJ03</td>
<td>PARTITION ID=00001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PREVIOUS PARTITION=<strong>NULL</strong></td>
<td>NEXT PARTITION=POHIDJB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REORG#=00001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OLRMISD=<strong>NULL</strong></td>
<td>ACTIVE DBDS=A-J</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FREE BLOCK FREQ FACTOR=0  FREE SPACE PERCENTAGE=0

PARTITION HIGH KEY/STRING (CHAR):  (LENGTH=5 ) J0200
PARTITION HIGH KEY/STRING (HEX):  D1F0F2F0F0404040404040404040404040404040404040404040404040404040

OSAM BLOCK SIZE:
A = 4096
B = 4096

FLGS:
BACKOUT NEEDED =OFF
READ ONLY =OFF
PROHIBIT AUTHORIZATION=OFF
DATABASE LEVEL TRACK =YES
TRACKING SUSPENDED =NO
OFR REQUIRED =NO
PARTITION INIT NEEDED =NO
OLREORG CURSOR ACTIVE =NO
PARTITION DISABLED =NO
ONLINE REORG CAPABLE =YES

COUNTERS:
RECOVERY NEEDED COUNT =0
IMAGE COPY NEEDED COUNT =0
AUTHORIZED SUBSYSTEMS =1
HELD AUTHORIZATION STATE=0
EEQE COUNT =0
RECEIVE REQUIRED COUNT =0
OLR ACTIVE HARD COUNT =0
OLR INACTIVE HARD COUNT =0

ASSOCIATED SUBSYSTEM INFORMATION:
-SSID- -ACCESS INTENT- -STATE- -COUNT- -SS ROLE-
IMS2 EXCLUSIVE 7 0 TRACKING

Figure 85. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 1 of 11)
Sample RECON Listing Tracking Site

---

03.287 13:23:51.7 LISTING OF RECON PAGE 0015
---

DBDS

DBS=IMSTESTS.DBHIDJ03.A00001
TYPE=PART
DBD=POHIDJA DDN=POHIDJAA DSID=001 DBORG=HIDAM DSORG=OSAM
CAGRP=**NULL** GENMAX=2 IC AVAIL=0 IC USED=1 DSSN=000000002
NOCUSE RECOVPD=0 OTHER DDN=**NULL**
DEFLTJCL=**NULL** ICJCL=PICJCL OICJCL=POICJCL RECOVJCL=PRECOJCL
RECIVJCL=PRECVJCL

FLAGS:
COUNTERS:
IC NEEDED =OFF
RECOV NEEDED =OFF
RECEIVE NEEDED =OFF
EEQE COUNT =0
---

ALLOC

ALLOC =03.287 10:49:51.9 * ALLOC LRID =000000000000000C
DSSN=0000000001 USID=0000000002 START = 03.287 10:49:50.0

ALLOC =03.287 10:55:40.2 * ALLOC LRID =0000000000000012
DSSN=0000000002 USID=0000000003 START = 03.287 10:55:36.8

IMAGE

RUN = 03.287 11:22:19.0 * RECORD COUNT =0
STOP = 00.000 00:00:00.0 BATCH USID=0000000003

IC1

DSN=IMSVS.POHIJA.POHIJAA.IC.IC112153 FILE SEQ=0001
UNIT=SYSDA VOLS DEF=0001 VOLS USED=0001
VOLSER=222222

REORG

RUN = 03.287 10:49:13.7 * USID = 0000000001

RECOV

RUN = 03.287 11:47:59.9 * RUN USID = 0000000000

---

*Figure 85. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 2 of 11)*

---

518 DBRC Guide and Reference
Figure 85. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 3 of 11)
Sample RECON Listing Tracking Site

Figure 85. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 4 of 11)
Sample RECON Listing Tracking Site

03.287 13:23:51.7 LISTING OF RECON PAGE 0020

DBDS
DSN=IMSTESTS.DBHIDJ03.A00002 TYPE=PART
DBD=POHIDJB DDN=POHIDJBA DSID=001 DBORG=HIDAM DSORG=OSAM
CAGRP=**NULL** GENMAX=2 IC AVAIL=0 IC USED=1 DSSN=00000002
NOREUSE RECOVPD=0 OTHER DDN=**NULL**
DEFLTJCL=**NULL** ICJCL=PICJCL OICJCL=POICJCL RECOVJCL=PRECOJCL
RECVJCL=PRECVJCL
FLAGS: COUNTERS:
   IC NEEDED =OFF
   RECOV NEEDED =OFF
   RECEIVE NEEDED =OFF   EEQE COUNT =0

ALLOC
ALLOC =03.287 10:49:56.6 * ALLOC LRID =0000000000000C62
DSSN=0000000001 USID=0000000002 START = 03.287 10:49:50.0

ALLOC
ALLOC =03.287 10:55:44.5 * ALLOC LRID =0000000000000992
DSSN=0000000002 USID=0000000003 START = 03.287 10:55:36.8

IMAGE
RUN = 03.287 11:23:42.4 * RECORD COUNT =0
STOP = 00.000 00:00:00.0 BATCH USID=0000000003

IC1
DSN=IMSVS.POHDJB.POHDJBA.IC.IC12155 FILE SEQ=0001
UNIT=SYSDA VOLS DEF=0001 VOLS USED=0001
VOLSER=222222

REORG
RUN = 03.287 10:49:14.0 * USID = 0000000001

RECOV
RUN = 03.287 12:01:41.5 * RUN USID = 0000000000

Figure 85. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 5 of 11)
Sample RECON Listing Tracking Site

03.287 13:23:51.7  LISTING OF RECON  PAGE 0021

--- DBDS ---

DSN=IMSTESTS.DBHIDJ03.B00002  TYPE=PART
DBD=POHIDJB  DDN=POHIDJBB DSID=006 DBORG=HIDAM DSORG=OSAM
CAGRP=**NULL**  GENMAX=2  IC AVAIL=0  IC USED=1  DSN=000000002
NOREUSE  RECOVPD=0  OTHER DDN=**NULL**
DEFLTJCL=**NULL**  ICJCL=PICJCL  OICJCL=POICJCL  RECOVJCL=PRECOJCL
RECVJCL=PRECVJCL
FLAGS:  COUNTERS:
        IC NEEDED =OFF
        RECOV NEEDED =OFF
        RECEIVE NEEDED =OFF
        EEQE COUNT =0

--- ALLOC ---

ALLOC  =03.287 10:49:55.4  * ALLOC LRID =0000000000000C2A
DSN=0000000001 USID=0000000002 START = 03.287 10:49:50.0
ALLOC  =03.287 10:55:44.4  * ALLOC LRID =000000000000092E
DSN=0000000002 USID=0000000003 START = 03.287 10:55:36.8

--- IMAGE ---

IMAGE  = 03.287 11:24:19.6  * RECORD COUNT =0
STOP = 00.000 00:00:00.0  BATCH USID=0000000003

--- ICI ---

ICI  DSN=IMSVS.POHIDJB.POHIDJBB.IC.IC112155  FILE SEQ=0001
UNIT=SYSDA  VOLS DEF=0001 VOLS USED=0001
VOLSER=222222

--- REORG ---

REORG  = 03.287 10:49:14.0  * USID = 0000000001
RECOV  = 03.287 12:10:36.0  * RUN USID = 0000000000

03.287 13:23:51.7  LISTING OF RECON  PAGE 0022

--- DBDS ---

DSN=IMSTESTS.DBHIDJ03.L00002  TYPE=PART
DBD=POHIDJB  DDN=POHIDJBL DSID=003 DBORG=INDEX DSORG=VSAM
OTHER DDN=**NULL**
FLAGS:  COUNTERS:
        RECOV NEEDED =OFF
        EEQE COUNT =0

--- RECOV ---

03.287 13:23:51.7  LISTING OF RECON  PAGE 0023

--- DBDS ---

DSN=IMSTESTS.DBHIDJ03.X00002  TYPE=PART
DBD=POHIDJB  DDN=POHIDJBX DSID=005 DBORG=INDEX DSORG=VSAM
OTHER DDN=**NULL**
FLAGS:  COUNTERS:
        RECOV NEEDED =OFF
        EEQE COUNT =0

--- ALLOC ---

03.287 13:23:51.7  LISTING OF RECON  PAGE 0024

--- DBDS ---

DSN=IMSTESTS.DBHIDJ03.L00002  TYPE=PART
DBD=POHIDJB  DDN=POHIDJBL DSID=003 DBORG=INDEX DSORG=VSAM
OTHER DDN=**NULL**
FLAGS:  COUNTERS:
        RECOV NEEDED =OFF
        EEQE COUNT =0

--- ALLOC ---

03.287 13:23:51.7  LISTING OF RECON  PAGE 0025

Figure 85. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 6 of 11)

522  DBRC Guide and Reference
List 8. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 7 of 11)
Sample RECON Listing Tracking Site

Figure 85. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 8 of 11)
Figure 85. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 9 of 11)
Sample RECON Listing Tracking Site

03.287 13:23:51.7 LISTING OF RECON PAGE 0030

DBDS
DSN=IMSTESTS.DBHIDJ03.A00004 TYPE=PART
DBD=POHIDJD DDN=POHIDJD030004 DBORG=HIDAM DSORG=OSAM
CAGRP=**NULL** GENMAX=2 IC AVAIL=0 IC USED=1 DSN=00000000
NORUSE RECOVPD=0 OTHER DDN=**NULL**
DEFLTJCL=**NULL** ICJCL=PICJCL OICJCL=POICJCL RECOVJCL=PRECOJCL
RECVJCL=PRECVJCL
FLAGS: COUNTERS:
   IC NEEDED =OFF
   RECOV NEEDED =OFF
   RECEIVE NEEDED =OFF EEQE COUNT =0

IMAGE
RUN = 03.287 11:26:12.4 * RECORD COUNT =0
STOP = 00.000 00:00:00.0 BATCH USID=0000000001

IC1
DSN=IMSVS.POHIDJD.POHIDJD0300001 IC.IC112157 FILE SEQ=0001
UNIT=SYSDA VOLS DEF=0001 VOLS USED=0001
VOLSER=222222

RECOV
RUN = 03.287 12:26:02.7 * RUN USID = 0000000000
03.287 13:23:51.7 LISTING OF RECON PAGE 0031

DBDS
DSN=IMSTESTS.DBHIDJ03.B00004 TYPE=PART
DBD=POHIDJD DDN=POHIDJD030006 DBORG=HIDAM DSORG=OSAM
CAGRP=**NULL** GENMAX=2 IC AVAIL=0 IC USED=1 DSN=00000000
NORUSE RECOVPD=0 OTHER DDN=**NULL**
DEFLTJCL=**NULL** ICJCL=PICJCL OICJCL=POICJCL RECOVJCL=PRECOJCL
RECVJCL=PRECVJCL
FLAGS: COUNTERS:
   IC NEEDED =OFF
   RECOV NEEDED =OFF
   RECEIVE NEEDED =OFF EEQE COUNT =0

IMAGE
RUN = 03.287 11:26:54.2 * RECORD COUNT =0
STOP = 00.000 00:00:00.0 BATCH USID=0000000001

IC1
DSN=IMSVS.POHIDJD.POHIDJD0300001 IC.IC112157 FILE SEQ=0001
UNIT=SYSDA VOLS DEF=0001 VOLS USED=0001
VOLSER=222222

RECOV
RUN = 03.287 12:30:52.7 * RUN USID = 0000000000

Figure 85. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 10 of 11)
Sample RECON Listing Tracking Site

03.287 13:23:51.7  LISTING OF RECON  PAGE 0032
-----------------------------------------------------------------------------------------------------
DBDS
DSN=IMSTESTS.DBHIDJ03.L00004  TYPE=PART
DBD=POHIDJD  DDN=POHIDJDL DSID=003 DBORG=INDEX DSORG=VSAM
OTHER DDN=**NULL**
FLAGS:
  COUNTERS:
    RECOV NEEDED =OFF
    EEQE COUNT =0
-----------------------------------------------------------------------------------------------------
03.287 13:23:51.7  LISTING OF RECON  PAGE 0033
-----------------------------------------------------------------------------------------------------
DBDS
DSN=IMSTESTS.DBHIDJ03.X00004  TYPE=PART
DBD=POHIDJD  DDN=POHIDJDX DSID=005 DBORG=INDEX DSORG=VSAM
OTHER DDN=**NULL**
FLAGS:
  COUNTERS:
    RECOV NEEDED =OFF
    EEQE COUNT =0
-----------------------------------------------------------------------------------------------------

Figure 85. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 11 of 11)
### Sample RECON Listing Tracking Site

#### DB (FP) and Related Records

```
LIST.DB DBD(DEDBDD01) DBDS
04.197 11:48:53.5 LISTING OF RECON PAGE 0002

DB
DBD=DEDBDD01 DMB#=7 TYPE=FP
SHARE LEVEL=1
FLAGS:
    COUNTERS:
        RECOVERY NEEDED COUNT = 0
        IMAGE COPY NEEDED COUNT = 0
PROHIBIT AUTHORIZATION=OFF AUTHORIZED AREAS = 0
RECOVERABLE =YES EEQE COUNT = 0

04.197 11:48:53.5 LISTING OF RECON PAGE 0003

DBDS
DBD=DEDBDD01 AREA=DD01AR0 TYPE=FP
SHARE LEVEL=1 DSID=00001 DBORG=DEDB DSORG=VSAM
GSGNAME=IMSGSG1 USID=0000000003
AUTHORIZED USID=0000000000 RECEIVE USID=0000000000 HARD USID=0000000000
RECEIVE NEEDED USID=0000000000
CAGRP=**NULL** GENMNX=2 IC AVAIL=0 IC USED=0 DSSN=00000002
NOREUSE RECOVPD=0 NOVSO PREOPEN NOPRELOAD
CFSTR1=**NULL** CFSTR2=**NULL** NOLKASID NOMAS
DEFLTJCL=**NULL** ICJCL=ICJCL RECVJCL=ICRCVJCL RECOVJCL=RECOVJCL
DBRCVGPR=**NULL**
FLAGS:
    COUNTERS:
        PROHIBIT AUTHORIZATION=OFF HELD AUTHORIZATION STATE=0
        IC NEEDED=OFF ADs AVAIL # = 1
        IC RECOMMENDED=ON
        RECOV NEEDED=OFF REGISTERED ADS # = 1
        DATABASE LEVEL TRACK=YES EEQE COUNT = 0
        RECEIVE NEEDED=OFF
        OFR REQUIRED=NO
        TRACKING SUSPENDED=NO
        HSSP CIC IN PROGRESS = NO
ADS LIST:
    -ADS DDN--ADS DSN-
    -DDO1AR0 DDO1AR0

<table>
<thead>
<tr>
<th>ALLOC</th>
<th>04.196 09:48:51.8</th>
<th>* ALLOC LRID =00000000000000FB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSSN</td>
<td>00000000001 USID=0000000002</td>
<td>START = 04.196 09:48:49.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ALLOC</th>
<th>04.196 08:51:31.6</th>
<th>* ALLOC LRID =00000000000003B6</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSSN</td>
<td>00000000002 USID=0000000003</td>
<td>START = 04.196 08:51:28.5</td>
</tr>
</tbody>
</table>
```

Figure 86. Sample Listing of a RECON at the Tracking Site - DB (FP) and Related Records

### Fields Present in a Listing of a RECON by Record Type

The following sections describe the fields that can be present in a listing of the RECON by record type:

- "Fields Present in a RECON Record" on page 529
- "Fields Present in a Log Record" on page 532
- "Fields Present in a LOGALL Record" on page 533
- "Fields Present in an Online Log Record" on page 533
Fields in a RECON Listing

- “Fields Present in a GSG Record” on page 537
- “Fields Present in a SSYS Record” on page 537
- “Fields Present in a BACKOUT Record” on page 538
- “Fields Present in a CAGRP Record” on page 539
- “Fields Present in a CA Record” on page 540
- “Fields Present in a Data Group Record” on page 542
- “Fields Present in a DB (IMS) Record” on page 542
- “Fields Present in a DB (HALDB) Record” on page 544
- “Fields Present in a DB (PART) Record” on page 545
- “Fields Present in a DB (Fast Path) Record” on page 549
- “Fields Present in a DBDS (non-Fast Path) Record” on page 550
- “Fields Present in a DBDS (Fast Path) Record” on page 552
- “Fields Present in an ALLOC Record” on page 555
- “Fields Present in an IMAGE Record” on page 556
- “Fields Present in a REORG Record” on page 558
- “Fields Present in a RECOV Record” on page 558

Examples of many of these fields appear in Figure 67 on page 476 and Figure 77 on page 504.

Definition:

A group of lines refers to the lines following the referring statement whose row spans the width of the table. The group of lines ends at the next statement that spans the width of the table, unless otherwise specified.

Fields Present in a RECON Record

A RECON record’s fields and their corresponding line numbers are described in Table 37.

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECON</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>RECOVERY CONTROL DATA SET, IMS VxRx</td>
<td>Identifies the version and release of the RECON.</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>LAST USED DMB#= nnn</td>
<td>nnn represents the DMB number last assigned, if less than 32 767. If the DMB number has reached 32 767, nnn represents the last reused DMB. Only printed if greater than zero.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INIT TOKEN= token</td>
<td>The token assigned to this RECON when it was initialized.</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>FORCER</td>
<td>Indicates whether databases must be registered in RECON. FORCER indicates that all databases must be registered. NOFORCER indicates that all databases do not need to be registered.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOG DSN CHECK= xxxxxx</td>
<td>The type of log data set name checking; xxxxxx is CHECK17, CHECK44, or NOCHECK.</td>
</tr>
</tbody>
</table>
### Fields in a RECON Record

**Table 37. Fields Present in the RECON Record (continued)**

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>**STARTNEW= YES</td>
<td>NO**</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td><strong>TAPE UNIT= unittype</strong></td>
<td>The default unit type for log data sets, NOREUSE image copy data sets, and NOREUSE change accumulation data sets that reside on tape devices.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>DASD UNIT= unittype</strong></td>
<td>The default unit type for log data sets, NOREUSE image copy data sets, and NOREUSE change accumulation data sets that reside on DASD devices.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>**TRACEON</td>
<td>TRACEOFF**</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>SSID= xxxxxxxxx</strong></td>
<td>The IMS subsystem default name.</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>**LISTDLOG= YES</td>
<td>NO**</td>
</tr>
<tr>
<td></td>
<td></td>
<td>**CA</td>
<td>IC</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>MINIMUM VERSION = x.x</strong></td>
<td>Indicates the minimum version of IMS that is allowed to sign on using these RECONs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>CROSS DBRC SERVICE LEVEL ID=nnnn</strong></td>
<td>The current enabled service level ID as set by the CHANGE.RECON CDSLID() command.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>LOG RETENTION PERIOD= yy.ddd  hh:mm:ss.t</strong></td>
<td>Indicates the minimum amount of time that DBRC is to keep log records in RECON.</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>**COMMAND AUTH=SAF</td>
<td>EXIT</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>HLQ=hlq name</strong></td>
<td>High level qualifier of the resource name used to determine command authorization. <strong>NULL</strong> is printed when command authorization is not enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TRACE OPTIONS= options</strong></td>
<td>This line is printed only if an IBM representative has instructed you to gather documentation for problem analysis. It indicates that an IBM representative has provided instructions to enable DBRC trace records to be written to the DBRC trace table for problem analysis. The IBM representative will have provided the options.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>SIZALERT= xxxxxxxxx xxxxxxxxx xx</strong></td>
<td>The (optional) thresholds that trigger messages to warn you that a record has grown unnaturally large. Field values can be dsnum, volnum and percent.</td>
</tr>
</tbody>
</table>
### Table 37. Fields Present in the RECON Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>11</td>
<td>LOGALERT=xxxxxxxx xxxxxxxx</td>
<td>The (optional) threshold that triggers the DSP0287W message (warning you when you just have time to shut down an online IMS subsystem before it ABENDs due to the PRILOG record exceeding RECORDSIZE). Field values can be dsnum, volnum and percent.</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>REORG NUMBER VERIFICATION=YES/NO</td>
<td>Identifies if reorganization number verification is performed for partition databases.</td>
</tr>
</tbody>
</table>

The following line is printed only if a failure occurred during a multiple update to the RECON:

<table>
<thead>
<tr>
<th>13</th>
<th>13</th>
<th>UPDATE TYPE=nnnn</th>
<th>The type of multiple update that was in progress.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>13</td>
<td>DBID=xxxxxxxx xxxxxxxx</td>
<td>Data set name and data set ddname of the DBDS involved in the multiple update. Can be blank.</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>CAGRP=xxxxxxxxx</td>
<td>Name of the CA group involved in the multiple update. Contains blanks if no group was involved.</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>NEW DDN=xxxxxxxxx</td>
<td>New ddname, if any, that was associated with the DBDS that was involved in the multiple update. Contains blanks if no new name or DBDS was involved.</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>OLD RECORD KEY</td>
<td>This field is printed only if the multiple update involves a key change.</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>KEY TYPE= *xxxxx</td>
<td>A description of the record type. The asterisk (*) is printed only if the record is available for future use (it is either new or scheduled for reuse). Whenever an unrecognizable record key type is found, BADTYP:KEY is printed, together with the key in hexadecimal characters.</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>DBD=xxxxxxxx, DDN= xxxxxxxx</td>
<td>The database name and data set ddname of a DBDS. If the record is PRILOG, SECLOG, IPRI, or ISEC, the fields are printed as <strong>NULL</strong>. If the record is a CAGRP or DB record, the DBD field is <strong>NULL</strong> and the DDN field contains the name of the CA group or database.</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>TIME= time_stamp</td>
<td>The time stamp of the key of the record.</td>
</tr>
<tr>
<td>16-17</td>
<td>16-17</td>
<td>NEW RECORD KEY</td>
<td>This field is printed only if a new record key is being added. Additional fields in this part of the record are printed as shown in line 8.</td>
</tr>
<tr>
<td>18-19</td>
<td>18-19</td>
<td>BASE RECORD KEY</td>
<td>This field is printed only if a new record key is being changed. The remaining fields in this part of the record are printed as shown in line 8.</td>
</tr>
<tr>
<td>20-21</td>
<td>20-21</td>
<td>SSID= ssidname IRLMID= irmidname SHARE LVL #= n #DB= n B/O#= n</td>
<td>Information about the multiple update.</td>
</tr>
<tr>
<td>22</td>
<td>22</td>
<td>FLAGS:</td>
<td>A heading for the following lines.</td>
</tr>
</tbody>
</table>
### Fields in a RECON Record

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>23</td>
<td>RECOV= ON</td>
<td>Information about the multiple update.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NORECOV= ON</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ICON= ON</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ICOFF= ON</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>NORMAL= ON</td>
<td>Information about the multiple update.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABNORMAL= ON</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>STARTRCV= ON</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ENDRECOV= ON</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>READON= ON</td>
<td>Information about the multiple update.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>READOFF= ON</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUTH= ON</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOAUTH= ON</td>
<td></td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>B/O DONE= ON</td>
<td>Information about the multiple update.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PASS 1= ON</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BACKOUT= ON</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SHARELVL= ON</td>
<td></td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>TIME STAMP INFORMATION:</td>
<td>Heading for section containing RECON time stamp information.</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>TIMEZIN= offset</td>
<td>User-specified input time stamp offset default. If no offset is specified, this field displays %SYS meaning that the offset of the z/OS clock is to be used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-LABEL- -OFFSET-</td>
<td>These column headings are printed if a Time Zone Label Table has been defined.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>label offset</td>
<td>The list of defined time zone offset labels with their corresponding offset values.</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>OUTPUT FORMAT:</td>
<td>Time-stamp output format settings.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DEFAULT= offset display form yearsize</td>
<td>Default time-stamp output format settings.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CURRENT= offset display form yearsize</td>
<td>Current time-stamp output format settings.</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>IMSPLEX=imsplex_name</td>
<td>The IMSplex name associated with the RECON.</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>-DDNAME- -STATUS- -DATA SET™ NAME-</td>
<td>Column headings for the following one to three lines.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECON1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECON2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECON3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>COPY1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>data name</td>
<td>The ddname, status, and data set name of the Copy1 RECON.</td>
</tr>
<tr>
<td></td>
<td>35-36</td>
<td>RECON1</td>
<td>These lines identify the ddname, status, and data set name of the RECON backup (Copy2), the RECON spare, and any RECONs that are not usable (DISCARDED). A RECON that could not be accessed is shown as UNAVAILABLE (unknown dsname and status).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECON2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECON3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>COPY2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DISCARDED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SPARE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UNAVAILABLE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>dsname or blank</td>
<td></td>
</tr>
</tbody>
</table>

### Fields Present in a Log Record

A log record’s fields, including types, and their corresponding line numbers are described in Table 38 on page 533.
### Table 38. Fields Present in a Log Record

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRILOG or SECLOG or PRISLD or SECSLD or IPRI or ISEC or IPRISL or ISECSL or PRITSLDS or SECTSLDS or IPRITSLD or ISECTSLD</td>
<td>1</td>
<td>RECORD SIZE=nnnnnn</td>
<td>The logical record size for PRILOG, SECLOG, PRISLD, SECSLD, PRITSLDS, and SECTSLDS records. This field is not present for other log record types.</td>
</tr>
<tr>
<td>PRILOG or SECLOG or PRISLD or SECSLD or IPRI or ISEC or IPRISL or ISECSL or PRITSLDS or SECTSLDS or IPRITSLD or ISECTSLD</td>
<td>2</td>
<td>START= time_stamp*</td>
<td>Time stamp of the start time (that is, original open time) of the log data set. An asterisk (*) indicates that this time stamp is part of the record key.</td>
</tr>
<tr>
<td>PRILOG or SECLOG or PRISLD or SECSLD or IPRI or ISEC or IPRISL or ISECSL or PRITSLDS or SECTSLDS or IPRITSLD or ISECTSLD</td>
<td>3</td>
<td>SSID= ssidname</td>
<td>The name of the IMS subsystem.</td>
</tr>
<tr>
<td>PRILOG or SECLOG or PRISLD or SECSLD or IPRI or ISEC or IPRISL or ISECSL or PRITSLDS or SECTSLDS or IPRITSLD or ISECTSLD</td>
<td>4</td>
<td>VERSION=version</td>
<td>The version of the IMS subsystem that created the log.</td>
</tr>
<tr>
<td>PRILOG or SECLOG or PRISLD or SECSLD or IPRI or ISEC or IPRISL or ISECSL or PRITSLDS or SECTSLDS or IPRITSLD or ISECTSLD</td>
<td>5</td>
<td>STOP= time_stamp</td>
<td>Time stamp of the stop time (that is, close time) of the log data set. Zeros indicate that the data set is still open.</td>
</tr>
<tr>
<td>PRILOG or SECLOG or PRISLD or SECSLD or IPRI or ISEC or IPRISL or ISECSL or PRITSLDS or SECTSLDS or IPRITSLD or ISECTSLD</td>
<td>6</td>
<td>#DSN= nn</td>
<td>The number of data sets in the log data set. A value of zero indicates that no data set has been created.</td>
</tr>
<tr>
<td>PRILOG or SECLOG or PRISLD or SECSLD or IPRI or ISEC or IPRISL or ISECSL or PRITSLDS or SECTSLDS or IPRITSLD or ISECTSLD</td>
<td>7</td>
<td>GSGNAME= gsgname</td>
<td>Identifies the name of the GSG to which the subsystem producing this log belongs.</td>
</tr>
</tbody>
</table>

The following 4 fields are printed if the condition in the contents column is true:

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRILOG or SECLOG or PRISLD or SECSLD or IPRI or ISEC or IPRISL or ISECSL or PRITSLDS or SECTSLDS or IPRITSLD or ISECTSLD</td>
<td>8</td>
<td>TRACKING</td>
<td>This log data set was originally created by an active IMS subsystem of the nonlocal service group and transported to the tracking site.</td>
</tr>
<tr>
<td>PRILOG or SECLOG or PRISLD or SECSLD or IPRI or ISEC or IPRISL or ISECSL or PRITSLDS or SECTSLDS or IPRITSLD or ISECTSLD</td>
<td>9</td>
<td>GAP</td>
<td>There is a gap in the log data sets of this record.</td>
</tr>
<tr>
<td>PRILOG or SECLOG or PRISLD or SECSLD or IPRI or ISEC or IPRISL or ISECSL or PRITSLDS or SECTSLDS or IPRITSLD or ISECTSLD</td>
<td>10</td>
<td>PREV-GAP</td>
<td>There is a gap in a previous log record of the same global service group.</td>
</tr>
<tr>
<td>PRILOG or SECLOG or PRISLD or SECSLD or IPRI or ISEC or IPRISL or ISECSL or PRITSLDS or SECTSLDS or IPRITSLD or ISECTSLD</td>
<td>11</td>
<td>BBO</td>
<td>Identifies the log record as it was created by batch backout. If the record was not created by batch backout, this field is blank.</td>
</tr>
<tr>
<td>PRILOG or SECLOG or PRISLD or SECSLD or IPRI or ISEC or IPRISL or ISECSL or PRITSLDS or SECTSLDS or IPRITSLD or ISECTSLD</td>
<td>12</td>
<td>FIRST RECORD ID= lsn</td>
<td>The log record sequence number of the first log record that was written during initialization of the IMS subsystem.</td>
</tr>
<tr>
<td>PRILOG or SECLOG or PRISLD or SECSLD or IPRI or ISEC or IPRISL or ISECSL or PRITSLDS or SECTSLDS or IPRITSLD or ISECTSLD</td>
<td>13</td>
<td>PRILOG TOKEN= n</td>
<td>The numeric log token assigned sequentially to PRILOG records for the same GSG.</td>
</tr>
</tbody>
</table>

Rows 7-13 apply to active log records and repeat for each log data set. Rows 7-10 apply to tracking records and repeat for each log data set.
### Fields in Log Records

#### Table 38. Fields Present in a Log Record  (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>DSN= log.dname</td>
<td>The data set name of the log data set described in this record. ****COMPRESSED DATA SET **** indicates that PRILOG compression has removed unneeded DSN entries.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ERR</td>
<td>This field is printed if a previous command was used to indicate that an error exists in the log data set.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UNIT= unittype</td>
<td>Unit type to be used for substitution during the GENJCL process.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>START= time_stamp</td>
<td>Start time of the data set entry in the log data set.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FIRST RECORD LSN= lsn</td>
<td>The log record sequence number of the first log record of the data set.</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>STOP= time_stamp</td>
<td>Stop time of the data set entry in the log data set. Zeros indicate that the data set is still open.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LAST RECORD LSN= lsn</td>
<td>The log record sequence number of the last log record of the data set.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>FILE SEQ= mmmm</td>
<td>The file sequence number of the log data set. This field is not printed for tracking logs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>#VOLUMES= mnn</td>
<td>The number of volumes for each data set entry. This field is not printed for tracking logs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>#DS CHECKPOINTS= n CHKPT ID= time_stamp</td>
<td>Count of checkpoint records in the log data set, and the checkpoint ID value. This field is printed only for tracking logs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CHECKPOINT TYPES=xx</td>
<td>This line is printed only for SLDS records. xx is a hexadecimal representation of the checkpoint types contained on the log. The values for xx can be:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SIMPLE= Y</td>
<td>N</td>
<td>Bit 0</td>
</tr>
<tr>
<td></td>
<td>SNAPQ= Y</td>
<td>N</td>
<td>Bit 1</td>
</tr>
<tr>
<td></td>
<td>DUMPQ= Y</td>
<td>N</td>
<td>Bit 2</td>
</tr>
<tr>
<td></td>
<td>PURGE= Y</td>
<td>N</td>
<td>Bit 3</td>
</tr>
<tr>
<td></td>
<td>FREEZE= Y</td>
<td>N</td>
<td>Bit 4</td>
</tr>
<tr>
<td></td>
<td>Bit 5 - 7</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For more information, see the note that follows this table.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>VOLSER= volser STOPTIME= time_stamp</td>
<td>Volume serial number and stop time. This field is not printed for tracking logs.</td>
<td></td>
</tr>
<tr>
<td>(ACTIVE)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>CKPTCT= n CHKPT ID= time_stamp</td>
<td>Checkpoint count and ID value.</td>
<td></td>
</tr>
<tr>
<td>(ACTIVE)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>LOCK SEQUENCE # =</td>
<td>A unique number, issued sequentially, given to a database resource to identify that it is in locked by a requestor. A locked resource cannot be used until the current requestor has finished using it.</td>
<td></td>
</tr>
<tr>
<td>(ACTIVE)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fields in Log Records

Table 38. Fields Present in a Log Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FEOV</td>
<td>Indicates that the corresponding log data set at the active site was closed by a /DBR with the forced end of volume option. This field is printed only for tracking PRILOGs.</td>
</tr>
</tbody>
</table>

Note: DBRC sets the CHECKPOINT TYPES bit values and also sets the Y or N values. Y is set if at least one checkpoint of the given type starts on the data set. There could be more than one checkpoint on the log and there is no guarantee that there is a complete checkpoint on the log. The number of checkpoints listed on the CHECKPOINT TYPES line might not be the same number that is listed as the CKPTCT value. For example, there could be three simple checkpoints on a log. Only SIMPLE=Y would be displayed and CKPTCT or #DS CHECKPOINTS would have a value of two. CKPTCT or #DS CHECKPOINTS would have a value of two because that is the maximum allowed.

Fields Present in a LOGALL Record

A log allocation record’s fields and their corresponding line numbers are described in Table 39.

Table 39. Fields Present in the LOGALL Record

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGALL</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>START= time_stamp(*)</td>
<td>The start time of the log data set that contains change records for DBDSs included in this record. If the asterisk (*) is present, it indicates that this time stamp also occurs in the key of the LOGALL record.</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>EARLIEST ALLOC TIME=time_stamp</td>
<td>The start time of the oldest DBDS allocation for the log. The ALLOC record with the earliest allocation time is for the DBDS that is listed first in the following lines. If there are no DBDS allocations for the log, zeroes are displayed.</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>DBDS ALLOC= mnn</td>
<td>The number of allocated DBDSs for this log data set.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-DBD- -DDN- -ALLOC-</td>
<td>These headings are printed only if the LOGALL record contained records of each DBDS that was allocated between the start and stop time stamps of the log data set. The DBDSs are listed in order of their earliest ALLOC time, starting with the DBDS with the oldest allocation.</td>
</tr>
<tr>
<td></td>
<td>5-mnn</td>
<td>ddname ddname mnn</td>
<td>The database name, data set ddname, and number of allocations of the DBDS during the time span of the allocated log.</td>
</tr>
</tbody>
</table>

Fields Present in an Online Log Record

An online log record’s fields and their corresponding line numbers are described in Table 40 on page 536.
### Fields in an Online Log Record

**Table 40. Fields Present in an Online Log Record**

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIOLD or SECOLD or I PRIOL or ISECOL</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>SSID= xxxxxxxxx</td>
<td>The name of the IMS subsystem.</td>
</tr>
<tr>
<td></td>
<td></td>
<td># DD ENTRIES= n</td>
<td>The number of ddnames in the OLDS.</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>EARLIEST CHECKPOINT= time_stamp</td>
<td>This line is printed only for PRIOLD records. It indicates the earliest checkpoint required by IMS emergency restart (/ERE or /ERE BU1LDQ).</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>DDNAME= ddname</td>
<td>The ddname of the OLDS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN=log-dsname</td>
<td>The data set name of the OLDS.</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>START= time_stamp</td>
<td>Time stamp of the start time (that is, original open time) of the log data set.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FIRST DS LSN= lsn</td>
<td>The log record sequence number of the first log record of the data set.</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>STOP= time_stamp</td>
<td>Time stamp of the stop time (that is, close time) of the log data set. Zeros indicate that the data set is not closed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LAST DS LSN= lsn</td>
<td>The log record sequence number of the last log record of the data set.</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>STATUS= ARC NEEDED</td>
<td>The status of the DD entry in the OLDS record. This field applies to the PRIOLDS record only.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ARC STARTED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ARC SCHED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ARC COMPLT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACTIVE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ERROR= CLOSE</td>
<td>Indicates a close error on an OLDS or on a previous OLDS record. This field applies to the PRIOLDS record only.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PREV</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FEOV= YES</td>
<td>A specification of FEOV=YES forces an end-of-output data set at the end of archiving DFSOLP02. This is done to conform to the recovery point established by the /DBR command. This field applies to the PRIOLDS record only.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AVAIL</td>
<td>Identifies whether the OLDS is available for regular use. UNAVAIL indicates that the data set has invalid data or an I/O error. This field applies to PRIOLDS or SECOLD records only.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UNAVAIL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>PRILOG TIME= time_stamp</td>
<td>Time stamp of the time of the first OLDS the subsystem used. This is also the start time of the PRILOG and PRISLD records corresponding to the subsystem invocation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ARCHIVE JOB NAME= jobname</td>
<td>Name of the archive job generated by the GENJCL ARCHIVE function.</td>
</tr>
<tr>
<td></td>
<td>blank line</td>
<td></td>
<td>These blank lines are printed as visual separators only when the OLDS has more than one DD entry. The format of each data set entry is a repetition of lines 4 through 8 above.</td>
</tr>
</tbody>
</table>
## Fields Present in a GSG Record

A global service group record’s fields and their corresponding line numbers are described in Table 41.

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSG</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>GSGNAME= gsname</td>
<td>Identifies the name of the GSG.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>#SGS=</td>
<td>The number of service groups in this GSG.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-SGNAME- -ROLE-</td>
<td>Headings for the following lines 3 through 10.</td>
</tr>
<tr>
<td></td>
<td>3-4</td>
<td>snamesgrole</td>
<td>Service group name and service group role (active or tracking).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOCAL</td>
<td>Identifies which service group is the local one of the GSG.</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>CURRENT PRILOG TOKEN= token</td>
<td>The highest PRILOG token at the local site of this GSG.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TAKEOVER TOKEN= token</td>
<td>The current PRILOG token at the time an RSR takeover was initiated at the local site.</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>MINIMUM PRILOG TOKEN= token</td>
<td>The minimum PRILOG that is maintained at the local site of the GSG.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSN SEQ NUMBER= number</td>
<td>The initial DSN sequence number value for the GSG group.</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>START TIME OF CURRENT LOG= time_stamp</td>
<td>The start time of the PRILOG with the current log token.</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>HIGHEST ACTIVE SITE TIME= time_stamp</td>
<td>The highest time received from the active site. This field is only meaningful to the tracking site.</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>TRACKING SUBSYSTEM ID= ssid</td>
<td>Subsystem identification of the tracking subsystem. This field is null at the active site.</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>TAKEOVER IN PROGRESS</td>
<td>This is printed when an RSR takeover has been initiated at the local site.</td>
</tr>
</tbody>
</table>

## Fields Present in a SSYS Record

A subsystem record’s fields and their corresponding line numbers are described in Table 42.

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSYS</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>SSID= ssidname</td>
<td>The name of the IMS subsystem.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOG START= time_stamp</td>
<td>The earliest log data set start time associated with this system or the log data set start time created by batch backout.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSTYPE= ONLINE | BATCH | TRACKER</td>
<td>Indication of whether this is an IMS online, batch, or tracking subsystem.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABNORMAL TERM= ON | OFF</td>
<td>The value of this flag field is normally OFF. If it is ON, the system has been abnormally ended and database recovery is required.</td>
</tr>
</tbody>
</table>
## Fields in a SSYS Record

Table 42. Fields Present in a SSYS Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RECOVERY STARTED=</td>
<td>If the value of this field is YES, this subsystem has signed on for recovery-started processing. Normally, this value is NO.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BACKUP=</td>
<td>If the value of this field is YES, there is an alternate subsystem.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>TRACKED=</td>
<td>Indicates whether this is the record of an active subsystem being tracked by the tracking site.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TRACKER TERM=</td>
<td>Indicates whether the tracking subsystem has terminated. This field only applies to the tracking subsystem record.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SHARING COVERED DBS=</td>
<td>Indicates that this active subsystem is sharing RSR-covered databases.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>IRLMID=</td>
<td>The IRLM with which this subsystem is communicating.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>xxxxxx</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>IRLM STATUS=</td>
<td>Indicates the status of the IRLM. NORMAL indicates no failure. This field does not apply to an RSR tracking subsystem.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NORMAL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>IRLM FAILURE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>COMM FAILURE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYS FAILURE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GSG NAME=</td>
<td>Identifies the name of the global service group to which the subsystem belongs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>gsurname</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>BACKUP IRLMID=</td>
<td>Identifies the IRLM with which the alternate subsystem is communicating. Listed only if an alternate subsystem exists.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>irlmname</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BACKUP TOKEN=</td>
<td>Identifies the backup token. Listed only if an alternate subsystem exists.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>nnn</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>AUTHORIZED DATABASES</td>
<td>Indicates that this subsystem is currently authorized to n databases or areas.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AREAS=</td>
<td>nmm</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>VERSION=</td>
<td>Identifies the IMS release level through which the subsystem signed on.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>n</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XRF CAPABLE=</td>
<td>Identifies if the subsystem is XRF capable.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NO</td>
<td>YES</td>
</tr>
</tbody>
</table>

If the number of authorized databases or areas is not zero, the following lines (up to the BACKOUT record line) list those that are currently authorized:

<table>
<thead>
<tr>
<th>9-10</th>
<th>-DBD-</th>
<th>-AREA-</th>
<th>-LEVEL-</th>
<th>-ACCESSINTENT-</th>
<th>ENCODED</th>
<th>-STATE-</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| 11-nnn | dbname areaname sharelvl access encoded state |
|        | This is the name of the database or area that is currently authorized by this subsystem, the share level, the access intent by this subsystem, and the encoded state. |

### Fields Present in a BACKOUT Record

A backout record's fields and their corresponding line numbers are described in Table 43 on page 539.

---

538 DBRC Guide and Reference
### Fields in a BACKOUT Record

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>BACKOUT</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>SSID= xxxxxxxx</td>
<td>The name of the associated IMS subsystem.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>#UORS= nn</td>
<td>The number of units of recovery (UORs) in the BACKOUT record.</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>RECOVERY TOKEN= 32</td>
<td>Describes a specific UOR.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hexadecimal digits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>TIME= time_stamp</td>
<td>The time stamp of the beginning of the UOR (found in the X'5607' log record).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSB= psbname</td>
<td>Name of the PSB associated with the UOR.</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>CANDIDATE INFLT</td>
<td>INDT BMP COLDEND CMD BATCH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>ASSOCIATED DATABASES= nn</td>
<td>Number of databases associated with the UOR.</td>
</tr>
<tr>
<td></td>
<td>7, 8</td>
<td>BACKED DYN BKOUT -DBD-</td>
<td>Heading for the following list of associated databases.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-OUT-- -FAILURE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9-nnn</td>
<td>dbname YES</td>
<td>NO YES</td>
</tr>
</tbody>
</table>

### Fields Present in a CAGRP Record

A change accumulation group record’s fields and their corresponding line numbers are described in Table 44

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAGRP</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>GRPNAME= cagrpnam</td>
<td>The name of the CA group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GRPMAX= nnnn</td>
<td>The maximum number of change accumulation run records that can be associated with this group, whether available or in use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CA AVAIL= nnnn</td>
<td>The number of available change accumulation run records currently in RECON for this group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CA USED= nnnn</td>
<td>The number of in-use change accumulation run records currently in RECON for this group.</td>
</tr>
</tbody>
</table>
Table 44. Fields Present in a CAGRP Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>REUSE</td>
<td>NOREUSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CAJCL= cajclmem</td>
<td>The name of the member of a partitioned data set that contains skeletal JCL for this CA group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DEFLTJCL= member</td>
<td>The name of the member of a partitioned data set. This member contains the skeletal JCL default values for the user-defined keywords to be used for this CA group.</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>#MEMBERS= nnnn</td>
<td>This number of DBDSs and areas that belong to this CA group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-DBD- -DDN-</td>
<td>These headings and the following lines are printed only if some DBDSs are members of this group.</td>
</tr>
<tr>
<td></td>
<td>5-nnn</td>
<td>dbdname ddname</td>
<td>The database name and data set ddname of a DBDS that is a member of this CA group.</td>
</tr>
</tbody>
</table>

Fields in a CAGRP Record

Fields Present in a CA Record

A change accumulation record's fields and their corresponding line numbers are described in Table 45.

Table 45. Fields Present in a CA Record

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>*DSN= chge.accum.dsnamed</td>
<td>The asterisk is printed if the change accumulation data set identified in this record is available for use. DSN= is the change accumulation data set name that has been or could be used as the output data set during a run of the Database Change Accumulation utility.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FILE SEQ= nnnn</td>
<td>The file sequence number of the first volume of this data set.</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>CAGRP= cagrpname</td>
<td>Name of the CA group to which the change accumulation record belongs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UNIT= unittype</td>
<td>Unit type to be used for substitution during the GENJCL process.</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>CREATE</td>
<td>STOP= time_stamp*</td>
</tr>
</tbody>
</table>
Table 45. Fields Present in a CA Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>VOLS DEF= nnnn VOLS USED= nnnn</td>
<td>The number of volumes that have been specified for use by this change accumulation data set and the number of volumes that were used in a Database Change Accumulation utility run. For available change accumulation data sets, if the data set can be reused, this field contains the same value as the VOLS DEF field. Otherwise, it indicates the number of volumes that were actually used.</td>
</tr>
<tr>
<td>5-nnn</td>
<td></td>
<td>VOLSER= volser1, volser2,...</td>
<td>Volume serial numbers of the volumes on which the change accumulation data set resides or is to reside. Only as many lines as necessary are used to list the volume.</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>RUN= time_stamp</td>
<td>This time stamp represents the time of the run of the Change Accumulation utility during which this record was used.</td>
</tr>
<tr>
<td></td>
<td>ERR</td>
<td></td>
<td>This indicates that you have marked the change accumulation data set as unusable because of a previous error.</td>
</tr>
<tr>
<td></td>
<td>SUBSET</td>
<td></td>
<td>This indicates that a subset of logs were processed when the CA was created.</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td></td>
<td>The asterisk (*) indicates that COMP or SUBSET were specified in the CHANGE.CA or NOTIFY.CA commands.</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>DBD=dbname</td>
<td>The database name and data set ddname of the DBDS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DDN= dbname</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PURGETIME= time_stamp</td>
<td>Change records occurring before this time stamp for the corresponding DBDS have been ignored by the Database Change Accumulation utility.</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>CHANGES ACCUMULATED= YES</td>
<td>YES is printed if the Database Change Accumulation utility run accumulated any changes for the corresponding DBDS. NO is printed if no changes were found for DBDS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COMPLETE CA= YES</td>
<td>YES is printed if the log subset for the corresponding DBDS is complete. NO is printed if the log subset for the DBDS is incomplete.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INDOUBT EEQES= YES</td>
<td>YES is printed to indicate that in doubt EEQEs were accumulated for the corresponding DBDS. NO is printed if no in doubt EEQEs were accumulated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>LSN= lsn</td>
<td>The lock sequence number of the last change accumulated for the DBDS is listed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSSN= dssn</td>
<td>The data set sequence number of the last change accumulated for the DBDS is listed.</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>LRID= log_record_ID</td>
<td>The last log record ID of the last change accumulated for the DBDS is listed in this field.</td>
</tr>
</tbody>
</table>
### Fields in a CA Record

Table 45. Fields Present in a CA Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>USID= update_set_ID</td>
<td>The last update set ID of the last change accumulated for the DBDS is listed in this field.</td>
</tr>
</tbody>
</table>

### Fields Present in a Data Group Record

A database data group record’s fields and their corresponding line numbers are described in [Table 46](#).

Table 46. Fields Present in the DBDSGRP Record

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBDSGRP</td>
<td>1</td>
<td>Type</td>
<td>Type of group: DBDS, DB, or Recovery.</td>
</tr>
<tr>
<td>DBGRP or RECOVGRP</td>
<td></td>
<td>#MEMBERS= nnn</td>
<td>Number of members belonging to the group.</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>GRPNAME= grpname</td>
<td>Name of the group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-DBD- -DDN/AREA-</td>
<td>Any or all of these headings can be printed, depending on the group members that exist. For DBDS groups, the heading is -DBD- -DDN/AREA-. For DB groups, the heading is -DBD/AREA-. For recovery groups, the heading is -DBD- -AREA-.</td>
</tr>
<tr>
<td></td>
<td>3-nnn</td>
<td>dbname ddname/areaname</td>
<td>For DBDSGRP, dbname = DB name, ddname/areaname = DBDS name or area name. For DBGRP, dbname = DL/I DB name or DEDEB area name. ddname/areaname is blank. For RECOVGRP, dbname = DB name, ddname/areaname = area name if DB is a DEDEB, or blank if DB is DL/I.</td>
</tr>
</tbody>
</table>

### Fields Present in a DB (IMS) Record

An IMS database record’s fields and their corresponding line numbers are described in [Table 47](#).

Table 47. Fields Present in the DB (IMS) Record

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB (IMS)</td>
<td>1</td>
<td>DBD= dbname</td>
<td>Name of the database.</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>IRLMID= xxxx</td>
<td>Identifies the IRLM when the share level of this database is 2. This field is printed only when SHARE LEVEL=2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DMB#= nnn</td>
<td>The value assigned when this database was registered.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TYPE= IMS</td>
<td>Indicates that this is a DL/I database (rather than a Fast Path DEDEB).</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>SHARE LEVEL= n</td>
<td>The level of data sharing for which authorized subsystems can share this database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GSGNAME= gsgname</td>
<td>Identifies the name of the GSG to which this database belongs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USID= n</td>
<td>The highest update set identifier for this database.</td>
</tr>
</tbody>
</table>
### Fields in a DB (IMS) Record

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td>AUTHORIZED USID= n</td>
<td>The identifier of the current update set that is being applied to the database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECEIVE USID= n</td>
<td>The update set identifier of the last image copy received for this database. The update set ID only applies to shadow databases at the tracking site.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HARD USID= n</td>
<td>The update set identifier of the latest changes that were written to the database.</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>RECEIVE NEEDED USID= n</td>
<td>This only applies to shadow databases at the tracking site. If n is not zero, this indicates that image copies with the identified USID are required for the DBDSs marked “receive needed”.</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>DBRCVGRP=rcvgrpnm</td>
<td>The name of the recovery group to which this database belongs, if any. Otherwise, <strong>NULL</strong> is printed.</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>FLAGS: COUNTERS:</td>
<td>This heading line is printed for lines 8 through 14, which describe the status of this database.</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>BACKOUT NEEDED= n</td>
<td>This flag indicates whether this database needs to be backed-out by any subsystem.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECOVERY NEEDED COUNT= n</td>
<td>This counter is the number of DBDSs that are associated with this database that need to be recovered. The printed output of the DBDS record indicates which DBDSs need recovery.</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>READ ONLY= ON</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IMAGE COPY NEEDED COUNT= n</td>
<td>A count of how many DBDSs that are associated with this database require an image copy or forward recovery. The printed output of the DBDS records indicates which DBDSs require an image copy.</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>PROHIBIT AUTHORIZATION= n</td>
<td>This flag indicates whether this database is available for authorization processing. If the database can be authorized, the value is OFF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUTHORIZED SUBSYSTEMS= n</td>
<td>The count of subsystems that have current authorization to this database.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>RECOVERABLE= YES</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HELD AUTHORIZATION STATE= n</td>
<td>This is the state derived by the database authorization call process in IMS. It represents the composite use of the database by all currently authorized subsystems. If n= 0, no subsystem is authorized to use this database. If n&gt; 0, see line the 17 encoded state field below for the subsystem encoded state. If n&gt; 128, block-level data sharing is in effect. Subtract 128 to determine the true held authorization state.</td>
</tr>
</tbody>
</table>
### Fields in a DB (IMS) Record

**Table 47. Fields Present in the DB (IMS) Record (continued)**

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>DATABASE LEVEL TRACK=</td>
<td>YES | NO</td>
<td>Indicates the level of tracking for the database. YES is printed for database-readiness tracking, NO indicates recovery-readiness tracking. This is listed only if the database is contained in a GSG.</td>
</tr>
<tr>
<td></td>
<td>EEQE COUNT=</td>
<td>$n$</td>
<td>Number of extended error queue elements for this database.</td>
</tr>
<tr>
<td>13</td>
<td>TRACKING SUSPENDED=</td>
<td>YES | NO</td>
<td>Indicates whether tracking has been suspended for this shadow database.</td>
</tr>
<tr>
<td></td>
<td>RECEIVE REQUIRED COUNT=</td>
<td>$n$</td>
<td>Indicates how many of the DBDSs (for this shadow database) need image copies to be received from the active site.</td>
</tr>
<tr>
<td>14</td>
<td>OFR REQUIRED=</td>
<td>YES | NO</td>
<td>Indicates whether online forward recovery is required for this shadow database.</td>
</tr>
<tr>
<td>15</td>
<td>IC NEEDED DISABLED</td>
<td></td>
<td>For this non-recoverable DB, image copies will not be required after initial load (PROCOPT=L) or REORG of the DBDS.</td>
</tr>
</tbody>
</table>

The following lines 16 through 19 are printed only when one or more subsystems are currently authorized to this database:

| 16, 17, 18 | ASSOCIATED SUBSYSTEM INFORMATION: -SSID- -AC INTENT- ENCODED -STATE- B/O NEEDED -COUNT- -SS ROLE- | | Headings for lines 18-nnn. These headings are printed if subsystems exist. |
| 19-nnn     | ssidname | | The subsystem associated with this authorization. |
|            | access intent | | The intended access for the subsystem: READ, UPDATE, EXCLUSIVE, or READ-GO. |
|            | encoded state= | $n$ | An internal value derived by IMS to indicate the subsystem’s intended use of the database. The values of $n$ are as follows: 1 - Read only 2 - Read share 3 - Multiple update 4 - Read exclusive 5 - Batch update 6 - Single update 7 - Exclusive |
|            | backout needed count | | Determines that the database needs $n$ backouts by this subsystem. |
|            | ACTIVE \| TRACKING | | Role of authorized subsystem |

### Fields Present in a DB (HALDB) Record

The record fields and their corresponding line numbers for a HALDB master are described in Table 48.

**Table 48. Fields Present in the DB (HALDB) Record**

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB (HALDB Master)</td>
<td>1</td>
<td>DBD=dbname</td>
<td>Name of the HALDB.</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>DMB#=nnn</td>
<td>The value assigned when this database was registered.</td>
</tr>
</tbody>
</table>
### Fields Present in a DB (HALDB) Record

The record fields and their corresponding line numbers for a HALDB partition database are described in Table 48.

#### Table 48. Fields Present in the DB (HALDB) Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB (HALDB)</td>
<td></td>
<td>\textbf{CHANGE#} = nnn</td>
<td>The change version number. This is updated when the definition of the HALDB is changed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textbf{TYPE} = HALDB</td>
<td>Indicates that this is the HALDB master.</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>\textbf{SHARE LEVEL} = n</td>
<td>The level of data sharing for which authorized subsystems can share any partition of this HALDB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textbf{GSGNAME} = gsgname</td>
<td>Identifies the name of the GSG to which this database belongs.</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>\textbf{PSNAME} = psname</td>
<td>The name of the partition selection routine used by this HALDB, if any. Otherwise, <strong>NULL</strong> is printed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textbf{DBORG} = dbaseorg \textbf{DSORG} = dsetorg</td>
<td>The database and data set organization of the data DBDSs in this HALDB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textbf{CURRENT PARTITION ID} = nnnnn</td>
<td>The partition ID of the latest partition defined for the HALDB.</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>\textbf{FLAGS: COUNTERS:}</td>
<td>This heading line is printed for line 6, which describe the status of this HALDB.</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>\textbf{RECOVERABLE} = YES</td>
<td>The HALDB is recoverable or not.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textbf{PARTITIONS} = n</td>
<td>Number of partitions that exist for this HALDB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textbf{ONLINE REORG CAPABLE} = YES</td>
<td>The HALDB is capable (or not) for online reorganization. The ONLINE REORG CAPABLE = YES flag is included in the master HALDB and in the DB partition record.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textbf{IC NEEDED DISABLED}</td>
<td>For this non-recoverable DB, image copies will not be required after initial load (PROCOPT=L) or REORG of the DBDS.</td>
</tr>
</tbody>
</table>

### Fields Present in a DB (PART) Record

The record fields and their corresponding line numbers for a HALDB partition database are described in Table 49.

#### Table 49. Fields Present in the DB (PART) Record

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB (HALDB)</td>
<td>1</td>
<td>\textbf{DBD} = dbdname</td>
<td>Name of the HALDB partition.</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>\textbf{MASTER DB} = HALDB master name</td>
<td>Name of the HALDB master.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textbf{IRLMID} = xxxxx</td>
<td>Identifies the IRLM name. This field is printed only when SHARE LEVEL &gt;= 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textbf{CHANGE#} = nnn</td>
<td>The change version number is updated when the definition of the partition is changed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textbf{TYPE} = PART</td>
<td>Indicates a HALDB partition.</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>\textbf{USID} = n</td>
<td>The highest update set identifier for this partition.</td>
</tr>
</tbody>
</table>
### Table 49. Fields Present in the DB (PART) Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AUTHORIZED USID=(n)</td>
<td>The identifier of the current update set being applied to the partition.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HARD USID=(n)</td>
<td>The update set identifier of the latest changes written to the partition.</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>RECEIVE USID=(n)</td>
<td>The update set identifier of the last image copy received for this partition, which applies only to shadow databases at the tracking site.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECEIVE NEEDED USID=(n)</td>
<td>The update set identifier that indicates that image copies with the identified USID are required for the DBDSs marked “receive needed”, if (n) is not zero. It applies only to shadow databases at the tracking site.</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>DBRCVGRP=rcvgrpnm</td>
<td>The name of the recovery group to which this partition belongs, if any. Otherwise, <strong>NULL</strong> is printed.</td>
</tr>
<tr>
<td></td>
<td>5A</td>
<td>DSN PREFIX=dsname</td>
<td>The prefix name used for all data sets</td>
</tr>
<tr>
<td></td>
<td>5B</td>
<td>PARTITION ID=nnnnn</td>
<td>The partition identifier.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PREVIOUS PARTITION=(name)</td>
<td>The names of the previous and next partitions, for HALDBs using high keys.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NEXT PARTITION=(name)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>REORG#=(number)</td>
<td>The reorganization number for the partition.</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>OLRIMSID=<strong>NULL</strong></td>
<td>The OLRIMSID field reflects the name of the IMS that owns the online reorganization. OLRIMSID is always null at the tracking site.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACTIVE DBDS=A-J or ACTIVE DBDS=M-V</td>
<td>If a HALDB online reorganization is in progress, “ACTIVE DBDS=A-J” or “ACTIVE DBDS=M-V” is displayed. The input data set is listed first and the output data set is listed second.</td>
</tr>
<tr>
<td></td>
<td>8-9</td>
<td>RANDOMIZER: ()</td>
<td>Name of the randomizing module. Number of root anchor. Maximum relative block number value that the user wishes to allow a randomizing module to produce for this DB. Maximum number of bytes of a DB record that can be stored into the root addressable area in a series of inserts unbroken by a call to another DB record.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NAME=</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ANCHOR=</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HIGH BLOCK#=</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BYTES=nnnnnnnn</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10-11</td>
<td>FREE SPACE: ()</td>
<td>Specifies that every (n)th control interval or block in this data set group is left as free space during DB load or reorganization. Specifies the minimum percentage of each control interval or block that is to be left as free space.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FREE BLOCK FREQ FACTOR=(nnn)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FREE SPACE PERCENTAGE=(nn)</td>
<td></td>
</tr>
</tbody>
</table>
### Table 49. Fields Present in the DB (PART) Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>12a</td>
<td></td>
<td>PARTITION HIGH KEY/STRING (CHAR): (LENGTH=NNN)</td>
<td>Printable and hexadecimal formats of the partition high key/selection string. The key/string is padded with blanks, so the hexadecimal format may contain extra X'40' characters beyond the length of the key/string (LENGTH= in line 10 lists the actual key/string length).</td>
</tr>
<tr>
<td>12b</td>
<td></td>
<td>PARTITION HIGH KEY/STRING (HEX):</td>
<td>Printable and hexadecimal formats of the partition high key/selection string. The key/string is padded with blanks, so the hexadecimal format may contain extra X'40' characters beyond the length of the key/string (LENGTH= in line 10 lists the actual key/string length).</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>OSAM BLOCK SIZE: s = nnnnn</td>
<td>OSAM block size is listed for each data set group member defined. s is the data set group identifier A through J.</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>FLAGS: COUNTERS:</td>
<td>This heading line is printed for lines 7 through 14, which describe the status of this partition.</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>BACKOUT NEEDED=ON</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECOVERY NEEDED COUNT=n</td>
<td>This counter is the number of DBDSs associated with this database that need to be recovered. The printed output of the DBDS record indicates which DBDSs need recovery.</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>READ ONLY=ON</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IMAGE COPY NEEDED COUNT=n</td>
<td>Count of how many DBDSs associated with this partition require an image copy or forward recovery. The printed output of the DBDS records indicates which DBDSs require an image copy.</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>PROHIBIT AUTHORIZATION=ON</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUTHORIZED SUBSYSTEMS=n</td>
<td>The count of subsystems that have current authorization to this partition.</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>HELD AUTHORIZATION STATE=n</td>
<td>The state derived by the database authorization call process in IMS, representing the composite use of the partition by all currently authorized subsystems. If n = 0, no subsystem is authorized to use this partition. If n &gt; 0, see the table below for the subsystem encoded state. If n &gt; 128, online image copy (OLIC) is in progress for a data set of this partition.</td>
</tr>
</tbody>
</table>
## Table 49. Fields Present in the DB (PART) Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>DATABASE LEVEL TRACK=YES</td>
<td>Indicates the level of tracking for the database. YES is printed for database-readiness tracking. NO indicates recovery-readiness tracking. This is listed only if the database is contained in a GSG.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EEQE COUNT=n</td>
<td>Number of extended error queue elements for this partition.</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>TRACKING SUSPENDED=YES</td>
<td>Indicates whether tracking has been suspended for this shadow database.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RECEIVE REQUIRED COUNT=n</td>
<td>Indicates how many of the DBDSs (for this shadow database) need image copies from the active site to be received.</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>OFR REQUIRED=YES</td>
<td>Indicates whether online forward recovery is required for this shadow database.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OLR ACTIVE HARD COUNT=0</td>
<td>If the count is greater than zero, it reflects the number of HALDB online reorganizations that have started which have been recorded in the RECON but not yet hardened for this shadow database at the RSR tracking site.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OLR INACTIVE HARD COUNT=0</td>
<td>If the count is greater than zero, it reflects the number of HALDB online reorganizations that have completed which have been recorded in the RECON but not yet hardened for this shadow database at the RSR tracking site.</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>PARTITION INIT NEEDED=YES</td>
<td>Indicates whether partition initialization is required for this HALDB partition.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OLR REORG CURSOR ACTIVE=NO</td>
<td>The flag for OLREORG CURSOR ACTIVE is on when the paired sets of data sets are active. At the tracking site, the flag is on when the output data sets have been validated or created and that validation or creation process occurs within a hardened milestone.</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>PARTITION DISABLED=YES</td>
<td>A disabled partition is known only to DBRC and the Partition Definition Utility. A disabled partition can be enabled or deleted.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>ONLINE REORG CAPABLE=YES</td>
<td>The partition is enabled (or not) for online reorganization. The ONLINE REORG CAPABLE = YES</td>
<td>NO flag is included in the master HALDB and in the DB partition records.</td>
</tr>
<tr>
<td></td>
<td>NO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 49. Fields Present in the DB (PART) Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG INTENT= OFF</td>
<td>26</td>
<td>ON</td>
<td>The REORG INTENT flag is used by online reorganization products (other than the HALDB Online Reorganization function that is delivered with IMS) to inform DBRC they are active (they intend to reorganize the database - REORG INTENT = ON) or inactive (REORG INTENT = OFF). When the REORG INTENT flag is set to ON, database authorization fails for other reorganization products, batch access, and online image copy processes. The REORG INTENT flag is included in DB partition records and is not included in HALDB master DB records.</td>
</tr>
</tbody>
</table>

The following lines 26 through 28 are printed only when one or more subsystems are currently authorized to this partition:

| access intent | access intent | The intended access for the subsystem: READ, UPDATE, EXCLUSIVE, or READ-GO. |
| encoded state=n | encoded state=n | An internal value derived by IMS to indicate the subsystem's intended use of the partition. The values of n are as follows: |
| backout needed count | backout needed count | Determines that the partition needs n backouts by this subsystem. |
| ACTIVE|TRACKING | Role of authorized subsystem |

**Fields Present in a DB (Fast Path) Record**

A Fast Path database record's fields and their corresponding line numbers are described in **Table 50**

Table 50. Fields Present in the DB (Fast Path) Record

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB (Fast Path)</td>
<td>1</td>
<td>DBD= dbdname</td>
<td>Name of the database.</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>DMB#= nnm</td>
<td>The value assigned when this database was registered.</td>
</tr>
</tbody>
</table>
### Fields in DB (Fast Path) Record

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE= FP</td>
<td></td>
<td></td>
<td>Indicates that the database being used is a Fast Path DEDB.</td>
</tr>
<tr>
<td>3</td>
<td>SHARE LEVEL= n</td>
<td></td>
<td>The level of data sharing for which authorized subsystems can share this database.</td>
</tr>
<tr>
<td>4</td>
<td>FLAGS: COUNTERS:</td>
<td></td>
<td>This heading line is printed for lines 4 through 6 describing status of this database.</td>
</tr>
<tr>
<td>5</td>
<td>RECOVERY NEEDED COUNT= n</td>
<td></td>
<td>This counter is the number of area data sets, associated with this database, that need to be recovered. The printed output of the area data set record indicates which area data sets need recovery.</td>
</tr>
<tr>
<td>6</td>
<td>IMAGE COPY NEEDED COUNT= n</td>
<td></td>
<td>Count of how many area data sets, associated with this database, require an image copy or forward recovery. The printed output of the area data set records indicates which area data sets require an image copy.</td>
</tr>
<tr>
<td>7</td>
<td>PROHIBIT AUTHORIZATION= ON</td>
<td></td>
<td>This flag indicates whether this database is available for authorization processing. If the database can be authorized, the value is OFF.</td>
</tr>
<tr>
<td>8</td>
<td>AUTHORIZED AREAS= n</td>
<td></td>
<td>The count of areas that are currently authorized.</td>
</tr>
<tr>
<td></td>
<td>RECOVERABLE= YES</td>
<td></td>
<td>An indication of whether the database is recoverable (YES) or non-recoverable (NO).</td>
</tr>
<tr>
<td></td>
<td>EEQE COUNT= n</td>
<td></td>
<td>Number of extended error queue elements for this DEDB.</td>
</tr>
</tbody>
</table>

### Fields Present in a DBDS (non-Fast Path) Record

A non-Fast Path database data set record’s fields and their corresponding line numbers are described in Table 51.

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBDS (non-Fast Path)</td>
<td>1</td>
<td>DSN= dsname</td>
<td>Data set name of the DBDS.</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>TYPE= IMS</td>
<td>Indicates that this is a DBDS rather than a FP DEDB area.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TYPE= PART</td>
<td>Indicates that this is a DBDS of a HALDB partition.</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>DBD= dbdname, DSN= ddname</td>
<td>The database name and data set ddname of the DBDS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSID= nn</td>
<td>The data set ID number that appears as part of the information in the DBDLIB data set about the DBDS. The DSID for the M-V and Y DBDSs has the high order bit turned on, which is ignored for printing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DBORG= dbaseorg, DSORG= dsetorg</td>
<td>The database and data set organization, as defined for the DBDS in the DBDLIB data set.</td>
</tr>
</tbody>
</table>
### Fields in a DBDS (non-Fast Path) Record

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>CAGRP= cagrpm</td>
<td>The name of the CA group to which this DBDS belongs, if any. Otherwise, <strong>NULL</strong> is printed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GENMAX= mmm</td>
<td>The maximum number of image copy data sets to be maintained for this DBDS.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IC AVAIL= mmm</td>
<td>The number of available and in-use image copy data sets for the DBDS.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IC USED= mmm</td>
<td>The data set synchronization number that is being used concurrently by sharing IMS subsystems. A DSSN is used to reflect the relative order in which changes are made to a DBDS.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DSSN= mmm</td>
<td>REUSE</td>
<td>NOREUSE</td>
</tr>
<tr>
<td></td>
<td>RECOVPD= mmm</td>
<td>This is the recovery period of the image copies.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OTHER DDN=<strong>NULL</strong></td>
<td>The partner DBDS, if one exists. Otherwise <strong>NULL</strong> is printed.</td>
<td></td>
</tr>
</tbody>
</table>

The following fields in lines 6 through 11, identify the JCLPDS data set member names that are to be used in order to generate IMS utility JCL for this DBDS:

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>DEFLTJCL= member</td>
<td>The name of the PDS member containing the skeletal JCL default values for the user-defined keywords that are to be used for this DBDS.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ICJCL= member</td>
<td>The name of the skeletal JCL PDS member that is to be used in order to generate JCL for the Database Image Copy utility for this DBDS.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OICJCL= member</td>
<td>The name of the skeletal JCL PDS member that is to be used in order to generate a job for the Online Database Image Copy utility for this DBDS.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RECOVJCL= member</td>
<td>The name of the skeletal JCL PDS member that is to be used in order to generate JCL for the Database Recovery utility for this DBDS.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RECVJCL= member</td>
<td>The name of the skeletal JCL PDS member for which the Database Recovery utility is to receive an image copy for this DBDS at an RSR tracking site.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>FLAGS: COUNTERS:</td>
<td>Heading for the following lines 9 through 11, which describe the status of this DBDS.</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>IC NEEDED= ON</td>
<td>Indicates whether an image copy needs to be taken for the DBDS.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>IC RECOMMENDED= ON</td>
<td>Indicates that DBRC recommends an image copy of the DBDS should be taken before using the database.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>RECOV NEEDED= ON</td>
<td>Indicates whether the DBDS needs to be recovered.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>RECEIVE NEEDED = ON</td>
<td>Indicates whether an image copy of this DBDS needs to be received at the tracking site. This indicator is only applicable in an RSR environment at the RSR tracking site.</td>
<td></td>
</tr>
</tbody>
</table>
### Fields in a DBDS (non-Fast Path) Record

**Table 51. Fields Present in the DBDS (non-Fast Path) Record (continued)**

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>EEQE COUNT= n</td>
<td>The number of extended error queue elements for this DBDS.</td>
</tr>
</tbody>
</table>

The following, lines 13 through 15, are printed only if one or more extended error queue elements exist.

<table>
<thead>
<tr>
<th>13, 14</th>
<th>ERROR QUEUE ELEMENTS: -EQERBA-EEQE TYPE-SSID-</th>
<th>Heading for the following list of extended error queue elements (lines 14-nnn).</th>
</tr>
</thead>
<tbody>
<tr>
<td>15-nnn</td>
<td>eeqe rba</td>
<td>The relative byte address (RBA) of the EEQE.</td>
</tr>
<tr>
<td></td>
<td>eeqe type</td>
<td>The type of extended error queue element.</td>
</tr>
<tr>
<td></td>
<td>ssid</td>
<td>The ID of the subsystem that created the EEQE (in doubt EEQEs only).</td>
</tr>
</tbody>
</table>

### Fields in a DBDS (Fast Path) Record

A Fast Path database data set record’s fields and their corresponding line numbers are described in Table 52.

**Table 52. Fields Present in the DBDS (Fast Path) Record**

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBDS (Fast Path)</td>
<td>1</td>
<td>DBD= dbdname</td>
<td>The database name and area name of the Fast Path DEDB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AREA= areaname</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>IRLMID= irlmname</td>
<td>Identifies the IRLM when the share level of this DEDB is 2. This field is printed only when SHARE LEVEL=2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TYPE= FP</td>
<td>Identifies this database as a Fast Path DEDB.</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>SHARE LEVEL= n</td>
<td>The level of data sharing for which authorized subsystems can share this area.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSID= nn</td>
<td>The data set ID number that appears as part of the information in the DBDLIB data set about the area data set.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DBORG= dbaseorg DSORG= dsetorg</td>
<td>The database and data set organization, as defined for the area data set in the DBDLIB data set.</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>GSGNAME= gsgname</td>
<td>Identifies the name of the GSG to which this area belongs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USID= n</td>
<td>The highest update set identifier for this area.</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>AUTHORIZED USID= n</td>
<td>The identifier of the current update set that is being applied to the area.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECEIVE USID= n</td>
<td>The update set identifier of the last image copy that was received for this area. The update set ID only applies to shadow areas at the tracking site.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HARD USID= n</td>
<td>The update set identifier of the latest changes that were written to the area.</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>RECEIVE NEEDED USID= n</td>
<td>This only applies to shadow areas at the tracking site. If n is not zero, this indicates that an image copy with the identified USID needs to be received for this area.</td>
</tr>
</tbody>
</table>
### Table 52. Fields Present in the DBDS (Fast Path) Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
<td>CAGRP= cagrpnam</td>
<td>The name of the CA group to which this area belongs, if any. Otherwise, <strong>NULL</strong> is printed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GENMAX= mnnn</td>
<td>The maximum number of image copy data sets to be maintained for this area.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IC AVAIL= mnnn</td>
<td>The number of available image copy data sets, and the number of in-use image copy data sets for the area.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IC USED= mnnn</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSSN= mnnn</td>
<td>This is the data set synchronization number that is being used concurrently by sharing IMS subsystems. A DSSN is used to reflect the relative order in which changes are made to an area data set.</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>REUSE</td>
<td>NOREUSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECOVPD= mnnn</td>
<td>This is the recovery period of the image copies.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VSO</td>
<td>NOVSO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PREOPEN</td>
<td>NOPREOPEN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PRELOAD</td>
<td>NOPRELOAD</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>CFSTR1= cfstr_name</td>
<td>The name of the first coupling facility structure for the area.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CFSTR2= cfstr_name</td>
<td>The name of the second coupling facility structure for the area.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LKASID</td>
<td>NOLKASID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAS</td>
<td>NOMAS</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>DEFLTJCL= member</td>
<td>The name of the member of the partitioned data set that contains the skeletal JCL default values that are to be used for the DEDB area.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ICJCL= member</td>
<td>The name of the skeletal JCL PDS member that is to be used in order to generate the JCL for the Database Image Copy utility for this area data set.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECVJCL= member</td>
<td>The name of the skeletal JCL PDS member for which the Database Recovery utility is to receive an image copy for this area data set at an RSR tracking site.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECOVJCL= member</td>
<td>The name of the skeletal JCL PDS member that is to be used in order to generate the JCL for the Database Recovery utility for this area data set.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>DBRCVGRP=rcogrpm</td>
<td>The name of the recovery group to which this DBDS belongs, if any. Otherwise, <strong>NULL</strong> is printed.</td>
</tr>
</tbody>
</table>
### Fields in a DBDS (Fast Path) Record

Table 52. Fields Present in the DBDS (Fast Path) Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td></td>
<td>FLAGS: COUNTERS:</td>
<td>This heading line is printed for lines 11 through 19 describing the status of this area.</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>PROHIBIT AUTHORIZATION= ON</td>
<td>The value of this flag is OFF if the area is available for authorization processing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AUTHORIZED SUBSYSTEMS= n</td>
<td>The count of subsystems that have current authorization to this area.</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>HELD AUTHORIZATION STATE= n</td>
<td>This is the state derived by the database authorization call process in IMS. It represents the composite use of the database by all currently authorized subsystems. If $n=0$, no subsystem is authorized to use this database. If $n&gt;0$, see the table below for the subsystem encoded state. If $n&gt;128$, block-level data sharing is in effect. Subtract 128 to determine the true held authorization state.</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>IC NEEDED= ON</td>
<td>Indicates whether an image copy needs to be taken for the DEDB area.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ADS AVAIL #= n</td>
<td>Indicates the number of available ADS in this area record.</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>IC RECOMMENDED= ON</td>
<td>Indicates that DBRC recommends an image copy of the area should be taken before it is used.</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>RECOV NEEDED= ON</td>
<td>Indicates whether the areas associated with the DEDB should be recovered.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REGISTERED ADS #= n</td>
<td>Indicates how many area data sets for this area are registered in RECON.</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>DATABASE LEVEL TRACK = YES</td>
<td>Indicates the level of tracking for the area. YES is printed for database-readiness tracking. NO indicates recovery-readiness tracking. This is listed only if the area is contained in a GSG.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NO</td>
<td>EEQE COUNT= n</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>RECEIVE NEEDED = ON</td>
<td>Indicates whether an image copy of this area needs to be received at the tracking site. This indicator is only applicable in an RSR environment at the RSR tracking site.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF</td>
<td>OFR REQUIRED= YES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NO</td>
<td>TRACKING SUSPENDED= YES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NO</td>
<td>HSSP CIC IN PROGRESS= YES</td>
</tr>
</tbody>
</table>

The following lines, 23 through 25, are printed only if one or more error queue elements exist.

| 23          |             | ERROR QUEUE ELEMENTS: -EGERBA-EEQE TYPE-SSID- | Heading for the following list of error queue elements. |
| 24          |             | eeqe rba | The type of extended error queue element. |
| 25-mm       |             | eeqe type | The type of extended error queue element. |
|             |             | ssid | The ID of the subsystem that created the EEQE (in doubt EEQEs only). |
Fields in a DBDS (Fast Path) Record

The following lines 26 through 28 represent the column headings for the following line 29:

<table>
<thead>
<tr>
<th>Line Number</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>ADS LIST: -ADS DDN- -ADS</td>
</tr>
<tr>
<td>27</td>
<td>DSN- -STAT- CREATE</td>
</tr>
<tr>
<td>28</td>
<td>-RUNNING-</td>
</tr>
</tbody>
</table>

The following line (29) is repeated for each area data set that is registered for this area:

<table>
<thead>
<tr>
<th>nn-mm</th>
<th>adsddn</th>
<th>adsdsn</th>
<th>AVAIL</th>
<th>UNAVAIL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>

The following lines nn+1 through nn+3, are printed only when one or more subsystems are currently authorized to this area:

<table>
<thead>
<tr>
<th>nn+1</th>
<th>nn+2</th>
<th>ASSOCIATED SUBSYSTEM INFORMATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn+3</td>
<td></td>
<td>-SSID- -ACCESS INTENT- ENCODED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-STATE- -SS ROLE-</td>
</tr>
</tbody>
</table>

The following line nn+3-mm, is repeated for each subsystem that has authorization for this database:

<table>
<thead>
<tr>
<th>mm-mm</th>
<th>ssidname</th>
<th>The subsystem associated with this authorization.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>access intent</td>
<td>The intended access for the subsystem: READ, UPDATE, EXCLUSIVE, or READ-GO.</td>
</tr>
<tr>
<td></td>
<td>encoded state= n</td>
<td>An internal value derived by IMS to indicate the subsystem’s intended use of the database. The values of n are as follows: 1 - Read only 2 - Read share 3 - Multiple update 4 - Read exclusive 5 - Batch update 6 - Single update 7 - Exclusive</td>
</tr>
<tr>
<td></td>
<td>ACTIVE</td>
<td>TRACKING</td>
</tr>
</tbody>
</table>

Fields Present in an ALLOC Record

An allocation record’s fields and their corresponding line numbers are described in [Table 53](#).
### Fields in an ALLOC Record

#### Table 53. Fields Present in the ALLOC Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>DSSN= nnnn</td>
<td>This is the data set synchronization number that is being concurrently used by sharing IMS subsystems. A DSSN is used to reflect the relative order in which changes are made to a DBDS or area.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USID= n</td>
<td>The update set identifier. An update set is a collection of updates that are made to the database or area while it is continuously updated by one or more IMS subsystems.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>START= time_stamp</td>
<td>The start time of the log data set that was logging change records when this DBDS was allocated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DEALLOC= time_stamp</td>
<td>The time stamp of a specific deallocation. This field is printed only if a specific deallocation time is in the record. If the DBDS was allowed to remain allocated to the closing of the log data set, this line is not printed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DEALLOC LRID= n</td>
<td>The log record sequence number of the end-update log record. This field is only printed if a specific deallocation time exists.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TRACKING SUSPENDED AT</td>
<td>This will only be printed for shadow databases that are being tracked at the tracking site. The suspend lrid indicates the current tracking log position.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECORD: suspend lrid</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TRACKING SUSPENDED NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECORDS APPLIED</td>
<td></td>
</tr>
</tbody>
</table>

### Fields Present in an IMAGE Record

An image record’s fields and their corresponding line numbers are described in [Table 54](#).

#### Table 54. Fields Present in the IMAGE Record

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAGE</td>
<td>1</td>
<td></td>
<td>The number of records contained in the image copy data set. This field is printed only for in-use image copy data sets; it does not apply to nonstandard or system managed storage image copy data sets.</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>*</td>
<td>The asterisk (*) is printed to the left of the record type only if the image copy data sets that are defined in this record are available for the first time for future use (they have never been used before).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CREATE</td>
<td>RUN= time_stamp*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECORD COUNT= nnnnnnn</td>
<td>This is printed for any nonstandard image copy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USER IC</td>
<td>This is printed for any nonstandard image copy.</td>
</tr>
</tbody>
</table>
### Fields in an IMAGE Record

**Table 54. Fields Present in the IMAGE Record (continued)**

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>USID= n</td>
<td>Line Number</td>
<td>USID= n</td>
<td>This is only listed here for a nonstandard image copy.</td>
</tr>
<tr>
<td>3</td>
<td>USERDATA= ecccccc...</td>
<td>This line is printed in this format for a nonstandard image copy data set. The characters following USERDATA= comprise the character string you kept in the record to describe the nonstandard image copy data set.</td>
<td></td>
</tr>
<tr>
<td>STOP=</td>
<td>time_stamp</td>
<td>This field contains either zeros or the stop time of an Online Database Image Copy utility run. This field applies to standard image copy data sets, but is not printed for available image copy data sets.</td>
<td></td>
</tr>
<tr>
<td>ONLINE</td>
<td>BATCH</td>
<td>CONCUR</td>
<td>SMSCIC</td>
</tr>
<tr>
<td>4</td>
<td>IN PROGRESS</td>
<td>This field is not printed for available image copy data sets.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>IC1</td>
<td>IC2</td>
<td>IC1 and accompanying lines are always printed, followed by IC2 and accompanying lines, if a duplicate image copy data set exists.</td>
</tr>
<tr>
<td>6</td>
<td>DSN= ic-dsname</td>
<td>ic-dsname is the data set name of this image copy data set (IC1 or IC2).</td>
<td></td>
</tr>
<tr>
<td>FILE SEQ= nnnn</td>
<td>The file sequence number of the image copy data set on the first volume on which it resides. This field is listed for non-HSSP image copies only.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>UNIT= unitype</td>
<td>Unit type to be used for substitution during the 6EJNCL process. This field is listed for non-HSSP image copies only.</td>
<td></td>
</tr>
<tr>
<td>ERR</td>
<td>EMP</td>
<td>ERR is printed if you have indicated that the image copy data set is not to be used as input to future utility runs. EMP is printed only for duplicate image copy data sets that have not been used, even though their corresponding image copy data set has been used.</td>
<td></td>
</tr>
</tbody>
</table>
Fields in an IMAGE Record

Table 54. Fields Present in the IMAGE Record (continued)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>VOLS DEF= nnnn</td>
<td>The number of volumes that have been specified for use by this image copy data set and the number of volumes that have been used to create the data set. The VOLS USED value might be less than the specified VOLS DEF. For available image copy data sets, the VOLS USED value is 0. This field is listed for non-HSSP image copies only.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VOLS USED= nnnn</td>
<td></td>
</tr>
</tbody>
</table>

8-nn  VOLSER= volser,...   A list of the volume serial numbers of volumes that contain the image copy data set. This field is listed for non-HSSP image copies only.

Fields Present in a REORG Record

A reorganization record’s fields and their corresponding line numbers are described in Table 55.

Table 55. Fields Present in the REORG Record

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORG</td>
<td>1</td>
<td>RUN= time_stamp*</td>
<td>The time stamp you have supplied for the time at which a reorganization occurred for this DBDS. An asterisk (*) indicates that the time stamp is part of the key of this record.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USID= n</td>
<td>This is the value of the update set identifier of the database or area when the reorganization of this DBDS occurred.</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>REORG#=number</td>
<td>The reorganization number for the partition at the time of the reorganization.</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>STOP=time_stamp*</td>
<td>STOP= is only present for online reorganizations. STOP= is zero for online reorganizations that have not yet completed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ONLINE RECOV=</td>
<td>YES indicates the online reorganization was taken when this DBDS was the output data set and the start time can be used as the purgetime for recovery. NO indicates the online reorganization was taken when this DBDS was the input data set and cannot be used for recovery.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>OLDSTOPT = time_stamp</td>
<td>If a point-in-time recovery (PITR) has been run to the middle of a completed online reorganization, the prior stop time of the online reorganization is moved to this field. OLDSTOPT= is only present if non-zero and is used for diagnostic information only.</td>
</tr>
</tbody>
</table>

Fields Present in a RECOV Record

A recovery record’s fields and their corresponding line numbers are described in Table 56 on page 559.
<table>
<thead>
<tr>
<th>Record Type</th>
<th>Line Number</th>
<th>Field</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECOV</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>RUN= time_stamp*</td>
<td>The time stamp of a Database Recovery utility run for this DBDS. The asterisk (*) indicates that the time stamp is part of the key of this record.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RUN USID= n</td>
<td>Listed only if a time-stamp recovery is being described. This identifies the update set identifier of the DBDS when the Database Recovery utility was run.</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>RECOV TO= time_stamp</td>
<td>This field indicates that a time-stamp recovery is being described. The time-stamp recovery restored the DBDS to the state it was in at the time represented by the time stamp.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECOV TO USID= n</td>
<td>Listed only if a time-stamp recovery is being described. This identifies the update set identifier of the DBDS at the time to which the DBDS was restored.</td>
</tr>
</tbody>
</table>

Table 56. Fields Present in the RECOV Record
Fields in a RECOV Record
Appendix C. Resource Names for Command Authorization

DBRC uses a list of resource names for command authorization support. For more information about command authorization, see “Preserving the Integrity of the RECON Data Set with DBRC Command Authorization Support” on page 79.

Table 57 lists the resources that can be protected.

Table 57. Resource Names for Command Authorization

<table>
<thead>
<tr>
<th>Verb</th>
<th>Modifier</th>
<th>Qualifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>BACKUP</td>
<td>.RECON</td>
<td></td>
</tr>
<tr>
<td>CHANGE</td>
<td>.ADS</td>
<td>.dbname</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.BKOUT</td>
<td>.ssid</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.CA</td>
<td>.grpname</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.CAGRP</td>
<td>.grpname</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.DB</td>
<td>.dbname</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.DB</td>
<td>.ALL</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.DBDS</td>
<td>.dbname</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.DBDSGRP</td>
<td>.grpname</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.IC</td>
<td>.dbname</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.PART</td>
<td>.dbname</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.PRILOG</td>
<td>.OLDS</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.PRILOG</td>
<td>.RLDS</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.PRILOG</td>
<td>.SLDS</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.PRILOG</td>
<td>.TSLDS</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.CATDS</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.CDSDLID</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.NOCATDS</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.CHECK17</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.CHECK44</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.NOCHECK</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.CMDAUTH</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.DASDUNIT</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.DUAL</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.REPLACE</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.FORCER</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.NOFORCER</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.IMSPLEX</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.NOPLEX</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.LISTDL</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.NOLISTDL</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.LOGALERT</td>
</tr>
</tbody>
</table>
Table 57. Resource Names for Command Authorization (continued)

<table>
<thead>
<tr>
<th>Verb</th>
<th>Modifier</th>
<th>Qualifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.LOGRET</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.MINVERS</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.RESET</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.SIZALERT</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.SSID</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.STARTNEW</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.NONEW</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.TAPEUNIT</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.TIMEZONE</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.TIMEZIN</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.TIMEFMT</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.TRACEON</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.TRACEOFF</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.TRACEOPT</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.RECON</td>
<td>.UPGRADE</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.SG</td>
<td>.gsgname</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.SUBSYS</td>
<td>.ssid</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.SUBSYS</td>
<td>.ALL</td>
</tr>
<tr>
<td>CHANGE</td>
<td>.UIC</td>
<td>.dbname</td>
</tr>
<tr>
<td>DELETE</td>
<td>.ADS</td>
<td>.dbname</td>
</tr>
<tr>
<td>DELETE</td>
<td>.ALLOC</td>
<td>.dbname</td>
</tr>
<tr>
<td>DELETE</td>
<td>.BKOUT</td>
<td>.ssid</td>
</tr>
<tr>
<td>DELETE</td>
<td>.CA</td>
<td>.grpname</td>
</tr>
<tr>
<td>DELETE</td>
<td>.CAGRP</td>
<td>.grpname</td>
</tr>
<tr>
<td>DELETE</td>
<td>.DB</td>
<td>.dbname</td>
</tr>
<tr>
<td>DELETE</td>
<td>.DBDS</td>
<td>.dbname</td>
</tr>
<tr>
<td>DELETE</td>
<td>.DBDSGRP</td>
<td>.grpname</td>
</tr>
<tr>
<td>DELETE</td>
<td>.GSG</td>
<td>.gsgname</td>
</tr>
<tr>
<td>DELETE</td>
<td>.IC</td>
<td>.dbname</td>
</tr>
<tr>
<td>DELETE</td>
<td>.LOG</td>
<td>.INACTIVE</td>
</tr>
<tr>
<td>DELETE</td>
<td>.LOG</td>
<td>.OLDS</td>
</tr>
<tr>
<td>DELETE</td>
<td>.LOG</td>
<td>.STARTIME</td>
</tr>
<tr>
<td>DELETE</td>
<td>.LOG</td>
<td>.TOTIME</td>
</tr>
<tr>
<td>DELETE</td>
<td>.PART</td>
<td>.dbname</td>
</tr>
<tr>
<td>DELETE</td>
<td>.RECOV</td>
<td>.dbname</td>
</tr>
<tr>
<td>DELETE</td>
<td>.REORG</td>
<td>.dbname</td>
</tr>
<tr>
<td>DELETE</td>
<td>.SG</td>
<td>.gsgname</td>
</tr>
<tr>
<td>DELETE</td>
<td>.SUBSYS</td>
<td>.ssid</td>
</tr>
<tr>
<td>DELETE</td>
<td>.UIC</td>
<td>.dbname</td>
</tr>
<tr>
<td>GENJCL</td>
<td>.ARCHIVE</td>
<td>.ssid</td>
</tr>
<tr>
<td>Verb</td>
<td>Modifier</td>
<td>Qualifier</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td>GENJCL</td>
<td>.CA</td>
<td>grpname</td>
</tr>
<tr>
<td>GENJCL</td>
<td>.CLOSE</td>
<td>ssid</td>
</tr>
<tr>
<td>GENJCL</td>
<td>.IC</td>
<td>dbname</td>
</tr>
<tr>
<td>GENJCL</td>
<td>.IC</td>
<td>grpname</td>
</tr>
<tr>
<td>GENJCL</td>
<td>.OIC</td>
<td>dbname</td>
</tr>
<tr>
<td>GENJCL</td>
<td>.OIC</td>
<td>grpname</td>
</tr>
<tr>
<td>GENJCL</td>
<td>.RECEIVE</td>
<td>dbname</td>
</tr>
<tr>
<td>GENJCL</td>
<td>.RECEIVE</td>
<td>grpname</td>
</tr>
<tr>
<td>GENJCL</td>
<td>.RECOV</td>
<td>dbname</td>
</tr>
<tr>
<td>GENJCL</td>
<td>.RECOV</td>
<td>grpname</td>
</tr>
<tr>
<td>GENJCL</td>
<td>.USER</td>
<td>mbrname</td>
</tr>
<tr>
<td>GENJCL</td>
<td>.USER</td>
<td>dbname</td>
</tr>
<tr>
<td>GENJCL</td>
<td>.USER</td>
<td>grpname</td>
</tr>
<tr>
<td>INIT</td>
<td>.ADS</td>
<td>dbname</td>
</tr>
<tr>
<td>INIT</td>
<td>.CA</td>
<td>grpname</td>
</tr>
<tr>
<td>INIT</td>
<td>.CAGRP</td>
<td>grpname</td>
</tr>
<tr>
<td>INIT</td>
<td>.DB</td>
<td>dbname</td>
</tr>
<tr>
<td>INIT</td>
<td>.DB</td>
<td>dbname</td>
</tr>
<tr>
<td>INIT</td>
<td>.DB</td>
<td>gsgname</td>
</tr>
<tr>
<td>INIT</td>
<td>.IC</td>
<td>dbname</td>
</tr>
<tr>
<td>INIT</td>
<td>.PART</td>
<td>dbname</td>
</tr>
<tr>
<td>INIT</td>
<td>.RECON</td>
<td></td>
</tr>
<tr>
<td>INIT</td>
<td>.SG</td>
<td>gsgname</td>
</tr>
<tr>
<td>LIST</td>
<td>.BKOUT</td>
<td>ssid</td>
</tr>
<tr>
<td>LIST</td>
<td>.BKOUT</td>
<td>ALL</td>
</tr>
<tr>
<td>LIST</td>
<td>.CAGRP</td>
<td>grpname</td>
</tr>
<tr>
<td>LIST</td>
<td>.CAGRP</td>
<td>ALL</td>
</tr>
<tr>
<td>LIST</td>
<td>.DB</td>
<td>dbname</td>
</tr>
<tr>
<td>LIST</td>
<td>.DB</td>
<td>ALL</td>
</tr>
<tr>
<td>LIST</td>
<td>.DB</td>
<td>TYPEIMS</td>
</tr>
<tr>
<td>LIST</td>
<td>.DB</td>
<td>TYPEFP</td>
</tr>
<tr>
<td>LIST</td>
<td>.DB</td>
<td>TYPHALDB</td>
</tr>
<tr>
<td>LIST</td>
<td>.DBDS</td>
<td>dbname</td>
</tr>
<tr>
<td>LIST</td>
<td>.DBDS</td>
<td>grpname</td>
</tr>
<tr>
<td>LIST</td>
<td>.DBDSGRP</td>
<td>grpname</td>
</tr>
<tr>
<td>LIST</td>
<td>.DBDSGRP</td>
<td>ALL</td>
</tr>
<tr>
<td>LIST</td>
<td>.GSG</td>
<td>gsgname</td>
</tr>
<tr>
<td>LIST</td>
<td>.GSG</td>
<td>ALL</td>
</tr>
<tr>
<td>LIST</td>
<td>.HISTORY</td>
<td>dbname</td>
</tr>
</tbody>
</table>
Table 57. Resource Names for Command Authorization  (continued)

<table>
<thead>
<tr>
<th>Verb</th>
<th>Modifier</th>
<th>Qualifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST</td>
<td>.HISTORY</td>
<td>.grpname</td>
</tr>
<tr>
<td>LIST</td>
<td>.LOG</td>
<td>.ALL</td>
</tr>
<tr>
<td>LIST</td>
<td>.LOG</td>
<td>.ALLOLDS</td>
</tr>
<tr>
<td>LIST</td>
<td>.LOG</td>
<td>.OLDS</td>
</tr>
<tr>
<td>LIST</td>
<td>.LOG</td>
<td>.STARTIME</td>
</tr>
<tr>
<td>LIST</td>
<td>.RECON</td>
<td></td>
</tr>
<tr>
<td>LIST</td>
<td>.RECON</td>
<td>.STATUS</td>
</tr>
<tr>
<td>LIST</td>
<td>.SUBSYS</td>
<td>.ONLINE</td>
</tr>
<tr>
<td>LIST</td>
<td>.SUBSYS</td>
<td>.ssid</td>
</tr>
<tr>
<td>LIST</td>
<td>.SUBSYS</td>
<td>.ALL</td>
</tr>
<tr>
<td>LIST</td>
<td>.SUBSYS</td>
<td>.BATCH</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.ALLOC</td>
<td>.dbname</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.BKOUT</td>
<td>.ssid</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.CA</td>
<td>.grpname</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.IC</td>
<td>.dbname</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.PRILOG</td>
<td>.OLDS</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.PRILOG</td>
<td>.RLDS</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.PRILOG</td>
<td>.SLDS</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.PRILOG</td>
<td>.TSLDS</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.PRILOG</td>
<td>.DPROP</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.RECOV</td>
<td>.dbname</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.REORG</td>
<td>.dbname</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.SECLOG</td>
<td>.OLDS</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.SECLOG</td>
<td>.RLDS</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.SECLOG</td>
<td>.SLDS</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.SECLOG</td>
<td>.TSLDS</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.SECLOG</td>
<td>.DPROP</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.SUBSYS</td>
<td>.ssid</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>.UIC</td>
<td>.dbname</td>
</tr>
<tr>
<td>RESET</td>
<td>.CSG</td>
<td>.gsgname</td>
</tr>
</tbody>
</table>
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.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

**Trademarks**

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

- BookManager
- CICS
- DataPropagator
- DB2
- DB2 Universal Database
- DFSMS/MVS
- DFSMSdss
- IBM
- IMS
- IMS/ESA
- MVS
- NetView
- OS/390
- RACF
- Tivoli
- WebSphere
- z/OS

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

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

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

This bibliography includes all the publications cited in this book, including the publications in the IMS library.

- CICS Transaction System for z/OS CICS Supplied Transaction, SC34-5992
- CICS Transaction System for z/OS IMS Database Control Guide, SC34-6010
- DFMS/MVS V1R5 MDFSMSdfp Advanced Services, SC26-4921
- High Level Assembler for MVS & VM & VSE Language Reference, SC26-4940
- z/OS MVS Planning: Global Resource Serialization, GC28-1759
- z/OS MVS Programming: Authorized Assembler Services Reference, GC28-1763
- z/OS DFMS Access Method Services for Integrated Catalog Facility
- z/OS V1R3 MVS Programming : Assembler Services, Volume 1, SA22-7606
- z/OS V1R3 MVS Programming : Assembler Services, Volume 2, SA22-7607

IMS V9 Library

<table>
<thead>
<tr>
<th>Title</th>
<th>Acronym</th>
<th>Order number</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS V9: Administration Guide: Database Manager</td>
<td>ADB</td>
<td>SC18-7806</td>
</tr>
<tr>
<td>IMS V9: Administration Guide: Transaction Manager</td>
<td>ATM</td>
<td>SC18-7808</td>
</tr>
<tr>
<td>IMS V9: Application Programming: Database Manager</td>
<td>APDB</td>
<td>SC18-7809</td>
</tr>
<tr>
<td>IMS V9: Application Programming: EXEC DLI Commands for CICS and IMS</td>
<td>APTM</td>
<td>SC18-7812</td>
</tr>
<tr>
<td>IMS V9: Application Programming: Transaction Manager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMS V9: Base Primitive Environment Guide and Reference</td>
<td>BPE</td>
<td>SC18-7813</td>
</tr>
<tr>
<td>IMS V9: Command Reference</td>
<td>CR</td>
<td>SC18-7814</td>
</tr>
</tbody>
</table>

Supplementary Publications

<table>
<thead>
<tr>
<th>Title</th>
<th>Order number</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS Connector for Java 2.2.2 and 9.1.0.1</td>
<td>SC09-7869</td>
</tr>
<tr>
<td>Online Documentation for WebSphere Studio Application Developer Integration Edition 5.1.1</td>
<td></td>
</tr>
<tr>
<td>IMS V9 Fact Sheet</td>
<td>GC18-7697</td>
</tr>
</tbody>
</table>
## IMS V9: Licensed Program Specifications

### Publication Collections

<table>
<thead>
<tr>
<th>Title</th>
<th>Format</th>
<th>Order number</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS V9 Softcopy Library</td>
<td>CD</td>
<td>LK3T-7213</td>
</tr>
<tr>
<td>IMS Favorites</td>
<td>CD</td>
<td>LK3T-7144</td>
</tr>
<tr>
<td>Licensed Bill of Forms (LBOF): IMS V9 Hardcopy and Softcopy Library</td>
<td>Hardcopy and CD</td>
<td>LBOF-7789</td>
</tr>
<tr>
<td>Unlicensed Bill of Forms (SBOF): IMS V9 Unlicensed Hardcopy Library</td>
<td>Hardcopy</td>
<td>SBOF-7790</td>
</tr>
<tr>
<td>OS/390 Collection</td>
<td>CD</td>
<td>SK2T-6700</td>
</tr>
<tr>
<td>z/OS Software Products Collection</td>
<td>CD</td>
<td>SK3T-4270</td>
</tr>
<tr>
<td>z/OS and Software Products DVD Collection</td>
<td>DVD</td>
<td>SK3T-4271</td>
</tr>
</tbody>
</table>

### Accessibility Titles Cited in This Library

<table>
<thead>
<tr>
<th>Title</th>
<th>Order number</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS V1R1.0 TSO Primer</td>
<td>SA22-7787</td>
</tr>
<tr>
<td>z/OS V1R5.0 TSO/E User's Guide</td>
<td>SA22-7794</td>
</tr>
<tr>
<td>z/OS V1R5.0 ISPF User's Guide, Volume 1</td>
<td>SC34-4822</td>
</tr>
</tbody>
</table>
analyzing requirements for data sharing, assigning a sharing level with DBRC 17
application programming interface
DBRC (Database Recovery Control) accessing RECON data sets 353
addressing and residency 350
coding parameters 351
DSPAPI macro 349
ending the environment 350
establishing the environment 350
how to access 349
overview 349
prerequisite knowledge required 349
QUERY request 357
runtime considerations 353
sample application program 349
services available 349
token 352
use in an IMSplex 355
use of registers 350
using EQU statements 351
ARCHIVE, GENJCL
archiving OLDS 9
ARCHIVED parameter
CHANGE.PRILOG (for OLDS) command 167
archiving
online log data sets, DBRC 8
ARCHJCL skeletal JCL execution member 440
area data sets
INIT.ADS command 21
AREA parameter
commands
CHANGE.ADS 133
CHANGE.DBDS 151
CHANGE.JC 159
CHANGE.IUC 200
DELETE.ADS 201
DELETE.ALOC 202
DELETE.DBDS 205
DELETE.IC 207
DELETE.RECOV 212
DELETE.UIC 214
GENJCL.IC 231
GENJLCO.IOC 239
GENJLRECEIVE 244
GENJL.RECOV 248
INIT.ADS 260
INIT.DBDS 267
INIT.IC 274
LIST.DBDS 291
LIST.HISTORY 295
NOTIFY.ALOC 306
NOTIFY.IC 311
NOTIFY.RECOV 326
NOTIFY.UIC 343
defining a data set 259
deleting information from RECON 201
AREANEW parameter
CHANGE.DBDS command 152
ARNEEDED parameter
CHANGE.PRILOG (for OLDS) command 167
ARSCHED parameter
CHANGE.PRILOG (for OLDS) command 167
ARSTART parameter
CHANGE.PRILOG (for OLDS) command 167
assigning a sharing level with DBRC 17
authorization commands
CHANGE.DB 141, 143
CHANGE.DBDS 149, 152
authorization, changing database 112
Automatic RECON Loss Notification 181
AVAIL parameter commands
CHANGE.ADS 134
CHANGE.PRILOG (for OLDS) 167
CHANGE.SECLOG (for OLDS) 187
INIT.ADS 260
BACKIRLM parameter commands
CHANGE.SUBSYS 198
BACKOUT parameter
CHANGE.DB command 144
BACKOUT query
description 359
output 361
parameters 359
return codes 360
syntax 359
backout, BACKOUT record for recovery 66
backout, with data sharing
batch 22
dynamic 22
backup
creating a copy of RECON 131
database 35
command for 37
description 33
RECON data set 73
BACKUP.RECON command 73
description 131
example 131
Parameters
BOTH 131
RECON 131
RECON 131
batch backup utility 22, 25
adding backup records to RECON 307
changing backup records in RECON 134
deleting from RECON 203
limitations 23
SSID naming convention 111
batch environment
command
LIST.SUBSYS 302
NOTIFY.SUBSYS 342
NOTIFY.BUTTON command 341
batch job authorization 79
BKO (backup) parameter commands
CHANGE.BKOUT 136
NOTIFY.BKOUT 307
BOTH parameter
BACKUP.RECON command 131
C
CADSN parameter commands
CHANGE.CA 137
INIT.CA 261
NOTIFY.CA 308
CAJCL parameter commands
CHANGE.CAGRP 139
INIT.CAGRP 262
skeletal JCL execution member 446
calls from IMS to DBRC, elements of DBRC 4
catalog management of data sets in RECON 113
CATDS
CHANGE.RECON 178
CATDS parameter commands
CHANGE.RECON 177
INIT.RECON 281
category of records
listing records 298
CATIME parameter
GENJCL.CA command 224
CDSLID
CHANGE.RECON 178
CFSTR1 parameter CHANGE.DBDS
command 151, 270
CFSTR2 parameter CHANGE.DBDS
command 152, 270
change accumulation data set
defining 260
naming convention 34
selecting 100
definition of 10
group
changing information in the RECON 138
defining 47, 261
defining for future use 47
deleting information from RECON 204
listing 288
reusing 47
using 48
record
group 66
run 66
stops processing logs, when 112
change accumulation says nothing to process, when 112
change accumulation stops processing logs, when 112
Change Accumulation utility
See Database Change Accumulation utility (DFSUCUM0)
CHANGE.ADS command 133
CHANGE.BKOUT command 134
CHANGE.CA command 136
CHANGE.CAGRP command 138
CHANGE.DB
NOREORGI parameter 149
REORGI parameter 149
CHKPTCT parameter commands  
(continued)  
NOTIFY.PRILOG (for SLDS and TSLDS) 323  
NOTIFY.SECLOG (for RLDS) 335  
NOTIFY.SECLOG (for SLDS) 339  
CIC (concurrent image copy)  
for concurrent image copy (CIC)  
CIC parameter commands  
GENJCL.IC 231  
NOTIFY.IC 312  
for online image copy 38  
CHANGE.RECON, commands  
GENJCL.IC 231  
NOTIFY.IC 312  
CMDAUTH  
CHANGE.RECON 179  
CMDAUTH parameter  
INIT.RECON 281  
cold start  
commands  
CHANGE.BKOUT 134  
DELETE.BKOUT 203  
LIST.BKOUT 287  
multiple in a test environment 114  
command  
/DBDUMP database backup  
copies 37  
/RESTART  
restart after DBRC failure 25  
restart after IMS failure 24  
/NRESTART, restart after IMS failure 24  
/REPRO 73, 77  
/RMGENJCL 13  
/START 247  
/START command 17  
BACKUP.RECON 73  
CHANGE.PART 161  
CHANGE.PRILOG 8  
CHANGE.RECON 60  
CHANGE.RECON, recovering RECONs 77  
CHANGE.SECLOG 8  
DELETE.LOG 75  
GENJCL 13  
GENJCL.ARCHIVE 9  
GENMAX 40  
INIT.ADS 21  
INIT.CAGRPF 47  
INIT.DB 20  
INIT.DBDS 20, 40  
INIT.PART 275  
INIT.RECON 56  
INIT.RECON, establishing RECON  
data sets 20  
INIT.RECON, recovering RECONs 77  
NOTIFY.UIC 35  
online  
command (continued)  
valid DBRC log-related commands  
CHANGE.PRILOG 9  
CHANGE.RECON 9  
CHANGE.SECLOG 9  
CHANGE.SECLOG (for RLDS) 189  
DELETE.LOG 9  
GENJCL.ARCHIVE 9  
GENJCL.CLOSE 9  
LIST.LOG 9  
NOTIFY.PRILOG 9  
NOTIFY.SECLOG 9  
command authorization 79  
using DSPDCA 80  
using DSPDCA0 and RACF 80  
command syntax  
comment 121  
continuation characters 121  
definition 121  
description for DBRC utility 120  
parameters 121  
separators 121  
commands  
GENJCL.ICA 50  
GENJCL.OIC 50  
NOTIFY.IC  
HALDB considerations 50  
NOTIFY.UIC  
HALDB considerations 50  
COMP parameter commands  
CHANGE.CA 138  
NOTIFY.CA 310  
complex expressions 92  
compression  
PRILOG 74, 108  
concurrent image copy (CIC)  
CIC parameter 231  
registered database copies 38  
recovered database with DBRC 39  
restrictions 39  
considerations  
when using DBRC 33  
combinations, avoiding RECON 56  
continuation characters 121  
control group, skeletal JCL 88  
control keywords, skeletal JCL 88, 103  
CONTROLINTERVALSIZE keyword  
DEFINE CLUSTER keywords 61  
COPIES parameter commands  
GENJCL.ICA 230  
GENJCL.OIC 239  
coupling facility structure (CFSTR1)  
270  
CURRENT parameter commands  
NOTIFY.RECOV 326  
NOTIFY.REORG 329  
NOTIFY.UIC 343  
CYLINDERS keyword  
DEFINE CLUSTER keywords 61  
D  
damaged RECON data set 76  
DASD UNIT parameter commands  
CHANGE.RECON 179  
INIT.RECON 281  
data set allocation, RECON data set  
creating 61
data set allocation, RECON data set
(continued)
shared among multiple processors 57
data set, RECON  55
data sets
A-through-J  50
M-through-V  50
data sharing  16
assigning a sharing level with
DBRC  17
batch backout  22
database integrity  17
dynamic backout  22
forward recovery  24
information in the RECON data
set  17
levels of  16
block level  16
database level  16
merging logs  47
online reorganization  52
planning for recovery  21
record  72
database
adding information to RECON  328
allocation record  72
authorization, changing  112
backup  35, 43
commands for  37
changing information  141
data set groups, changing
information  157
data set, changing information  149
defining  263
deleting information from
RECON  204
integrity in data sharing  17
listing  289
making HISAM copies  42
making image copies  35
open exit  16
record  67
database allocation record  72
Database Change Accumulation
utility  44
Database Change Accumulation utility
(DFSUCUM0) (continued)
CA group
defining  47
defining for future use  47
reusing  47
commands
BACKUP.RECON  131
CHANGE.ADS  133
CHANGE.BKOUT  134
CHANGE.CA  136
CHANGE.CAGRP  138
CHANGE.DBDSGRP  157
CHANGE.IC  159
CHANGE.PRILOG (for
OLDS)  166
CHANGE.PRILOG (for
RLDS)  168
CHANGE.PRILOG (for
SLDS)  172
CHANGE.PRILOG (for
TSLDS)  172
Database Change Accumulation utility
(DFSUCUM0) (continued)
commands (continued)
CHANGE.RECON  177
CHANGE.SECLOG (for
OLDS)  187
CHANGE.SECLOG (for
RLDS)  188
CHANGE.SECLOG (for SLDS and
TSLDS)  192
CHANGE.SG  197
CHANGE.SUBSYS  198
CHANGE.UIC  200
DELETE.ADS  201
DELETE.ALLOC  202
DELETE.BKOUT  203
DELETE.CA  203
DELETE.CAGRP  204
DELETE.DB  204
DELETE.DBDS  205
DELETE.DBDSGRP  206
DELETE.GSG  206
DELETE.IC  207
DELETE.LOG (for OLDS)  208
DELETE.LOG (for RLDS)  208
DELETE.LOG (for SLDS)  208
DELETE.RECOV  212
DELETE.REORG  212
DELETE.SG  213
DELETE.SUBSYS  214
DELETE.UIC  214
GENJCL.ARCHIVE  217
GENJCL.CA  221
GENJCL.CLOSE  225
GENJCL.IC  229
GENJCL.OIC  238
GENJCL.RECEIVE  243
GENJCL.RECOV  247
GENJCL.USER  253
INIT.ADS  259
INIT.CA  260
INIT.CAGRP  261
INIT.DB  263
INIT.DBDS  266
INIT.DBDSGRP  271
INIT.GSG  273
INIT.IC  274
INIT.RECON  280
INIT.SG  285
LIST.BKOUT  287
LIST.CAGRP  288
LIST.DB  289
LIST.DBDS  290
LIST.DBDSGRP  292
LIST.GSG  293
LIST.HISTORY  294
LIST.LOG (for a category of
records)  298
LIST.LOG (for a PRILOG
family)  296
LIST.RECON  300
NOTIFY.ALOC  305
NOTIFY.BKOUT  307
NOTIFY.CA  308
NOTIFY.IC  310
NOTIFY.PRILOG (for OLDS)  313
NOTIFY.PRILOG (for RLDS)  317
Database Change Accumulation utility
(DFSUCUM0) (continued)
commands (continued)
NOTIFY.PRILOG (for SLDS and
TSLDS)  322
NOTIFY.RECOV  326
NOTIFY.REORG  328
NOTIFY.SECLOG (for OLDS)  331
NOTIFY.SECLOG (for RLDS)  334
NOTIFY.SECLOG (for SLDS)  338
NOTIFY.UIC  342
RESET.GSG  345
description  43
execution recorded by DBRC  47
input  45
subset of log volumes  45
valid log subset with DBRC  47
database change accumulation utility JCL
adding information to RECON  308
changing information about a
run  136
deleting information from
RECON  203
generating a job  221
skeletal JCL  446
database data set record  68
Database Image Copy 2 utility
(DFPSUDMT0)  37
Database Image Copy utility
(DFPSUDP0)  37
adding information to RECON  310
creating data sets for future use  39
description  35
execution recorded by DBRC  35
generating a job  229
maximum number of generations  40
nonstandard image copy data sets  43
recovery period  40
reusing image copy data sets  41
skeletal JCL  451
database recovery
Database Image Copy 2 output
format  38
defining groups  271
groups  67
database recovery control
changing information  159
changing partition information  161
data set, creating a backup copy  131
Database Recovery Control
5e DBRC (Database Recovery
Control)
Database Recovery Control utility
(DSPURX00)
description  27
generating a job  247, 253
GENJCL.RECOV command  247
overview  27
database recovery records  66
database recovery with data sharing
batch  22
data sharing  24
dynamic  22
forward  24
managing system logs  47
DB Groups  49
DB header record
HALDB 69
DB partition
HALDB 69, 70
DB query
output 368
overview 363
parameters 364
return codes 366
syntax 363
DB.INIT command
defining databases 20
DBD parameter commands
CHANGE.ADS 134
CHANGE.BKOUT 136
CHANGE.DB 142
CHANGE.DBDS 151
CHANGE.IC 159
CHANGE.UIC 200
DELETE.ADS 201
DELETE.ALLOC 202
DELETE.DB 205
DELETE.DBDS 205
DELETE.IC 207
DELETE.PART 211
DELETE.RECOV 212
DELETE.REORG 213
DELETE.UIC 214
GENJCL.IC 230
GENJCL.OIC 238
GENJCL.RECEIVE 243
GENJCL.RECOV 248
GENJCL.USER 253
INIT.ADS 260
INIT.DB 263
INIT.DBDS 267
INIT.IC 274
LIST.DB 289
LIST.DBDS 291
LIST.HISTORY 294
NOTIFY.ALLOC 306
NOTIFY.BKOUT 307
NOTIFY.IC 310
NOTIFY.RECOV 326
NOTIFY.REORG 328
NOTIFY.UIC 342
DBDS (database data set) group
A-through-J, L and X 51
commands 248
GENJCL.RECOV 248
LIST.DB 290
NOTIFY.RECOV 327
defining 266
listing 290
M-through-V and Y 51
qualifier 90
RECON, deleting information
from 205, 207
selecting DBDS records 101, 102
DBDS (database data set) group
commands that affect the definition
CHANGE.DBDSGRP 48
DELETE.DBDS 48
DELETE.DBDSGRP 48
INIT.DBDSGRP 48
LIST.DBDSGRP 48
DBDS (database data set) group
(continued)
commands that specify
GENJCL.IC 48
GENJCL.OIC 48
GENJCL.RECEIVE 48
GENJCL.RECOV 48
GENJCL.USER 48
LIST.DBDS 48
LIST.HISTORY 48
defining 271
deleting information from
RECON 206
LIST command 292
record 67
using 48
DBDS, INIT command
defining databases 20
DBDS, INIT.DBDS command
specifying image copy requirements 40
DBRC (Database Recovery Control)
See also Database Recovery Control
utility (DSPURX00)
active and spare RECON 62
application programming interface
accessing RECON data sets 353
addressing and residence 350
coding parameters 351
DSPAPI macro 349
ending the environment 350
establishing the environment 350
how to access 349
overview 349
QUERY request 357
runtime considerations 353
sample application program 349
services available 349
tokens 352
use in an IMSplex 355
use of registers 350
using EQU statements 351
assigning a sharing level 17
authorization, changing database 112
BACKOUT query
output 361
return codes 360
calls from IMS 4
catalog management of data sets 113
change accumulation says nothing to
process, when 112
change accumulation stops processing
logs, when 112
closing an open online PRILOG 109
commands
INIT.CA 47
INIT.CAGRP 47
introduction 119
components 4
controlling database recovery 11, 14,
18
data sets
defining recovery
requirements 11
partitioned 5
RECON 74
RECON data set 4, 55
DBRC (Database Recovery Control)
(continued)
data sets (continued)
SYS record 72
use of output data set 355
database backup 35
Concurrent Image Copy 38
controlling the number of image
copies managed 39
creating image copy data sets for
future use 39
Database Image Copy
(DFSUDMP0) 37
Database Image Copy 2
(DFSUDMTO) 37
guidelines 43
HISAM copies 42
methods 35
Online Database Image Copy
(DFSUCICP0) 38
recovering a database 12
recovery period of image copy
data sets and GENMAX 40
reusing image copy data sets 41
Database Recovery Control utility
(DSPURX00) 4
DB query
output 368
overview 363
parameters 364
return codes 366
syntax 363
DBRC components 4
DBRC log control 7
DBRC parameter
IMSCTRL macro 19
DBRC procedure 20
defining to IMS 19
description 33
generating JCL 5
GENMAX, resetting 107
GROUP query
output 389
overview 386
parameters 386
return codes 388
syntax 386
IMS procedures and 19
initial RECON access 62
initializing the RECON data set 20
log control 29
log data sets 61
LOG query
output 399
overview 395
parameters 396
return codes 398
syntax 396
log records, deleting 110
maintaining RECON records 74
notifying that log data sets have
moved 113
OLDS query
output 406
overview 404
parameters 404
return codes 405
Index 575
DBRC (Database Recovery Control) (continued)

OLDS query (continued)
syntax 404
online commands 120
overview 3
parameters
IMSPROCCLIB execution parameter 19
partitioned data set members 5
QUERY request
BACKOUT 359
RECON data set
creating 61
shared among multiple processors 57
RECON data set, registering databases in 20
RECON status query
output 411
overview 409
parameters 409
return codes 410
syntax 409
RECON, defining recovery requirements 11
recording change accumulations 47
recording image copies 35
recovery utilities 14
RELBUF query
overview 419
return codes 420
syntax 419
request time-stamp format 354
share control 16
SLDS stop time, locating the last in RECON data set 106
specifying when it is to be used 19
STARTDBRC request
overview 423
parameters 423
return codes 424
syntax 423
STOPDBRC request
overview 427
parameters 427
syntax 427
SUBSYS query
output 417
overview 413
parameters 414
return codes 415
syntax 413
subsystem (SSYS) records 111
system, considerations for a 33
tailoring JCL 9
time stamp 124
tips 105
variables 61
when should you use it 5
DBRC (Database Recovery Control) commands
Database Recovery Control utility (DSPURX00) 4
DBRC Command Authorization exit routine (DSPDCA0) 80
DBRC command authorization support 79
using a security product 80
DBRC Command Authorization Support using both DSPDCA0 and RACF 80
using DSPDCA0 80
DBTRACK parameter
CHANGE.DB command 146
CHANGE.DBDS command 155
DDN parameter commands
CHANGE.DBDS 151
CHANGE.UIC 159
CHANGE.UIC 200
DELETE.ALLOC 202
DELETE.DBDS 205
DELETE.IUC 207
DELETE.RECOV 212
DELETE.REORG 213
DELETE.UIC 215
GENJCL.IC 231
GENJCL.UIC 239
GENJCL.RECEIVE 244
GENJCL.RECOV 248
GENJCL.USER 254
INIT.DBDS 267
INIT.IUC 274
LIST.DBDS 291
LIST.HISTORY 295
NOTIFY.ALLOC 306
NOTIFY.IUC 311
NOTIFY.RECOV 326
NOTIFY.REORG 329
NOTIFY.UIC 343
DDNNEW parameter
CHANGE.DBDS command 152
deallod, avoiding RECON data set 57
deallocation, changes to processing 16
DEALTIME parameter command
NOTIFY.ALLOC 306
DEDB Fast Path record 70
default members
See skeletal JCL
DEFAULTS parameter commands
GENJCL.ARCHIVE 218
GENJCL.CA 222
GENJCL.CLOSE 226
GENJCL.IC 233
GENJCL.UIC 239
GENJCL.RECEIVE 244
GENJCL.RECOV 248
GENJCL.USER 254
DEFINE CLUSTER keywords 61
CONTROLINTERVALSIZE 61
CYLINDERS 61
FREESPACE 61
INDEXED 61
KEYS 61
NAME 62
NOWRITECHECK 62
RECORDSIZE 62
SHAREOPTIONS 62
SPEED 62
defining
DB groups 271
DBDS groups 271
recovery groups 271
DELFTJCL parameter
CHANGE.CAGR P 139
CHANGE.DBDS 152
INIT.CAGR P 262
INIT.DBDS 268
DELEQE parameter CHANGE.DBDS command 151
delete group 88
delete log records, how to 110
DELETE parameter commands
CHANGE.BKOUT 135
CHANGE.CAGR P 139
DELETE.ADS command 201
DELETE.ALLOC command 202
DELETE.BKOUT command 203
DELETE.CA command 203
DELETE.CAGR P command 204
DELETE.DB command 204
DELETE.DBDS command 205
DELETE.DBDSGRP command 206
DELETE.GSG command 206
DELETE.IUC command 207
DELETE.LOG (for OLDS) command 208
DELETE.LOG (for RLDS) command 208
DELETE.LOG (for SLDS) command 208
DELETE.PART command
DBD parameter 211
PART parameter 211
syntax 211
DELETE.RECOV command 212
DELETE.REORG command 212
DELETE.GSG command 213
DELETE.SUBSYS command 214
DELETE.UIC command 214
deleting a SSYS record 111
deleting information
all change accumulation group records 204
all database data set records 205
all database records 204
allocation record of database data set 202
area data set 201
backup record 203
change accumulation run record 203
database data set group records 206
global service group records 206
image copy data set records 207
log records 110
nonstandard image copy data sets 214
online log data set records 208
recovery log data set records 208
recovery run record 212
reorganization records 212
service group records 213
subsystem records 214
system log data set records 208
deleting unnecessary RECON records 75
DELMEM parameter,
CHANGE.DBDSGRP command 157
DEQ macro 58
DFSUARCO (Log Archive utility) description 8
DSN commands (continued)
NOTIFY.SECLOG (for RLDS) 335
NOTIFY.SECLOG (for SLDS) 338

DSPI
accessing 353
forms
execute 353
list 352
modify 353
standard 352
versions 352

DSPAPI macro
overview 349

DSPAPSMP sample DBRC application program 349

DSPDBHRC 69
partition DB record
HALDB 69

DSPDCAX0 80

DSPDSHRRC
partition DBDS records
HALDB 70

DSPPTNRRC
HALDB partition record
HALDB 69

DSPSCIX0 78

DSPSCIX0 exit 181

DSPURREX0 Database Recovery Control utility
See Database Recovery Control utility
(DSPURREX0)

DSSN parameter NOTIFY.ALLOC
command 306

DSSTART parameter commands
CHANGE.PRLOG (for RLDS) 170
CHANGE.PRLOG (for SLDS) 174
CHANGE.SECLOG (for RLDS) 190
CHANGE.SECLOG (for SLDS) 194

DUAL parameter CHANGE.RECON
command 179
dump format 38
dynamic allocation 60
of RECON data set 60
dynamic backout with data sharing 22

ENDRECOV parameter commands
CHANGE.SUBSYS 199
NOTIFY.SUBSYS 342
enqueue problems, causes of
RECON 115
error, RECON 1/O 76
execution members
See JCL (job control language), skeletal
execution members explained
exit
database open 16
RECON 1/O 81

F
failure and restart with data sharing 21
Fast Path
registering databases and DEDB
areas 17

Fast Path DEDB record 70
FILESEQ parameter commands
CHANGE.CA 137
CHANGE.IC 160
CHANGE.PRLOG (for RLDS) 170
CHANGE.PRLOG (for SLDS) 174
CHANGE.SECLOG (for RLDS) 190
CHANGE.SECLOG (for SLDS) 194
INIT.CA 261
INIT.IC 274
NOTIFY.CA 309
NOTIFY.IC 311
NOTIFY.PRLOG (for RLDS) 319
NOTIFY.PRLOG (for SLDS and
TSLDS) 324
NOTIFY.REORG 329
NOTIFY.SECLOG (for RLDS) 335
NOTIFY.SECLOG (for SLDS or
TSLDS) 339

FILESEQ parameter commands
CHANGE.CA 160
INIT.CA 274
NOTIFY.CA 311
NOTIFY.REORG 329

FORCER parameter commands
CHANGE.RECON 180
INIT.RECON 282

forward recovery with data sharing 24

FPAREA parameter, GENJCL.RECOV
command 248

FREESPACE keyword
DEFINE CLUSTER keywords 61
FROMTIME parameter commands
LIST.HISTORY 295
LIST.LOG 299

functions of DBRC
data sharing 7
database recovery 7
logs 7
fuzzy image copies
definition of 36

G

GAP parameter, CHANGE.PRLOG (for
RLDS) 170, 171
Generalized Trace Facility (GTF) USR
records 184
generating a job
Change Accumulation utility 221
Database Image Copy utility 229
Database Recovery utility 247, 253
Log Archive utility 217
Log Recovery utility 225
Online Database Image Copy
utility 238
generating JCL 83
generating user output 83
GENJCL command, generating JCL 13
GENJCL commands, description 83
GENJCL.ARCHIVE command
log control requirements in
RECON 9
syntax 217
TIMEFMT parameter 219
GENJCLCA command 221
TIMEFMT parameter 223
GENJCL.CLOSE command 225
TIMEFMT parameter 227
GENJCL.DDD 37
HALDB online reorganization 50
GENJCL.INCLUDE command 229
TIMEFMT parameter 234
GENJCL.REC LV 37
HALDB online reorganization 50
GENJCL.RECEIVE command 238
TIMEFMT parameter 241
GENJCL.USER command 243
TIMEFMT parameter 245
GENJCL.RECOV command 247
parameters
USEAREA 251
USEDBDS 251
USEIC 251
USEOLRGG 251
RESTORE parameter 250
TIMEFMT parameter 250
GENJCL.USER command 253
TIMEFMT parameter 255
GENMAX parameter
image copy data sets for future use 39
of the INIT.DBDS command 267
resetting 107
specifying image copy requirements 40
Global Resource Serialization macro 57
global service group
deleting information 206
listing 293
record 71
resetting 345
GROUP parameter commands 230
GENJCL.CA 230
GENJCL.OIC 238
GENJCL.RECEIVE 243
GENJCL.RECOV 248
GENJCL.USER 253
LIST.DBDS 291
LIST.HISTORY 294
GROUP query
output 389
overview 386
parameters 386
return codes 388
syntax 386
GRPMAX parameter commands
CHANGE.CAGRP 139
INIT.CAGRP 261
GRPMEM parameter INIT.CAGRP command 262
GRPNAME parameter commands
CHANGE.CA 137
CHANGE.CAGRP 139
CHANGE.DBDSGRP 157
DELETE.CA 203
DELETE.CAGRP 204
DELETE.DBDSGRP 206
GENJCL.CA 222
INIT.CA 261
INIT.CAGRP 262
INIT.DBDSGRP 272
LIST.CAGRP 288
LIST.DBDSGRP 292
GRPNAME parameter commands (continued)
NOTIFY.CA 308
GRS macro 57
GSG parameter commands
CHANGE.PRILOG (for RLDS) 170
CHANGE.PRILOG (for SLDS) 174
CHANGE.SECCLOG (for RLDS) 190
CHANGE.SECCLOG (for SLDS) 194
GSGNAME parameter commands
CHANGE.DB 143
CHANGE.DBDS command 153
GTFC (Generalized Trace Facility) USR records 184

HALDB
DBRC commands supported 49
master (DSPDBHRC)
DB header record 69
OLR coexistence support 51
partition 69
partition DB record (DSPDBHRC) 69
partition DBDS records
(DSPDBHRC) 70
partition record (DSPPTNRC)
PHDAM 69
types of DBDS 70
HALDB partition
registering 275
header record, RECON data set 64
hints and tips
using DBRC 105
adjusting GENMAX when it is reached or is too high 107
locating the last SLDS Stop Time in RECON 106
PRILOG compression not working 108
PRILOG record sizes 109
HISAM Reorganization Unload utility (DSPSUICP)
for backup 42
HISTORY command 294
HSSP data set 42
database registered with DBRC 42
ICDNSN parameter commands
CHANGE.IC 160
creating for future use 39
defining 274
duplicate, naming convention 34
INIT.IC 274
maximum number of generations 40
naming convention 34
nonstandard 43
NOTIFY.IC 311
NOTIFY.REORG 330
RECON
adding information 326
changing information 159
record 71
recovery period 40
ICDNSN parameter commands (continued)
reusing 41
selecting 99
ICDNSN parameter commands
CHANGE.IC 160
DELETE.IC 207
INIT.IC 275
NOTIFY.IC 312
NOTIFY.REORG 330
ICJCL parameter commands
CHANGE.DBDS 153
GENJCL.CA 222
GENJCL.CLOSE 226
GENJCL.OIC 233
GENJCL.RECEIVE 244
GENJCL.RECOV 249
GENJCL.USER 254
INIT.DBDS 268
skeletal JCL execution member 451
ICOFF parameter CHANGE.DBDS
command 153
ICON parameter CHANGE.DBDS
command 153
ICRCVJCL parameter skeletal JCL
execution member 459
ICREQ parameter CHANGE.DB
command 144
ICREQ parameter INIT.DB
command 264
ILDS (Indirect List Data Set) 48
Index/ILDS Rebuild Utility
(DFPSRECO) 14
image copies
fuzzy 36
image copy 2 JCL 451
image copy data set
recovery period 40
Image Copy Group record 71
Image Copy utility
See also Database Image Copy utility
(DFSUDDMP)
See Online Database Image Copy
utility (DFSUICP)
IMS recovery utilities 14
IMS Structured Call Interface (SCI) 78
IMS.PROCLIB 5
IMS.SDFSISRC 5
IMSCtrl macro
archiving OLDS 9
parameters 19
IMSpelx
use of DSPAPI 355
IMSPLEX
CHANGE.RECON 181
DSPSCIX0 181
IMSpelx, DBRC support of 18
INACTIVE parameter DELETE.LOG (for
RLDS and SLDS) command 209
Index/ILDS Rebuild utility
(DFPSRECO) 14
INDEXED keyword
DEFINE CLUSTER keywords 61
INIT.ILDS command
description 259
INIT.CA command 260
INIT.CAGRP command 261
INIT.DB command description 263
INIT.DBDS command description 266
REUSE keyword 41
INIT.DBDSGRP command 271
INIT.GSG command 273
INIT.IC command 274
INIT.PART command 275
INIT.RECON command description 280
establishing RECON data sets 20
initializing the RECON data set 56
recovering RECONS 77
INIT.GSG command 285
initializing DBRC DBRC procedure 20
IMSCTRL Macro 19
initializing the RECON data set 20
input/output error processing 76
INTERIM parameter commands
DELETE.LOG (for OLDS) 208
DELETE.LOG (for RLDS and SLDS) 210
NOTIFY.PRILOG (for OLDS) 315
NOTIFY.PRILOG (for RLDS) 319
NOTIFY.PRILOG (for SLDS and TSLDS) 324
NOTIFY.SECLOG (for OLDS) 333
NOTIFY.SECLOG (for RLDS) 336
NOTIFY.SECLOG (for SLDS or TSLDS) 340
INVALID parameter commands
CHANGE.CA 137
CHANGE.IC 160
INVALID2 parameter
CHANGE.IC command 160
description 160
IRLM (Internal Resource Lock Manager)
data sharing 17
database integrity 17
with batch backout 22
IRLMID parameter commands
CHANGE.SUBSYS 198
NOTIFY.SUBSYS 341
JCL (job control language)
generated by DBRC 5
generating 83
skeletal execution members explained 5
tailoring for utilities 9
JCL allocation 60
JCLOUT parameter command GENJCL.ARCHIVE 218
skeletal JCL 84
JCLPDS parameter commands 218
GENJCL.ARCHIVE 218
GENJCL.CA 222
GENJCL.CLOSE 226
GENJCL.IC 233
GENJCL.OIC 240
GENJCL.RECEIVE 244
GENJCL.RECOV 249
GENJCL.USER 254
skeletal JCL 84
job control language (JCL)
See JCL (job control language)
JOB parameter commands
GENJCL.ARCHIVE 218
GENJCL.CA 222
GENJCL.CLOSE 226
GENJCL.IC 233
GENJCL.OIC 240
GENJCL.RECEIVE 244
GENJCL.RECOV 249
GENJCL.USER 254
skeletal JCL execution member 13, 440
K
KEYS keyword
DEFINE CLUSTER keywords 61
keyword
%IC 456
%ICDSN2 456
%ICDSN3 456
%ICDSN4 456
%ICUNIT2 456
%ICUNIT3 456
%ICUNIT4 456
%ICVCNT2 456
%ICVCNT3 456
%ICVCNT4 456
%ICVOLS2 456
%ICVOLS3 456
%ICVOLS4 456
DEFINE CLUSTER 61
substitution 86
keywords, symbolic
See symbolic keywords, JCL execution member
L
level of sharing definitions 17
LIST parameter commands
GENJCL.ARCHIVE 219
GENJCL.CA 222
GENJCL.CLOSE 227
GENJCL.IC 233
GENJCL.OIC 240
GENJCL.RECEIVE 245
GENJCL.RECOV 249
GENJCL.USER 254
LIST.BKOUT command 287
LIST.CAGRP command 288
LIST.DB command 289
LIST.DBDS command 290
LIST.DBDSGRP command 292
LIST.GSG command 293
LIST.HISTORY command 294
LIST.LOG (for a category of records) command 298
LIST.LOG (for a PRILOG family) command 296
LIST.RECON command description 300
fields displayed by command 529
sample listing 469
LIST.SUBSYS command 302
listing
backout records 287
category of records 298
change accumulation group 288
database 289
database data set 290
database groups 292
database histories 294
global service group 293
PRILOG family 296
RECON 300, 469
subsystem 302
LKASID parameter
CHANGE.DBDS command 154
INIT.DBDS command 270
Local Shared Resources (LSR) option 61
log allocation record 72
Log Archive utility (DFSUARCC0) description 8
log control
valid API requests
Log query 10
OLDS query 10
valid commands
CHANGE.PRILOG 9
CHANGE.RECON 9
CHANGE.SECLOG 9
DELETE.LOG 9
GENJCL.ARCHIVE 9
GENJCL.CLOSE 9
LIST.LOG 9
NOTIFY.PRILOG 9
NOTIFY.SECLOG 9
log data sets, records 65, 66
LOG query
output 399
overview 395
parameters 396
return codes 398
syntax 396
log records, deleting 110
Log Recovery utility (DFSULTR00)
generating a job 225
log volumes, specifying
a subset 45
all 45
LOGCLJCL skeletal JCL execution member 449
logging
accumulating logs using DFSUCUM0 43
logging (continued)
condensing logs using
DFUSUCUM0 43
LOGRET parameter
INIT.RECON 282
LOGRET parameter of
CHANGE.RECON 182
LSR (Local Shared Resources option) 61

M
macro
DEQ 58
DFP Record Management Services 58 DSPAPI
accessing 353
forms 352
overview 349
versions 352
GRS 57
IMSCtrl 9
OBTAIN 58
RESERVE 57
maximum number of generations, image
copy data sets, GENMAX
parameter 153, 267
MAXOLDS parameter
GENJCL.ARCHIVE command 219
MEMBER parameter
GENJCL.ARCHIVE command 219
GENJCL.CA 223
GENJCL.CLOSE 227
GENJCL.IC 233
GENJCL.OIC 240
GENJCL.RECEIVE 245
GENJCL.RECOV 249
GENJCL.USER 253
INIT.DBDSGRP 272
members
See skeletal JCL
merging logs 47
MULTIOBJ parameter commands
GENJCL.CA 233
GENJCL.OIC 240
GENJCL.RECOV 249
GENJCL.USER 255
multiple
cold starts in a test environment 114

N
NAME keyword
DEFINE CLUSTER keywords 62
naming conventions
change accumulation data sets 34
DBRC data sets 33
duplicate image copy data sets 34
image copy data sets 34
SSIDs in RECON SSYS records, for 111
SSIDs processed by batch backout, for 111
NEWTIME parameter commands
CHANGE.PRILOG (for RLDS) 170
CHANGE.PRILOG (for SLDS) 174
CHANGE.SECLOG (for RLDS) 190
NEWTIME parameter commands
(continued)
CHANGE.SECLOG (for SLDS) 194
NEWVOL parameter commands
CHANGE.PRILOG (for RLDS) 170
CHANGE.PRILOG (for SLDS) 175
CHANGE.SECLOG (for RLDS) 190
CHANGE.SECLOG (for SLDS) 195
NOAUTH parameter
CHANGE.DB 143
CHANGE.DBDS 152
NOBACK parameter CHANGE.DB command 144
NOBACKUP parameter
CHANGE.SUBSYS command 198
NOCATDS
CHANGE.RECON 178
NOCATDS parameter commands
CHANGE.RECON 177
INIT.RECON 281
NOCFSTR2 parameter CHANGE.DBDS command 152
NOCHECK parameter
CHANGE.RECON 181
INIT.RECON 282
NODEFAULT parameter commands
CHANGE.CAGRP 139
GENJCL.CA 223
GENJCL.IC 234
GENJCL.OIC 240
GENJCL.RECOV 250
GENJCL.USER 255
NOFORCER parameter commands
CHANGE.RECON 180
INIT.RECON 282
NOICREQ parameter CHANGE.DB command 144
NOICREQ parameter INIT.DB
command 264
NOJOB parameter commands
GENJCL.ARCHIVE 218
GENJCL.CA 222
GENJCL.CLOSE 226
GENJCL.IC 233
GENJCL.OIC 240
GENJCL.RECEIVE 244
GENJCL.RECOV 249
GENJCL.USER 254
NOLIST parameter commands
GENJCL.ARCHIVE 219
GENJCL.CA 222
GENJCL.CLOSE 227
GENJCL.IC 233
GENJCL.OIC 240
GENJCL.RECEIVE 245
GENJCL.RECOV 249
GENJCL.USER 254
NOKASID parameter
CHANGE.DBDS command 154, 270
NONEW parameter commands
CHANGE.RECON 183
INIT.RECON 284
NONRECOV parameter,
CHANGE.DB 145
nonstandard image copy data sets
description 43
nonstandard image copy data sets
(continued)
RECON
adding information 342
changing information 200
deleting information 214
NONEWER parameter 146
NOPREL parameter CHANGE.DBDS
command 154
NOPREQ parameter CHANGE.DBDS
command 155
NORECOV parameter
NOTIFY.REORG 330
NORECOV parameter CHANGE.DBDS
command 155
NOREORGI parameter
CHANGE.DB 149
NOREUSE parameter
CA data set 47
CHANGE.DBDS 154
command
command 152
CHANGE.CAGRP 140
image copy data sets for future
use 39
INIT.CAGRP 262
INIT.DBDS 268
NORMAL parameter commands
CHANGE.PRILOG (for OLDS) 167
CHANGE.PRILOG (for RLDS) 170
CHANGE.PRILOG (for SLDS) 174
CHANGE.SECLOG (for OLDS) 187
CHANGE.SECLOG (for RLDS) 190
CHANGE.SECLOG (for SLDS) 194
CHANGE.SUBSYS 199
NOTIFY.SUBSYS 342
normal restart 24
NOTCOVER parameter
CHANGE.DB command 143
CHANGE.DBDS command 153
nothing to process, when change
accumulation says 112
NOTIFY.ALLOC command 305
NOTIFY.BKOUT command 307
NOTIFY.CA command 308
NOTIFY.IC command
HALDB considerations 50
NOTIFY.PRILOG (for OLDS)
command 313
NOTIFY.PRILOG (for RLDS)
command 317
NOTIFY.PRILOG (for SLDS and TSLDS)
command 322
NOTIFY.RECOV command 326
NOTIFY.REORG command 328
NOTIFY.SECLOG (for OLDS)
command 331
NOTIFY.SECLOG (for RLDS)
command 334
NOTIFY.SECLOG (for SLDS)
command 338
NOTIFY.SECLOG command 341
NOTIFY.UIC command
description 342
HALDB considerations 50
notifying DBRC that log data sets have
moved 113
NOVSO parameter 156
NOVSO parameter (continued)
CHANGE.DBDS 156, 270
NOWRITECHECK keyword
DEFINE CLUSTER keywords 62
NXTOLDS parameter commands
NOTIFY:PRILOG (for OLDS) 316
NOTIFY:SECLOG (for OLDS) 333

O

OBTAIN macro 58
OFFLINE parameter 330
OICJCL parameter
commands 154
CHANGE.DBDS 154
INIT.DBDS 269
skeletal JCL execution member 457
OLDS (online log data set)
See also logging
archiving DBRC 8
changing information 166, 187
commands
CHANGE.PRILOG 167
CHANGE.SECLOG 187
DELETE.LOG (for OLDS) 208
GENJCL.ARCHIVE 217
GENJCL.CLOSE 227
NOTIFY:PRILOG (for OLDS) 314
NOTIFY:SECLOG (for OLDS) 332
RECON
adding information 313, 331
deleting information 208
records 65
selecting in JCL 94
OLDS query
output 406
overview 404
parameters 404
return codes 405
syntax 404
OLDVOL parameter commands
CHANGE.PRILOG (for RLDS) 171
CHANGE.PRILOG (for SLDS) 175
CHANGE.SECLG (for RLDS) 191
CHANGE.SECLG (for SLDS) 195
OLRACAP parameter 145, 266
OLRDDBS parameter 146
OLRIMSID parameter 146
OLRNOCAP parameter 145, 266
OLRRGOFF parameter 146
OLRRGON parameter 146
ONEJOB parameter commands
GENJCL.IC 233
GENJCL.OIC 240
GENJCL.RECOV 249
GENJCL.USER 255
online commands 120
Online Database Image Copy utility
(DFSUICCP) 38
creating data sets for future use 39
description 35
execution recorded by DBRC 35
generating a job 238
Online Database Image Copy utility JCL
See OICJCL parameter
online log data set
See OLDS (online log data set)
online log data set (OLDS)
records 65
ONLINE parameter 330
ONLINE parameter commands
LIST:SUBSYS 302
NOTIFY:IC 312
NOTIFY:SUBSYS 342
online reorganization
data sharing 52
DBRC commands supported 49
maintaining recovery records 52
OLR coexistence support 51
process description 53
recommendations 54
records used 50
REORG records 52
restrictions 54
RSR tracking system 52

P

parameter
CIC 38
DBRC for online IMS 19
GENMAX 39
NOREUSE 39, 47
record type 89
RECOVPD 40
REUSE 39
selection criteria 90
STARTNEW 77
parameters
coding for DSPAPI macro 351
DBRC application programming
interface 351
NOOWNER 146
NORECOV 330
OFFLINE 330
OLRACAP 145, 266
OLRDDBS 146
OLRIMSID 146
OLRNOCAP 145, 266
OLRRGOFF 146
OLRRGON 146
ONLINE 330
RECOV 330
STOPTIME 330
PART parameter
DELETE.PART 211
partition DB record (DSPDBHRC)
HALDB
 TYPE=PART 69
partition DBDS records (DSPDSHRC)
HALDB
types of DBDSs 70
partition record (DSPPTNRC)
HALDB
PHDAM 69
PDS members 5
PRELOAD parameter CHANGE,DBDS
command 154
PREOPEN parameter CHANGE,DBDS
command 155
PRILOG
closing an open online 109
compression 108
compression, automatic 74
PRILOG family
listing records 296
procedures, installation 19
PSB parameter commands 136
CHANGE.BKOUT 136
GENJCL.OIC 239
GENJCL.USER 255
NOTIFY:GBKOUT 307
purge time, defined 46
PURGLIST parameter NOTIFY:CA
command 309

Q

QUERY request
general format of output 358
output
BACKOUT 361
DB 368
GROUP 389
LOG 399
OLDS 406
RECON status 411
SUBSYS 417
output from 337
overview 337
SUBSYS 413
parameters
BACKOUT 359
DB 364
GROUP 386
LOG 396
OLDS 404
RECON status 409
SUBSYS 414
return codes
BACKOUT 360
DB 366
GROUP 388
LOG 398
OLDS 405
RECON status 410
SUBSYS 415
syntax
BACKOUT 359
DB 363
GROUP 386
LOG 396
OLDS 404
RECON status 409
SUBSYS 413
types
BACKOUT 359
DB 363
GROUP 386
LOG 398
OLDS 405
RECON status 410
SUBSYS 415

R

RCVTIME parameter
GENJCL.RECOV 250
NOTIFY:RECOV 326, 327
RCVTTRACK parameter
CHANGE,DB command 146
RCVTACK parameter (continued)
CHANGE.DBDS command 155
READOFF.DBDS command 147
READON parameter CHANGE.DB command 147
RECDCT parameter commands CHANGE.IC 161
NOTIFY.REORG 330
RECON
allocation 55
DSECTS 469
initialization 56
sample listing 469
RECON data set
accessing with DSPAPI 353
active 62
ALLOC record 72
availability 56
backup 73
changing log control records 8
concurrent image copy data set 39
contention problems 56
creating 61
data sharing record types 72
data-sharing records 17
Database Image Copy 2 data set 37
defining 55
defining requirements in 11
description 55
dynamic allocation 60
enqueue problems, causes of 115
extending 60
header records 64
HSSP image copy data set 42
initial access 62
LOGALL record 72
maintaining 73
maintaining records 74
overview 55
preserving integrity of 79
records, maintaining 74
recovering 76, 77
recovery record types 66
registering databases 20
REORG record 71
reorganizing 75
replacing a discarded 78
replacing damaged 76
retrieving log-related information from 9
security considerations 62
serialization 57
shared among multiple processors 57
spare 62
subsystem (SSYS) record 72
tracking changes to 81
upgrading 27
version migration 27
VSAM CREATE mode 60
RECON data sets, both are unusable 77
RECON I/O exit routine 81
RECON initialization token (RIT) 16
RECON Loss Notification 78
RECON record types 64
RECON status query
output 411
overview 409
parameters 409
return codes 410
syntax 409
RECON1 parameter, BACKUP:RECON
record log information 7
record type parameter 89
records
BACKOUT 66
change accumulation group 66
change accumulation run 66
data sharing 72
database 67
data allocation 72
data base data set 68
data base data set group 67
data base recovery 66
DBDS group 67
DEDB 70
deleting unnecessary 74
global service group 71
HALDB online reorganization 50
image copy 71
in RECON data set 64
log allocation 72
log data set 65
maintaining RECON 74
online reorganization 50
RECON data set header 64
recovery 72
reorganization 71
subsystem 72
RECORDSIZE keyword
DEFINE CLUSTER keywords 62
RECOV parameter
NOTIFY.REORG 330
RECOVBL parameter
CHANGE.DB 145
INIT.DB 265
RECOVER commands 14
recoverable databases
CHANGE.DB command 36
Image Copy utilities 36
INIT.DB command 36
recovering a database
archiving log records 9
batch support 22
DBRC role 12
dynamic backup 22
process 12
recovery facilities 22
setting up recovery mechanisms 21
without DBRC 24
Recovering the RECON
RECON Loss Notification 78
recovery
CHANGE.DBDS 155
corcepts 10
data base data set or area, adding information to RECON 326
DBRC 33
defining groups 271
description 77
groups 67
recovery (continued)
in data sharing
planning procedures 21
without DBRC 24
log data set
deleting information 208
log data set, adding information to RECON 317, 334
maintaining records 52
overview 3
period, of image copy data sets 40
point-in-time 11
process overview 10
RECON data set 76
record 72
time stamp 11
recovery control
data set
commands to generate JCL and user-defined output 83
listing 30
recovery control data set
See also RECON
I/O error processing 76
records 64
replacing discarded RECON 78
recovery functions for DBDSs 7
recovery log data set
See RLS (recovery log data set)
recovery log data set (RLDS)
records 65
recovery period, definition of 40
recovery records, database 66
recovery utilities 14
RECOVJCL parameter
commands
CHANGE.DBDS 156
INIT.DBDS 269
skeletal JCL execution member 461
RECOVPD parameter 40
INIT.DBDS 269
RECOVPD parameter commands
CHANGE.DBDS 156
INIT.DBDS 269
RECITEM parameter commands
CHANGE.CA 137
CHANGE.IC 160
CHANGE.UIC 200
DELETE.ALOC 202
DELETE.CA 204
DELETE.IC 207
DELETE.RECOV 212
DELETE.REORG 213
DELETE.UIC 215
RECOVJCL parameter CHANGE.DBDS
command 156
registering databases in the RECON data set 20
RELBUF query
overview 419
RELBUF request
parameters 419
return codes 420
syntax 419
Remote Site Recovery (RSR), DBRC
support
See RSR
Removing an SSYS from DB 112
Authorization 112
REORG record 52
fields used in online reorganization 52
RECON data set 71
reorganization of the RECON data set 328
deleting information 212
reorganization record 71
reorganizing RECON, using CHANGE.RECON 75
REORG parameter
CHANGE.DB 149
REPLACE parameter CHANGE.RECON command 179
REPRO function of VSAM AMS, offline reorganization 75
requests
DBRC Query 357
backout 359
database 363
group 386
log 395
OLDs 404
RECON status 409
subsystem 413
DBRC Release Buffer 419
return codes 420
syntax 419
DBRC Start Request
overview 423
parameters 423
return codes 424
syntax 423
DBRC Stop Request
overview 427
parameters 427
syntax 427
RESERVE command, during backup 73
RESERVE macro 57
RESET.GSG command 345
resetting the GENMAX parameter 107
restart in data sharing
after an IMS failure 24
after DBRC failure 25
emergency, after DBRC failure 25
emergency, after IMS failure 24
normal 24
RESTORE parameter
GENJCL.RECOV 250
restrictions
concurrent image copy (CIC) 39
online reorganization 54
REUSE parameter commands
CHANGE.CAGRP 140
CHANGE.DBDS 154
INIT.CAGRP 262
INIT.DBDS 41, 268
REUSE parameter, image copy data sets for future use 39
reusing image copy data sets 41
RIT (RECON initialization token) 16
RLDS (recovery log data set) 168
accumulating changes using DSUSCUMO 43
commands 169
RLDS (recovery log data set) (continued)
CHANGE.PRILOG (for RLDS) 169
CHANGE.SECCLOG (for RLDS) 188
NOTIFY.PRILOG (for RLDS) 320
NOTIFY.SECCLOG (for RLDS) 337
deleting information from RECON 208
records 65
selecting 97
ROLB call 22
RSR 18
online reorganization 52
RUNTIMES parameter commands
CHANGE.PRILOG (for RLDS) 171
CHANGE.PRILOG (for SLDS) 175
CHANGE.SECCLOG (for RLDS) 191
CHANGE.SECCLOG (for SLDS) 195
NOTIFY.CA 308
NOTIFY.IC 311
NOTIFY.PRILOG (for OLDS) 314
NOTIFY.PRILOG (for RLDS) 318
NOTIFY.PRILOG (for SLDS) 322
NOTIFY.PRILOG (for TSLDS) 322
NOTIFY.REORG 329
NOTIFY.SECCLOG (for OLDS) 332
NOTIFY.SECCLOG (for RLDS) 335
NOTIFY.SECCLOG (for SLDS) 338
NOTIFY.UIC 343
SAMPLE sample DB2 application program (DSPAPSM) 349
sample listing of RECON
Active Site 475
tracking site 504
security, command 79
select group, skeletal JCL 88
selection criteria parameter 90
serialization of RECON data set 57
strategies 58
service group changing information 197
deleting information 213
service utilities control statement parameters 89
share control 7
share level, in DBRC 17
SHARE_LVL parameter
CHANGE.DB 147
INIT.DB 275
specification descriptions 17
SHAREOPTIONS keyword
DEFINE CLUSTER keywords 62
sharing level, assigning one with DBRC 17
shortcut keys
keyboard xv
size calculation for SSYS record 112
skeletal JCL 459
coding execution members 86, 103
data set 85
default members explained 85
skeletal JCL (continued)
exeuction members
ARCHJCL 440
CAJCL 446
ICJCL 451
ICRCVJCL 459
LOGJCL 449
OICJCL 457
RECOVJCL 461
generating JCL and user-defined output 83
IBM-supplied 84, 439
writing your own 85, 105
SLDS (system log data set)
accumulating changes using DSUSCUMO 43
adding information to RECON 322, 338
CHANGE.PRILOG (for SLDS) 172
CHANGE.SECCLOG (for SLDS) 193
changing information 172, 192
deleting information from RECON 208
NOTIFY.PRILOG (for SLDS) 322
NOTIFY.SECCLOG (for SLDS) 338
records 65
selecting 95
SLDS stop time, locating the last in RECON data set 106
SMS concurrent image copy, SMSCIC parameter 231
SMSCIC (SMS concurrent image copy) 231
SMSCIC parameter command, GENJCLC. 231
spac requirements, RECON data set 55
specifying log retention intervals, CHANGE.RECON 182
SPEED keyword
DEFINE CLUSTER keywords 62
SSID parameter commands
CHANGE.BKOUT 135
CHANGE.DB 148
CHANGE.PRILOG (for OLDS) 168
CHANGE.PRILOG (for SLDS) 176
CHANGE.RECON 182
CHANGE.SECCLOG (for OLDS) 188
CHANGE.SECCLOG (for SLDS) 196
CHANGE.SUBSYS 198
DELETE.BKOUT 203
DELETE.LOG (for OLDS) 208
DELETE.SUBSYS 214
GENJCL.ARCHIVE 219
GENJCL.CLOSE 227
GENJCL.USER 255
INIT.RECON 283
LIST.BKOUT 287
LIST.LOG (for a category of records) 300
LIST.LOG (for a PRILOG family) 297
LIST.RECON 301
LIST.SUBSYS 302
NOTIFY.BKOUT 307
NOTIFY.PRILOG (for OLDS) 316
NOTIFY.PRILOG (for RLDS) 320
SSID parameter commands (continued)
NOTIFY:PRILOG (for SLDS and TSLDS) 325
NOTIFY:SECLOG (for OLDS) 333
NOTIFY:SECLOG (for RLDS) 337
NOTIFY:SECLOG (for SLDS) 340
NOTIFY:SUBSYS 341
standard form of time stamps, parameters of DBRC commands 124
START command
use of 17, 247
STARTDBRC request
overview 423
parameters 423
return codes 424
syntax 423
STARTTIME parameter commands
CHANGE:PRILOG (for RLDS) 169
CHANGE:PRILOG (for SLDS) 173
CHANGE:SECLOG (for RLDS) 188
CHANGE:SECLOG (for SLDS) 193
DELETE:PRILOG (for RLDS and SLDS) 209
NOTIFY:ALLOC 306
NOTIFY:PRILOG (for OLDS) 314
NOTIFY:PRILOG (for RLDS) 318
NOTIFY:PRILOG (for SLDS and TSLDS) 323
NOTIFY:SECLOG (for OLDS) 332
NOTIFY:SECLOG (for RLDS) 335
NOTIFY:SECLOG (for SLDS) 339
STARTNEW parameter commands
CHANGE:RECON 183
INIT:RECON 284
usage 77
STARTRCV parameter commands
CHANGE:SUBSYS 199
NOTIFY:SUBSYS 342
STATUS parameter commands LIST:RECON command 301
STOPDBRC request
overview 427
parameters 427
syntax 427
STOPTIME 330
STOPTIME parameter commands
CHANGE:IC 161
NOTIFY:CA 308
NOTIFY:IC 312
SUBSET parameter commands
CHANGE:CA 138
NOTIFY:CA 310
SUBSYS query
output 417
overview 413
parameters 414
return codes 415
syntax 413
subsystem
changing information 198
listing 302
RECON 214
adding information 341
deleting information 214
subsystem (SYS) record (continued)
initializing during IMS restart 72
size calculation 112
symbolic keywords
See also symbolic keywords, JCL execution member
control keywords 86
%DELETE 88, 91
%ENDDE 88, 91
%ENDSE 88
%SELECT 88, 94
%SET MEMBER 88, 92
%SET TIMEFMT 88, 93
description 85, 88, 103
JCL execution member
%ALLSEL 102
%ALTIME 102
%ALLUSID 102
%CADSN 100
%CAFSEQ 101
%CALCSTM 101
%CAOESN 92
%CASEL 101
%CATIME 101
%CAUNIT 101
%CAVCNT 101
%CAVOLS 101
%DALTIME 102
%DBADDN 103
%DBADSAV 103
%DBDDN 102
%DBDNSDEL 103
%DBDNSN 103
%DBDNSNRV 103
%DBNAME 102
%DBTYPE 102
%DBUSID 103
%DDNAME 89
%ICAT 100
%ICDSN 99
%ICFSEQ 99
%ICSEL 99
%ICTIME 99
%ICTYPE 99
%ICUNIT 99
%ICUSID 100
%ICVCNT 100
%ICVOLS 100
%LOGDSN 98
%LOGETIM 98
%LOGFRID 99
%LOGFSEQ 98
%LOGGLRID 99
%LOGMERG 98
%LOGONL 98
%LOGRMET 98
%LOGSEL 98
%LOGSTIM 98
%LOGSTIM 98
%LOGUNIT 98
%LOGVOLS 98
%LOGFRID 95
%OLDDCTIM 95
%OLDDRFRD 95
%OLDDRNRD 95
%OLDDTLM 95
%OLDSDSN 95
symbolic keywords (continued)
JCL execution member (continued)
%OLDSSSN 95
%OLDSSEL 95
%OLDSSTY 95
%PLGTIME 102
%SLDECRE 96
%SLDFRID 96
%SLDFSEQ 96
%SLDLRDRD 96
%SLDNRMT 96
%SLDSDDN 96
%SLDSSEL 96
%SLDSTIM 96
%SLUDUNIT 96
%SLUDVOLS 96
%SSID 87
%TIME 87
symbolic keywords 86, 102
recognized by DBRC 431
user-defined 87
symbolic keywords, JCL execution member
substitution 83
symbolic keywords
%SLDCHKT 96
syntax diagram
how to read 122
syntax, DBRC command 120
system log data set
See SLDS (system log data set)
system log data set (SLDS)
records 65
system log management, with data sharing 47
system logs, with data sharing 47

T
TAPEUNIT parameter commands
CHANGE:RECON 184
INIT:RECON 284
tasks of DBRC 4
time qualifier 90
time stamp 124
conversions and examples 128
DBRC commands affected by format 129
format for DBRC requests 354
recovery 11
specifying zero values 128
standard default settings for values 129
standard format 124
TIMEFMT parameter 125
two-digit year input 128
TIMEFMT parameter
GENJCL:ARCHIVE 219
GENJCL:CA 223
GENJCL:CLOSE 227
GENJCL:IC 234
GENJCL:OIC 241
GENJCL:RECEIVE 245
GENJCL:RECOV 250
GENJCL:USER 255
TIMEFMT parameter sublist
CHANGE:RECON 185

584  DBRC Guide and Reference
Program Number: 5655-J38

Printed in USA