IMS

Database Recovery Control (DBRC) Guide and Reference

Version 9

IMS

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 557.

Quality Partnership Program (QPP) Edition (June 2004) (Softcopy Only)

This QPP edition replaces or makes obsolete the previous edition, ZES1-2342-01. This edition is available in softcopy format only. The technical changes for this version are summarized under “Summary of Changes” on page xiii.

Contents

Figures

Tables

About This Book

This information is available in PDF and BookManager formats, and also as part of the IMS Version 9 QPP Information Center. 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. To get the most current versions of these books for the information center, go to the IMS V9 Vendor and Quality Partnership Program Library page at www6.software.ibm.com/dl/ims02/imsv9lib-p, where you can find updated plug-ins and instructions on how to install them in your IMS Version 9 QPP Information Center.

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.

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 121 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 Version 9: Master Index and Glossary*.
- For more information on upgrading the RECON data set from previous releases of IMS, see the *IMS Version 9: Release Planning Guide*.
- For installation and initialization topics, see the:

  - *IMS Version 9: Installation Volume 1: Installation Verification*
  - *IMS Version 9: Installation Volume 2: System Definition and Tailoring*
- For information on database and system administration, see the:
  - *IMS Version 9: Administration Guide: Database Manager*
  - *IMS Version 9: Administration Guide: System*
- For information on operating procedures, see the *IMS Version 9: Operations Guide*.
- For information on system utilities, see the *IMS Version 9: Utilities Reference: System*.
- For information on the utilities used in database recovery, see the *IMS Version 9: Utilities Reference: Database and Transaction Manager*.
- For information on the RECON I/O exit routine, see the *IMS Version 9: Customization Guide*.
- For diagnostic information and messages, see the:
  - *IMS Version 9: Diagnosis Guide and Reference*
  - *IMS Version 9: Messages and Codes, Volume 1* and *IMS Version 9: Messages and Codes, Volume 2*

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 do one of the following:

- 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 you are commenting on (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

New information on the following is included:

- Sysgen Conditional Bind Removal (see “Initializing DBRC” on page 20)
- Recommendations about “RECON Data Set Serialization Strategies” on page 61
- Chapter 1 (DBRC in the IMS Database Recovery Process) was completely rewritten and renamed to Chapter 1, “Overview of DBRC,” on page 5.
- Some reorganization has occurred:
  - Some sections from Chapter 3, “Considerations for a DBRC System,” on page 35 were moved and merged into Chapter 1, “Overview of DBRC,” on page 5.
  - Chapter 2, “Database Recovery Control Utility (DSPURX00),” on page 29 was moved to from being 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 29.
- A terminology change has occurred. All instances of “*simple* keyword” have been changed to “*symbolic* keyword”.

Changes to This Book for IMS Version 9

This edition is a draft version of this book intended for use during the Quality Partnership Program (QPP). Contents of this book are preliminary and under development.

New information on the following enhancements is included:

- Command Authorization for /RM Commands (see “DBRC Online Commands” on page 126)
- Eliminate the DSPSLDRC Part from DBRC (PRISLD / SECSLD / PRITSLDS / SECTSLDS now mapped by DSPLOGRC, see the RECON data set record DSECTs listed in Appendix B, “Sample Listings of RECONs,” on page 469)
- Fast Path Multiple Areas Shared VSO Structures (see “CHANGE.DBDS” on page 155, “INIT.DBDS” on page 266, and “DB (FP) and Related Records” on page 499)
- HALDB Online Reorganization. See:
  - “DBRC Considerations for HALDB Online Reorganization” on page 51
  - “CHANGE.DB” on page 147
  - “CHANGE.RECON” on page 181
  - “GENJCL.IC” on page 231
  - “GENJCL.OIC” on page 240
  - “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 473
- DBRC Application Programming Interface (API). See:
  - “DBRC Tasks” on page 6
  - “Communicating with DBRC” on page 7

| – Chapter 16, "Using the DBRC API," on page 349
| – Chapter 17, "DBRC Query Request," on page 357
| – Chapter 18, "DBRC Release Buffer Request," on page 417
| – Chapter 19, "DBRC Start Request," on page 421
| – Chapter 20, "DBRC Stop Request," on page 425

Library Changes for IMS Version 9

Changes to the IMS Library for IMS Version 9 include the addition of new titles, the change of one title, and a major terminology change. Changes are indicated by a vertical bar (|) to the left of the changed text.

New and Revised Titles

The following list details the major changes to the IMS Version 9 library:

- *IMS Version 9: HALDB Online Reorganization Guide*

  The library includes new information: *IMS Version 9: HALDB Online Reorganization Guide*. This information is available only in PDF and BookManager formats.
- *IMS Version 9: An Introduction to IMS*

  The library includes new information: *IMS Version 9: An Introduction to IMS*.
- The information formerly titled *IMS Version 8: IMS Java User's Guide* is now titled *IMS Version 9: IMS Java Guide and Reference*.
- The library includes new information: *IMS Version 9: IMS Connect Guide and Reference*. This information is available only in PDF and BookManager formats.

Terminology Changes

IMS Version 9 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 and can have a broader scope than type-1 commands. In IMS Version 8, these commands were called *IMSplex* commands or *enhanced* commands.

Accessibility Enhancements

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products. The major accessibility features in z/OS products, including IMS, enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

User Assistive Technologies

Assistive technology products, such as screen readers, function with the IMS user interfaces. Consult the documentation of the assistive technology products for specific information when you use assistive technology to access these interfaces.

Accessible Information

Online information for IMS Version 9 is available in BookManager format, which is an accessible format. All BookManager functions can be accessed by using a keyboard or keyboard shortcut keys. BookManager also allows you to use screen readers and other assistive technologies. The BookManager READ/MVS product is included with the z/OS base product, and the BookManager Softcopy Reader (for workstations) is available on the IMS Licensed Product Kit (CD), which you can download from the Web at www.ibm.com.

Keyboard Navigation of the User Interface

Users can access IMS user interfaces using TSO/E or ISPF. Refer to the *z/OS V1R1.0 TSO/E Primer*, the *z/OS V1R1.0 TSO/E User's Guide*, and the *z/OS V1R1.0 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.

Part 1. How to Use DBRC

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 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 9 for additional information about DBRC’s logging support.
- See “Supporting Data Sharing” on page 18 for additional information about DBRC’s data sharing support.
- See Part 2, “DBRC Command Reference,” on page 121 for detailed information about the DBRC commands.
- See Part 3, “DBRC Application Programming Interface (API) Reference,” on page 347 for detailed information about the DBRC API requests.

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 7

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 57.

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:

- 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 29.

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 87.

When Should You Use DBRC?

Most IMS configurations require DBRC, including:

- Online configurations: DB/DC, DCCTL, or DBCTL
- Data-sharing environments, including IMplexes
- 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 Version 9: Operations Guide* provides detailed descriptions of recovery procedures with and without DBRC.
- "How DBRC Helps in Recovery" on page 14 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" on page 8 and "DBRC API Requests" on page 8 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 125.

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” on page 9
- “Recovering Databases” on page 12
- “Recording Information About Opening and Updating Databases” on page 17
- “Supporting Data Sharing” on page 18
- “Supporting Remote Site Recovery” on page 19
- “Supporting IMSplexes” on page 20

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.

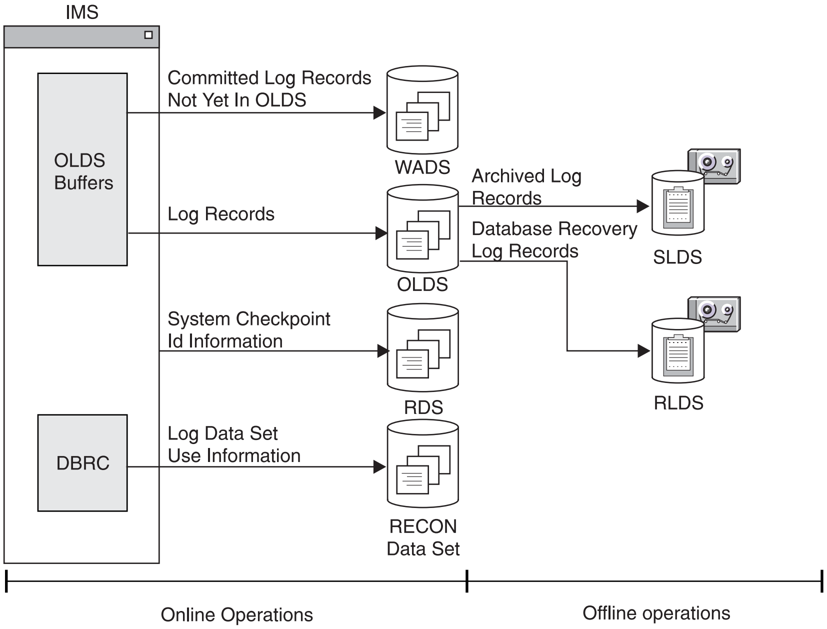

IMS
OLDS Buffers
DBRC
Committed Log Records Not Yet In OLDS
Log Records
System Checkpoint Id Information
Log Data Set Use Information
WADS
OLDS
RDS
RECON Data Set
Archived Log Records
Database Recovery Log Records
SLDS
RLDS
Online Operations
Offline operations

Figure 1. Logs Produced for Recovery and Restart

Related Reading: See the *IMS Version 9: Operations Guide* and *IMS Version 9: An Introduction to IMS* 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 Version 9: 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 Version 9: Operations Guide* for more information about automatic, manual and custom archiving of log records.
- See *IMS Version 9: Utilities Reference: System* for more information about specifying entry points and running the Log Archive utility.
- Refer to *IMS Version 9: 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`
- `DELETE.LOG`
- `GENJCL.ARCHIVE`
- `GENJCL.CLOSE`
- `LIST.LOG`
- `NOTIFY.PRILOG`
- `NOTIFY.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 121 for descriptions of the commands and "Log Query" on page 393 for the details of the Log and OLDS query requests.

Recovering Databases

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 45.
3. Back out any incomplete changes.

Figure 2 illustrates a simple database recovery.

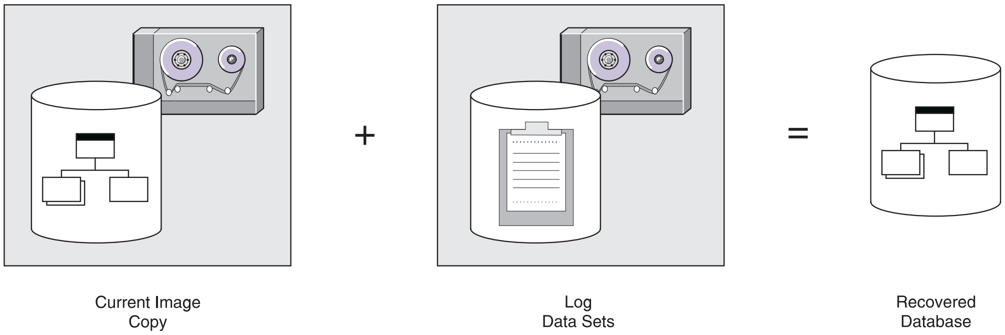


Figure 2. IMS Database Recovery process

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 Version 9: 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 14 shows how DBRC works with the Database Recovery utility.

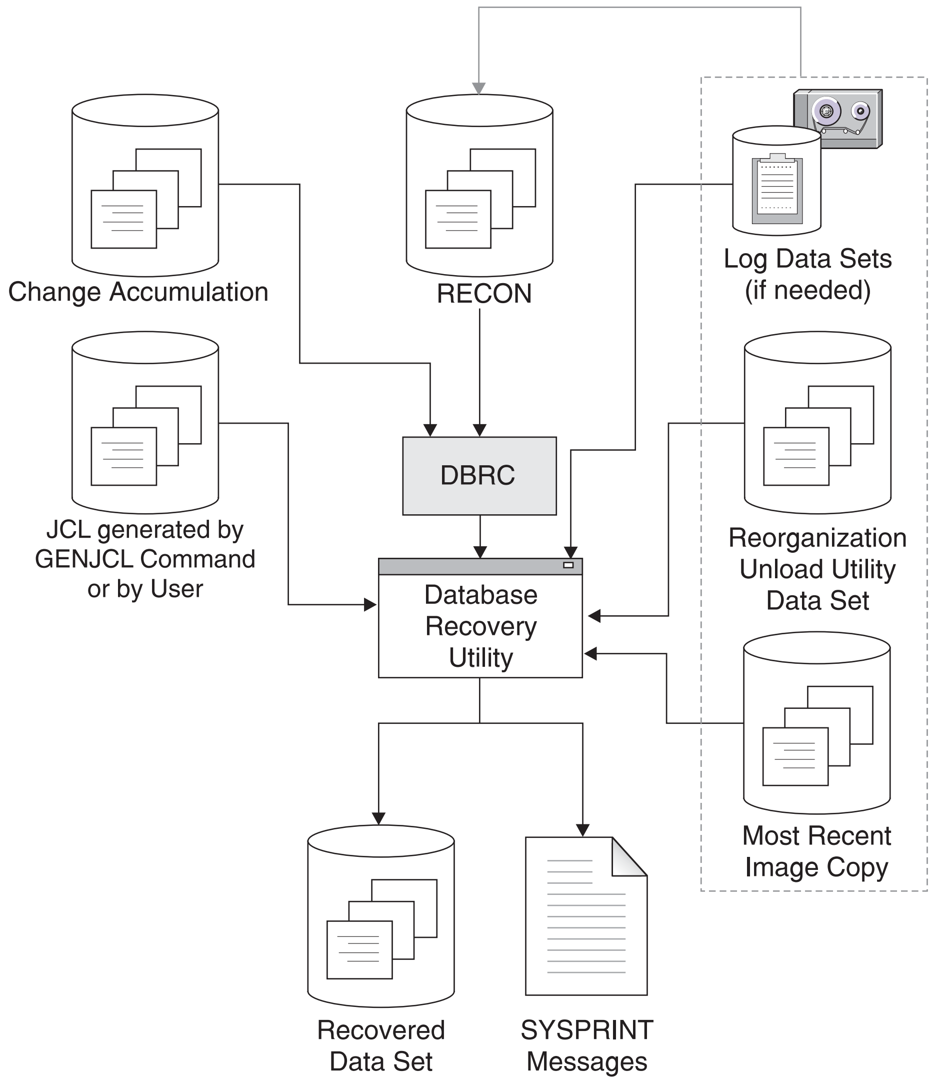

Change Accumulation
RECON
Log Data Sets (if needed)
JCL generated by GENJCL Command or by User
DBRC
Database Recovery Utility
Reorganization Unload Utility Data Set
Most Recent Image Copy
Recovered Data Set
SYSPRINT Messages

Figure 3. 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 15)
- Validate the input to those utilities (see “Validating Utility JCL” on page 16)
- Record the result (in the RECON data set) of running the utilities (see “Recording the Result” on page 17)

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.

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 6 on page 87. There is also a skeletal JCL execution member, JOBJCL, that produces a JOB statement.

In addition, the GENJCL.USER command generates 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 Version 9: Installation Volume 2: System Definition and Tailoring* for more information about the tailoring actions for IMS.PROCLIB members, the DBRC procedure, and the JCLOUT and JCLPDS DD statements.
- See Chapter 5, "Skeletal JCL," on page 87 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 GENJCL 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 (DFSUARC0)
- 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 17 shows DBRC's role in running the previously mentioned utilities.

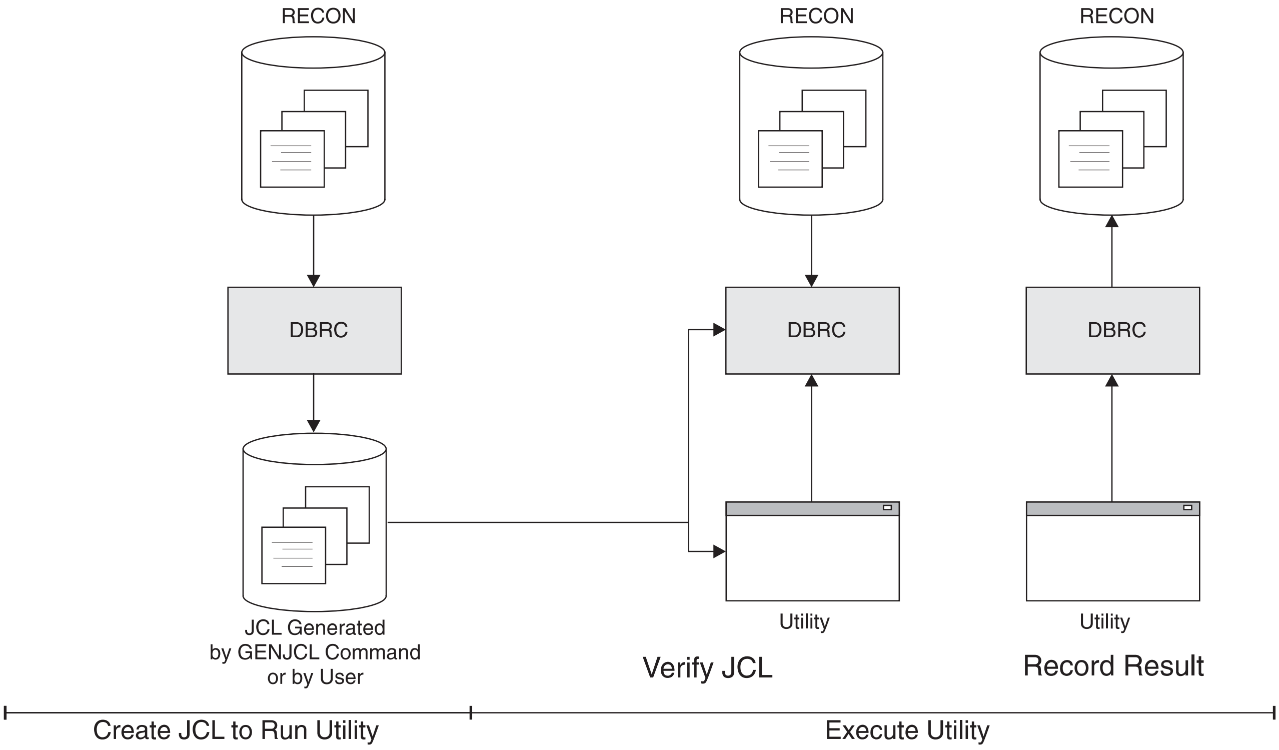


Figure 4. DBRC's Role in Utility Execution

Related Reading:

- See *IMS Version 9: Utilities Reference: Database and Transaction Manager* for more information on the IMS recovery utilities.
- See "NOTIFY.BKOUT" on page 307 for information about a related command, which manually creates a BACKOUT record in the RECON data set.

When you run the Batch Backout utility (DFSBBO00), 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 Version 9: 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.

Related Reading: See the *IMS Version 9: Operations Guide* and the *IMS Version 9: Administration Guide: System* for more information on data sharing.

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.

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.

  The RECON data set contains the definition of an RSR complex.

  Related Reading: See *IMS Version 9: Operations Guide* and *IMS Version 9: Administration Guide: System*, ZES1-2331 for more information about RSR.
- Services that are used by an active subsystem to identify the tracking subsystem and the databases covered by RSR.

  An active subsystem obtains the identity of its tracking subsystem from DBRC. As databases are updated by the active subsystem, DBRC tells the database component whether the database is covered by RSR. If the databases are being tracked by RSR, the active subsystem sends its log data to the tracking subsystem.
- Services used by a tracking subsystem to record information about log data that is received from an active subsystem.

  As logs are received and stored at the tracking site, DBRC records the receipt of the log data. When begin-update records are received for registered databases, DBRC records the database update.

- 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 Version 9: Administration Guide: System* for more information on controlling database recovery in an RSR environment.

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 Version 9: Common Service Layer Guide and Reference* for more information about IMSplexes.

Initializing DBRC

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:

- Refer to *IMS Version 9: Installation Volume 1: Installation Verification* and *IMS Version 9: Installation Volume 2: System Definition and Tailoring* for a complete description of IMS installation procedures and requirements.
- See Chapter 4, “Initializing and Maintaining the RECON Data Set,” on page 57 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 ADFSSMPL library contains a batch and utility region default of DBRC=YES. This value is coded on the DFSIDEF macro.

Related Reading: For more information about the DFSIDEF0 module, see the *IMS Version 9: Installation Volume 2: System Definition and Tailoring*.

The IMSCTRL Macro

As of IMS Version 9, 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 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 Version 9: 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 57 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.DBDS` 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 `INIT.DBDS` command to register all its data sets or DEDB areas. For DEDBs, use the `INIT.ADS` command to identify the data sets within each area. An area can have up to seven area data sets (ADSs).

Requirement: Utility jobs that issue either an `INIT.DB` or `INIT.PART` command (to register a HALDB) or any `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 `INIT.DB` command.

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`.

Related Reading:

- See the *IMS Version 9: Administration Guide: Database Manager* for an overview of HALDBs and detailed information about how to create them.
- See the *IMS Version 9: Utilities Reference: Database and Transaction Manager* for information about the HALDB Partition Definition utility.
- See Chapter 9, “CHANGE Commands,” on page 139 for information about the CHANGE commands.

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"
- "Recovery without DBRC" on page 26
- "Restart After an IMS Failure" on page 26
- "Restart After a DBRC Failure" on page 26
- "Recovery Involving IRLM Configurations" on page 26
- "Batch Backout" on page 26

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 Version 9: 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" on page 24
- "Batch Backout Support" on page 24
- "Forward Recovery" on page 25

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 Version 9: 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 Version 9: Operations Guide* and *IMS Version 9: Utilities Reference: Database and Transaction Manager* 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.
- 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 Version 9: 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 Version 9: 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 Version 9: 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 of RECONs,” on page 469 and *IMS Version 9: 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)?

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 "CHANGE.RECON" on page 181 and the *IMS Version 9: Release Planning Guide*, GC17-7831 for more information on RECON data set conversion.
- See Chapter 7, "DBRC Commands," on page 125 for information about command syntax.

Input and Output for the Database Recovery Control Utility

Figure 5 on page 30 shows the input and output requirements for the Database Recovery Control utility. Notes pertaining to the figure follow it.

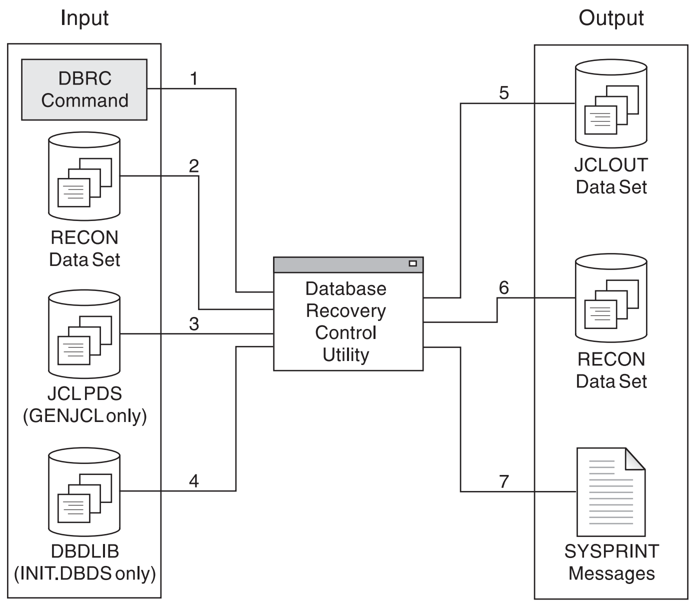


Figure 5. Database Recovery Control Utility Input and Output

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 to be placed under the control of DBRC (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 32 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.

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 32 for an illustration of the necessary registers and operand list contents.

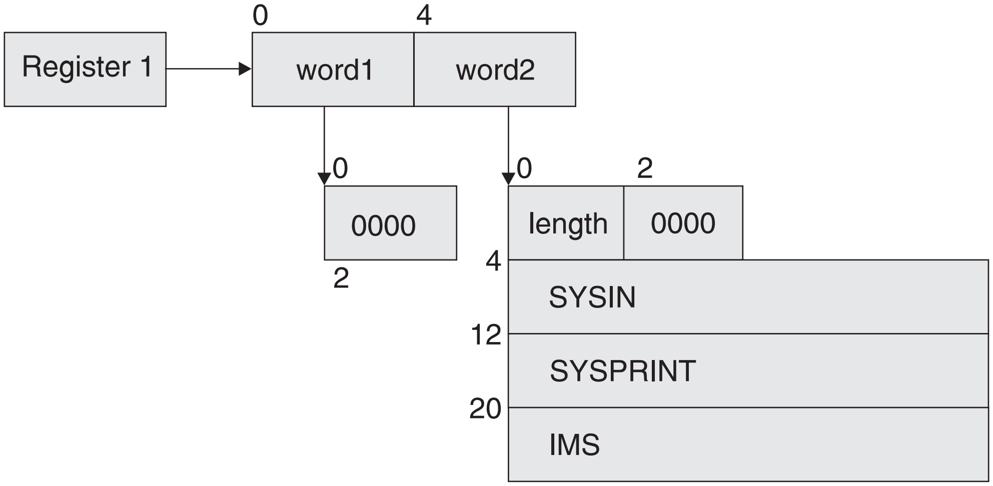


Figure 6. Registers and Operand List Contents Needed To Invoke the Database Recovery Control Utility

The second word is the address of the ddname list (see Figure 6 on page 32). 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 33 is a sample job that initializes the RECON data set and registers one database with two DBDSs. Descriptions of the individual numbered JCL statements are listed under Figure 7 on page 33.

```
   //INITRCON JOB
 1 //INIT04   EXEC PGM=DSPURX00
 2 //STEPLIB  DD   DSN=IMS.SDFSRESL
 3 //SYSPRINT DD   SYSOUT=A
 4 //*
 5 //IMS      DD   DSN=IMS.DBDLIB
 6 //JCLPDS   DD   DSN=IMS.JCLPDS
 7 //JCLOUT   DD   DSN=IMS.JCLOUT
 8 //SYSIN    DD   *
 9 INIT.RECON SSID(IMS3)
10 INIT.DB    DBD(DBDESDS1)  SHARELVL(2)
11 INIT.DBDS  DBD(DBDESDS1)  DDN(DDNESDSA)  GENMAX(3)  -
     REUSE  DSN(IMS.DBDESDS1.DDNESDSA.DSN)    -
     ICJCL(MYIC) RECOVJCL(MYRECOV)          -
12 INIT.IC    DBD(DBDESDS1)  DDN(DDNESDSA) -
     ICDSN(IMS.*.ICDSN1)
   INIT.IC    DBD(DBDESDS1)  DDN(DDNESDSA) -
     ICDSN(IMS.*.ICDSN2)  ICDSN2(IMS.*.ICDSN2)
   INIT.IC    DBD(DBDESDS1)  DDN(DDNESDSA) -
     ICDSN(IMS.*.ICDSN3)
   //*
13 INIT.DBDS  DBD(DBDESDS1)  DDN(DDNESDSB)  GENMAX(4)  -
     NOREUSE  DSN(IMS.DBDESDS1.DDNESDSB.DSN)
   //*
14 INIT.CAGRP  GRPNAME(CAGRP1)  GRPMAX(2) REUSE -
     GRPMEM((DBDESDS1,DDNESDSA),(DBDESDS1,DDNESDSB))
15 INIT.CA  GRPNAME(CAGRP1) CADSN(IMS.*.CADSN1) -
     VOLLIST(CAVOL1,CAVOL2,CAVOL3)  FILESEQ(4)
   INIT.CA  GRPNAME(CAGRP1) CADSN(IMS.*.CADSN2) -
     VOLLIST(CAVOL4)
   /*
```

Figure 7. Inputs and Outputs of the Database Recovery Control Utility

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`).

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 35 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 35.

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:

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-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.

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.IC` or `GENJCL.OIC` 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 45 for a discussion of how to notify DBRC about these data sets.

Related Reading: See *IMS Version 9: 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 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 40 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.

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.OIC 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/STOP AREA command, for example, to prevent updating of the database or area by transactions in the system previously doing updates.

Database Image Copy 2 (DFSUDMT0)

The Database Image Copy 2 utility (DFSUDMT0) 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.

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 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.

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 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 image copy information is maintained in the RECON data set. For example, if the recovery period of a DBDS or DEDB area is 14 days, image copies are maintained 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 1 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 1. Results of GENJCL.IC Processing When GENMAX and RECOVPD are Specified with REUSE

Number of Image Data Sets	Number of In-Use Image Copies	Age of Oldest Image Copy	GENJCL Result
=GENMAX	=GENMAX	<RECOVPD	Fail (DSP0063I)
>GENMAX	=GENMAX	<RECOVPD	Avail IC DS used (DSP0065I)
=GENMAX	=GENMAX	>RECOVPD	Oldest IC DS reused
>GENMAX	=GENMAX	>RECOVPD	AVAIL IC DS used

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

Number of Image Copies	Age of Oldest Image Copy	Utility EOJ Result
=GENMAX	>RECOVPD	Delete oldest image copy

Table 2. Results of GENJCL.IC Processing When GENMAX and RECOVPD are Specified with NOREUSE (continued)

Number of Image Copies	Age of Oldest Image Copy	Utility EOJ Result
=GENMAX	<RECOVPD	No delete (DSP0064I)
<GENMAX	N/A	No delete

If you issue a CHANGE.DBDS 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.IC 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.DBDS 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.DBDS 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.IC 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.IC 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 USEDBDS or USEAREA for the `GENJCL.RECOV` command or DFSDUMP 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:

- 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 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.

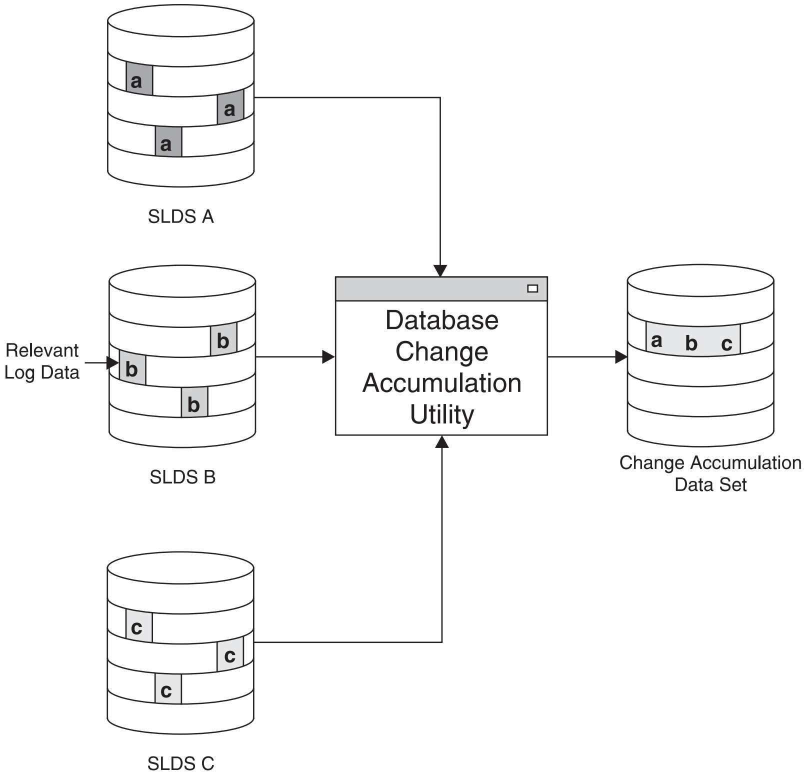

a
a
a
SLDS A
Relevant Log Data
b
b
b
SLDS B
Database Change Accumulation Utility
a b c
Change Accumulation Data Set
c
c
c
SLDS C

Figure 8. What Change Accumulation Does with Redo Data from Divergent Data Streams

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 Version 9: 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 RLDSs, 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, 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 Version 9: Utilities Reference: System* for more information on the Database Change Accumulation utility.
- See the *IMS Version 9: Operations Guide* for more information on the change accumulation process in general.

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 that 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.
- 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 431 and "Database Change Accumulation Utility JCL (CAJCL)" on page 445 for details on the Change Accumulation JCL specifications.

DBDS Groups

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.

Related Reading: See Chapter 11, “GENJCL Commands,” on page 221 for the impact of these data sets on the GENJCL commands.

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 Version 9: 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 sections describe the DBRC considerations with respect to HALDB OLR.

The following DBRC commands support HALDB OLR:

- CHANGE.DB
- GENJCL.IC
- GENJCL.OIC
- INIT.DB
- NOTIFY.IC
- NOTIFY.REORG
- NOTIFY.UIC

The records in the RECON data set that contain HALDB OLR information are the:

- HALDB Master record
- Partition DB record
- DBDS record
- REORG record

GENJCL.IC and GENJCL.OIC Considerations

The GENJCL.IC and GENJCL.UIC JCL can be generated in advance and not executed immediately. So, the active set of DBDSs may have changed 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 utilities. The SYSIN control statement identifies which DBDS to copy. This can be either the A-through-J or 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 Considerations

An image copy cannot be taken while HALDB OLR cursor is active. The NOTIFY.IC and NOTIFY.UIC commands fail if they attempt to record an image copy while HALDB OLR cursor is active.

DBRC RECON HALDB OLR Information

The DB record for a HALDB partition contains information for HALDB OLR:

- A field that indicates whether the database is capable of being reorganized online.
- A field that indicates whether HALDB OLR cursor is active.
- The SSID of the IMS currently performing a HALDB OLR.
- A field indicating which set or sets of DBDSs is or are active for the partition.
- Counters showing the number of cursor-active status changes and cursor-inactive status changes that are not yet hardened at the RSR tracking site.

The HALDB Master and Partition DB records contain a new field indicating if the HALDB can be reorganized online.

The M-through-V and Y DBDS records are defined in the RECON data set with the exact same attributes as the corresponding A-through-J and X DBDS records. One difference, in addition to DD and data set names, is the DSID. The DSID of the M-through-V and Y data set is the same as the corresponding A-through-J and X data set with its X'80' bit on. For the RECON data set list purposes, the X'80' bit is ignored.

Table 3. DSID DCB Numbers for the A-J and X and the M-V and Y data sets

DSID / DCB Number	A-J, L, and X Data Sets		M-V and Y Data Sets	
	PHIDAM	**PHDAM**	**PHIDAM**	**PHDAM**
1	A	A	M	M
2				
3	L	L		
4		B		N
5	X	C	Y	O
6	B	D	N	P
7	C	E	O	Q
8	D	F	P	R
9	E	G	Q	S
10	F	H	R	T
11	G	I	S	U
12	H	J	T	V
13	I		U	
14	J		V	

The M-through-V and Y DBDS records are added to the indicated change accumulation group. That is, to the same CAGROUP as the A-through-J and X DBDS.

The M-through-V and Y DBDS records are added to each DBDS group where the A-through-J DBDS records are found.

The REORG record for a HALDB partition DBDS contains new information (stop time) for HALDB OLR. In order to run an offline reorganization of it, a database must have been taken offline successfully with a /DBR command. So, only the start time of the reorganization is necessary. With HALDB OLR, the database is online so the duration of the process is important. The stop time of the reorganization is now recorded in the REORG record along with the new indicator showing that this is an HALDB OLR and whether the HALDB OLR may be used as input to recovery.

A REORG record is recorded under both the A-through-J 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 HALDB OLR has completed.

The HALDB OLR capability is set at the HALDB Master DB level.

The DB Record header record contains a new field which indicates whether HALDB OLR is enabled for this partition. Once enabled, sharing of HALDBs with an IMS Version 7, or IMS Version 8 without the OLR Coexistence SPE, 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 marked as not capable of being reorganized online. This field is set or reset with the `CHANGE.DB DBD(HALDB master)` command and is displayed in the `LIST.DB` output.

Table 4 on page 53 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

The ability to share data is also dependent on whether OLRCAP or OLRNOCAP is specified for a database and this table also depicts those cases.

Table 4. Data Sharing Between the Releases

	Version 7	**Version 8 *without* SPE Applied**	**Version 8 *with* SPE Applied**	**Version 9**
OLRCAP	NO	NO	YES	YES
OLRNOCAP	YES	YES	YES	YES

Resetting the field in the DB record header to disable HALDB OLR fails if HALDB OLR cursor is active, or if any of the M-through-V and Y data sets are active. Resetting the field in the DB record header to disable HALDB OLR also fails if any partition for the HALDB is authorized. If resetting the field is permitted, all of the M-through-V and Y data set information in the RECON data set is deleted.

RSR Tracking System

Ownership of the HALDB OLR is not recorded in the tracking RECON data set so the tracking the RECON data set do not reflect ownership of HALDB OLRs.

The RECON data set at the tracking site always reflect the current status of the covered databases. New counters keep track of outstanding HALDB OLRs that have been recorded in the RECON data set at the tracking site but not yet applied to the databases.

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: 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-to-M, B-to-N, C-to-O, and so on. Either set of data sets (A-J and X or M-V and Y) can be reorganized online at any time depending on which set is active prior to the start of the HALDB OLR. So data sets A and M are one logical data set during a

HALDB OLR. After an image copy is taken, DBRC removes any extraneous recovery related records (IMAGE, ALLOC, RECOV, and REORG) from the RECON data set. What is extraneous is determined by the GENMAX and the optional RECOVPD settings for the DBDS. GENMAX and RECOVPD apply to the A-through-J or X data sets and the corresponding M-through-V or Y data sets as if the two data sets were one logical data set.

For example, GENMAX=2 for data set A (and M) and data set A has two image copies. When data set M becomes the active data set and an image copy is taken, image copy cleanup results in deleting the oldest image copy for data set A along with any of its extraneous recovery records. Another image copy of data set M results in the rest of data set A's image copy and recovery records being deleted. Note that data set A would become unrecoverable in this case.

The HALDB OLR Execution Process

When HALDB OLR is started, the ID of the IMS 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 outcomes also occur:

- The DB record for the partition indicates 'OLREORG CURSOR ACTIVE =YES'.
- REORG records are written showing the start of the reorganization (the stop time is zero) indicating online.
- The REORG record under the data set that is the output indicates that it can be used as input for recovery.
- The REORG record under the data set that was the input does not indicate that it can be used for recovery. DBRC uses this record to ensure that recovery of this data set cannot occur after HALDB OLR is complete.
- 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 and X DBDSs, if one is assigned.
- The M-through-V and Y DBDSs are added to any DBDS group where the A-through-J and X DBDSs are found.

At this point the HALDB OLR must either complete or an offline reorganization must be run in order to reset the HALDB OLR settings.

HALDB OLR cannot be started if it is already running for this partition. However, another IMS can take over the reorganization if it is terminated; that is, when there is no IMS ID assigned.

When the entire HALDB OLR process completes successfully the following outcomes occur:

- The REORG records are updated to include the HALDB OLR stop time.
- The DB record for the partition is changed to indicate 'OLREORG CURSOR ACTIVE = NO'.
- The status of the active DBDSs is switched to the newly reorganized data sets.
- The HALDB OLR IMS ID is cleared.

Recommendation: You are not required to create image copies of a partition's DBDSs after a successful completion of a HALDB OLR. 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 HALDB OLR cursor is active. DBRC causes image copy to fail in this case.

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.

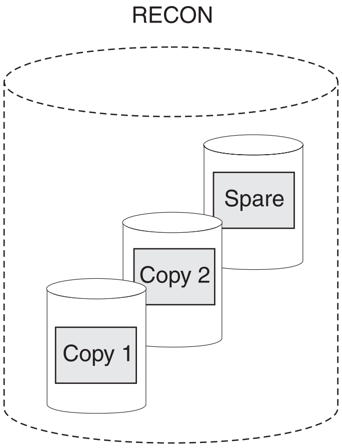


Figure 9. DBRC Dual RECON Set Up with a Spare

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

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

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 65).

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 Version 9 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.
- Specify the maximum record size for all RECON data sets when defining the RECON data sets using AMS.
- 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 65

Recommendations:

- 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.
- Use GRS STAR for converting the hardware reserves for the RECON data sets to enqueues. For more information, see “RECON Data Set Serialization Strategies” on page 61.

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 enqueueing 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 *MVS/ESA 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.

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:

- *OS/390 MVS Programming: Authorized Assembler Services Reference, GC28-1763*
- *OS/390® 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 MDFSMSdfp 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 GRSCNFxx parmlib member or the SETGRS operator command. The GRSDEF statement of GRSCNFxx contains the SYNCHRES (*Y* | *N*) parameter. The default value for SYNCHRES is *NOT*. 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: The following items discuss several serialization implementations and their effects:

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. This locks out access to the RECON data set by IMS code that does the enqueue on the RECON data set.

Recommendation: Implement DBRC using GRS RNL CONVERSION. All DBRC instances should be in the same GRSPLEX and use GRS STAR.

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.

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. Add the VSAM volume data QNAME to the RESERVE CONVERSION RNL. For example, `RNLDEF RNL(CON) TYPE(GENERIC) QNAME(SYSZVVDS)`.
3. Add VTOC QNAME to the RESERVE CONVERSION RNL. For example, `RNLDEF RNL(CON) TYPE(GENERIC) QNAME(SYSVTOC)`.
4. Add user catalog QNAME to the RESERVE CONVERSION RNL. For example, `RNLDEF RNL(CON) TYPE(GENERIC) QNAME(SYSIGGV2)\`.
5. Add system data sets QNAME to the SYSTEM INCLUSION RNL. For example, `RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(SYSDSN)`.

The following paragraphs discuss additional considerations for implementing GRS RNL CONVERSION.

The reserves for SYSIGGV2 are issued in three different ways. If the catalog name is less than 21 bytes in length, then the catalog issues a reserve for QNAME SYSIGGV2 RNAME of the catalog name and specifies a length 20 for the RNAME. If the catalog name is 21 bytes or large, the reserve is issued with a length of 44 bytes. SYSIGGV2 is also issued with a 88 byte length, where the rname is 44 bytes of the data set name and 44 bytes for the catalog name. If the catalogs involved are shared within the same GRSplex, there should be an entry in the GRSRNLxx member as follows:

```
RNLDEF RNL (CON) TYPE (GENERIC) QNAME (SYSIGGV2)
```

If there are catalogs that are shared with systems outside of the GRSplex, you need to code entries to ensure the integrity of your catalogs.

- If the catalog name is 20 bytes or less, code an entry for each catalog as follows:

  ```
  RNLDEF RNL(EXCL) TYPE(SPECIFIC) QNAME(SYSIGGV2)
   RNAME('your.short.name.cat')
  ```
- If the catalog name is 21 bytes or longer, code an entry as follows for each catalog:

  ```
  RNLDEF RNL(EXCL) TYPE(SPECIFIC) QNAME(SYSIGGV2)
   RNAME('your.longer.than.twenty.byte.catalog.name')
  ```

Requirement: If you are coding either of the above entries, the length must be either 20 or 44 bytes. If the length is not either 20 or 44 bytes, you will experience deadlocks. Also, these entries must be coded as SPECIFIC NOT GENERIC. This is due to the logic that GRS uses in determining if there is a match.

If you are sharing catalogs outside of GRSplex you may also need to code an RNL as follows:

```
RNLDEF RNL(EXCL) TYPE(PATTERN) QNAME(SYSIGGV2)
 RNAME(????????????????????????????????????????????outside.sysplex.ucat)
```

The RNAME in the previous example needs to specify a length of 88 bytes with the first 44 bytes being question marks and the last 44 bytes being the catalog name that is shared outside of the GRSplex.

For SYSZVVDS, the recommended method for it to be excluded is as follows:

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

However, in GRS STAR, you can convert this as follows:

```
RNLDEF RNL (CON) TYPE (GENERIC) QNAME (SYSZVVDS)
```

The major issue with converting or excluding SYSZVVDS is that it must be treated the same manner as the resource SYSVTOC. If you convert one, you must convert the other. Likewis,e if you exclude one, you must exclude the other. If you convert one, but not the other, you will experience deadlocks.

If you want to have the catalogs that contain the RECON data sets serialized using reserves rather enqueues, you need the following entries:

- With the catalog name of SYS1.RCN1.URP.CATALOG, code an RNL as follows:

  ```
  RNLDEF EXCL QNAME(SYSIGGV2)
  RNAME('SYS1.RCN1.URP.CATALOG ')
  /* 12345678901234567890123456789012345678901234 */
  ```
- To exclude the data sets cataloged in that catalog, code the following:

  ```
  RNLDEF RNL(EXCL) TYPE(PATTERN) QNAME(SYSIGGV2) RNAME
  ('????????????????????????????????????????????SYS1.RCN1.URP.CATALOG')
  /*123456789012345678901234567890123456789012345678901234567890123456789
  0123456789012345678 <= Total length = 88 */
  ```

```
/*The above shows how to code a RNAME that continues over 1 line */
/* 12345678901234567890123456789012345678901234
   <= Catalog name must be 44 bytes */
```

- You will also need the following two RNLs:

  ```
  RNLDEF RNL (EXCL) TYPE (GENERIC) QNAME (SYSZVVDS) RNAME(volser)

  RNLDEF RNL (EXCL) TYPE (GENERIC) QNAME (SYSVTOC) RNAME(volser)
  ```

  In the above RNLs, *volser* in the RNAME is the volume serial number the catalog resides on

You will need the above group of 4 RNLs for each catalog that you want to serialize with reserves. The first entry for SYSIGGV2 must be specific. This is due to how GRS processes the RNLs. The first RNL scanned when a reserve, enqueue, or dequeue is issued with a scope of SYSTEMS (all catalog requests are issued with a scope of SYSTEMS for catalogs with shareoptions 3,4 and residing on a volume defined as shared) is the exclusion RNL. Note that when GRS scans specific entries, it is checking for an exact match on the RNAME and the length of the RNAME. When GRS does its comparison for specific entries, it is a string compare; a difference in the length does not create a match. When GRS does a comparison for generic entries it is also a string comparison, but it is only looking for match of the length of RNAME.

Enhanced Catalog Sharing (ECS) will reduce the number of I/O operations to the VVDS on volumes that contain catalogs. Each time a shared catalog (shareoptions 3,4 and residing device defined as shared) is accessed, a check is needed to see if the catalog has been updated from another system. This requires a read of the VVDS for the volume the catalog resides on. With ECS, this data is stored in a coupling facility and therefore DBRC performs fewer I/Os to the VVDS. I/O to the VVDS is not eliminated, but reduced. Note that ECS is limited to a single sysplex.

To summarize, for catalogs that are shared within a GRSplex, you should have the following RNLs:

```
RNLDEF RNL CON QNAME(SYSIGGV2) TYPE (GENERIC)
RNLDEF RNL EXCL QNAME(SYSZVVDS) TYPE (GENERIC)
RNLDEF RNL EXCL QNAME(SYSVTOC) TYPE (GENERIC)
```

If you are using GRS STAR, you might want to substitute the following RNLs for catalogs shared within a GRSplex:

```
RNLDEF RNL CON QNAME(SYSIGGV2) TYPE (GENERIC)
RNLDEF RNL CON QNAME(SYSZVVDS) TYPE (GENERIC)
RNLDEF RNL CON QNAME(SYSVTOC) TYPE (GENERIC)
```

For those catalogs that are shared outside of the GRSplex or that you want to use reserve to serialize access, you will need the following entries for each catalog and for the volume serial numbers that the catalogs reside on:

```
RNLDEF RNL(EXCL) TYPE(SPECIFIC) QNAME(SYSIGGV2)
RNAME('your.short.name.cat ') /* if catalog name is 20 bytes or less*/
RNLDEF RNL(EXCL) TYPE(SPECIFIC) QNAME(SYSIGGV2)
RNAME('your.longer.than.twenty.byte.catalog.name ')
 /* if catalog name is longer that 20 bytes */
RNLDEF RNL(EXCL) TYPE(PATTERN) QNAME(SYSIGGV2) RNAME
('????????????????????????????????????????????
```

```
   your.outside.sysplex.catalog')
RNLDEF RNL (EXCL) TYPE (GENERIC) QNAME (SYSZVVDS) RNAME(volser)
RNLDEF RNL (EXCL) TYPE (GENERIC) QNAME (SYSVTOC) RNAME(volser)
```

The resources SYSVTOC and SYSZVVDS are protected by the code issuing the RESERVE macro. If you look at how reserve processing is done, the first check is to see if the QNAME is found in the EXCLUSION RNL. If it is not found there, then the CONVERSION RNL is searched. There is not a problem with converting SYSIGGV2 and using reserves for SYSVTOC and SYSZVVDS as long as all catalogs are shared only within the GRSplex.

If all the DASD used outside of the GRSplex has an identifiable pattern, such as ????C?, then it could be handled as follows:

```
RNLDEF RNL(EXCL) TYPE(PATTERN) QNAME(SYSZVVDS) RNAME(????C?*)
```

When a reserve is issued for SYSZVVDS it can be issued in two forms. The first is just with the VOLSER, the second is for the VOLSER and a two byte hex value. The addition of the asterisk (*) at the end of the name will cover both situations.

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. This implementation does not provide the positive performance that using RESERVE CONVERSION RNL provides.

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 *OS/390 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.

- 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 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.

Related Reading: See *IMS Version 9: Utilities Reference: System* for information about the DFSMDA macro.

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 82 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.

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 X (index_ci_size + 120) X (data_ci_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.

Related Reading: See *IMS Version 9: Customization Guide* for further details about using the DSPBUFFS Buffer Size Specification Facility.

CYLINDERS
Specifies the amount of space to allocate to the cluster.

FREESPACE
The default values of FREESPACE(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.

KEYS
KEYS(32 0) is required.

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 (4086,8182) 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 theRECON 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 5 shows how this determination is made. The columns in Table 5 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 5. Determining Which RECON Data Sets Are Accessed

Case	DD Statement	Data Set Status	Data Set Use	DBRC Selection Criteria
1	RECONA	Create Mode	Copy1	Not selected until `INIT.RECON` command is specified
	RECONB	Create Mode	Copy2	Not selected until `INIT.RECON` command is specified
	RECONC	Create Mode	Spare	
2	RECONA	RECON	Copy1	Current copy of RECON
	RECONB	Create Mode	Copy2	Produced from Copy1
	RECONC	Create Mode	Spare	
3	RECONA	RECON	Copy1	Current copy of RECON
	RECONB	RECON	Copy2	Current copy of RECON
	RECONC	Create Mode	Spare	
4	RECONA	RECON	Copy1	Current copy of RECON
	RECONB	RECON	Unused	Older copy than RECONA
	RECONC	Create Mode	Copy2	Produced from Copy1

Table 5. Determining Which RECON Data Sets Are Accessed (continued)

Case	DD Statement	Data Set Status	Data Set Use	DBRC Selection Criteria
5	RECONA	RECON	Copy1	Current copy of RECON
	RECONB	RECON	Unused	Older copy than RECONA
	RECONC	RECON	Unused	Older copy than RECONA
6	RECONA	RECON	Copy1	Current copy of RECON
	RECONB	RECON	Copy2	Current copy of RECON
	RECONC	RECON	Unused	Older copy than RECONA
7	RECONA	RECON	Copy1	Current copy of RECON
	RECONB	Create Mode	Copy2	Produced from Copy1
8	RECONA	Create Mode	Copy1	Not selected until `INIT.RECON` command is specified
	RECONB	Create Mode	Copy2	Not selected until `INIT.RECON` command is specified
9	RECONA	RECON	Copy1	Current copy of RECON
	RECONB	RECON	Copy2	Current copy of RECON
10	RECONA	RECON	Copy1	Current copy of RECON
	RECONB	RECON	Unused	Older copy than RECONA
11	RECONA	Create Mode	None	Discontinue processing
12	RECONA	RECON	Copy1	Current copy of RECON

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 on page 69 shows the major relationships between RECON record types. These relationships are described in detail in the sections that follow Figure 10 on page 69.

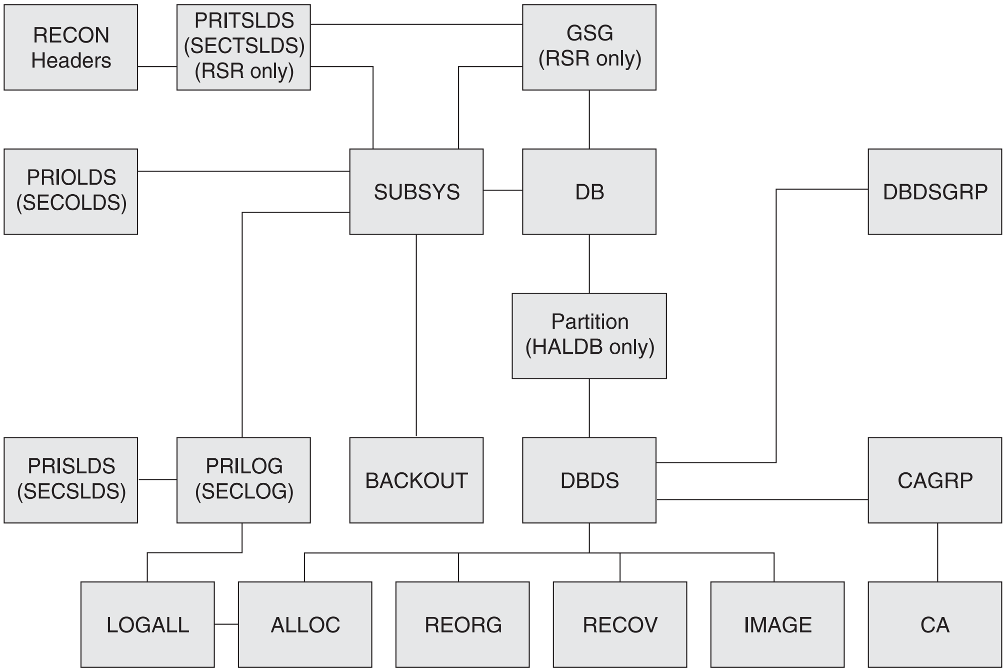


Figure 10. Major RECON Record Types and the Relationships Between Them

Related Reading:

- See Appendix B, “Sample Listings of RECONs,” on page 469 for details of what the records contain.
- See *IMS Version 9: Diagnosis Guide and Reference*, ZES1-2343 for a complete listing of RECON keys.

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.

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
  - PRIOLDS (primary online log data set)
  - SECOLDS (secondary online log data set)
  - IPRIOL (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 10 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 IPRIOL 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.

Related Reading: See *IMS Version 9: Operations Guide* for more information about interim primary and interim secondary-log data sets.

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)
- Database Data Set Group Record (DBDSGRP)
- Database Records (DB)
  - Database Record (DL/I and Fast Path)
  - Fast Path Area Authorization Record
  - DBDS (DL/I) / Area Recovery (FP) Record
- Global Service Group Record (GSG)
- Image Copy Record (IMAGE)
- Reorganization Record (REORG)
- Log Allocation Record (LOGALL)
- Database Allocation Record (ALLOC)
- Recovery Record (RECOV)
- Subsystem Record (SSYS)

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 DBs and DEDB areas that can be named in the DATAGROUP parameter for the `/STA`, `/STO`, and `/DBR` commands
- RECOVGRP, a group of DL/I DBs 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`, `/STO`, and`/DBR` commands, the use of a DBDSGRP is not recommended because it is inefficient.

Related Reading: See “DBDS Groups” on page 49 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
- 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.

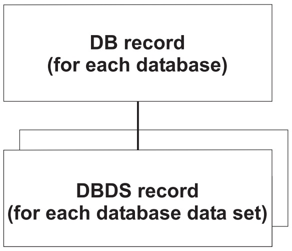


Figure 11. DBRC DL/I Records

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 shows the relationship of the RECON data set records that represent a HALDB.

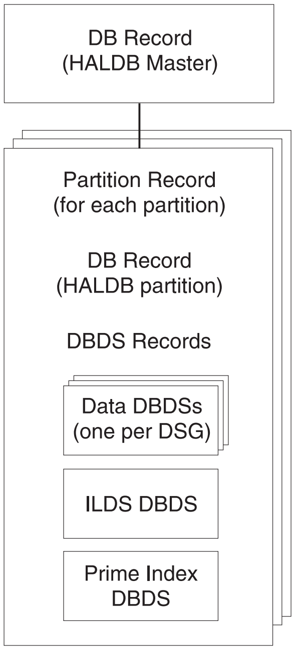


Figure 12. Record Structure for a HALDB in the RECON Data Set

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.
- 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 Version 9: 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 on page 75

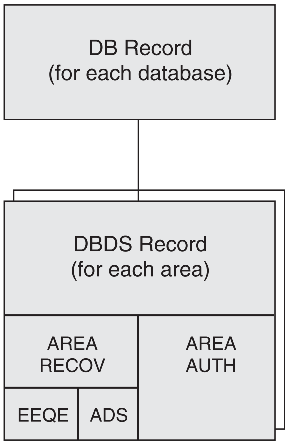


Figure 13. DBRC Fast Path Database Records

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 of RECONs,” 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.GSG` command and can be deleted by the `DELETE.GSG` 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.

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 206 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 325 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 copies
- Reorganizing the data sets
- Extending the data sets to obtain more space when necessary
- Deleting unnecessary RECON log records
- Compressing the PRILOG record
- Replacing a damaged RECON data set

- Replacing a discarded RECON data set
- Recovering the RECON data set
- Preserving the integrity of the RECON data set with DBRC Command Authorization Support

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 137 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

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 212 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.

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.

Recommendation: 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.

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 82 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 82for 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 upon your RECON data set set up and situation. The following paragraphs discuss various scenarios 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(gsg) command for each started ILS instance. Then issue START ILS(gsg) 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 REPRO 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 REPRO command to restore one of the RECON data sets.
5. Use the AMS REPRO 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 on page 81. 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.

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 83 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 65.

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 181 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 Version 9: Common Service Layer Guide and Reference* for introductory information about IMSplexes.
- See the *IMS Version 9: Customization Guide* for more information on the SCI Registration exit routine.
- See the *IMS Version 9: Utilities Reference: Database and Transaction Manager* and the *IMS Version 9: Installation Volume 2: System Definition and Tailoring* for more information about the IMSPLEX EXEC parameter.

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 84

- “Authorizing Commands with the DBRC Command Authorization Exit Routine (DSPDCAX0)” on page 85
- “Authorizing Commands with Both the DBRC Command Authorization Exit Routine (DSPDCAX0) and RACF” on page 85

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 Version 9: Utilities Reference: Database and Transaction Manager*, ZES1-2358 for more information about the HALDB Partition Definition utility.

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)

DBRC uses a list of resource names for command authorization support. See Appendix C, “Resource Names for Command Authorization,” on page 553

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 PRODRECN for the production RECON data sets and an HLQ name of TESTRECN 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 181 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 Version 9: Customization Guide*, ZES1-2341, 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 Version 9: 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 Version 9: 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

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 6 lists the eight GENJCL commands and what they generate.

Table 6. GENJCL Commands and What They Generate

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

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 keywords are data set names and volume information. DBRC performs the keyword substitution and then generates the JCL or user output you requested by issuing the GENJCL command.

Related Reading: See Appendix A, “Symbolic Keywords Recognized by DBRC,” on page 429 for a complete list of the symbolic keywords that are recognized by DBRC.

IBM supplies a JOB statement execution member that is used by all GENJCL commands. If the IBM-supplied skeletal JCL execution members meet your general requirements, you can modify them slightly to provide installation-specific information. Information on what needs to be modified is contained in “Using IBM-Supplied Skeletal JCL.”

If the IBM-supplied skeletal JCL does not meet your general requirements or if you plan to use the GENJCL.USER command, you must write your own skeletal JCL members or define new keywords to include in the IBM-supplied skeletal JCL. Information on these topics is in “Writing Your Own Skeletal JCL” on page 89.

Using IBM-Supplied Skeletal 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 “Understanding the IBM-Supplied Skeletal JCL Execution Members” on page 438 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 DFSWADS0 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 90 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 88) 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 108 for more information).

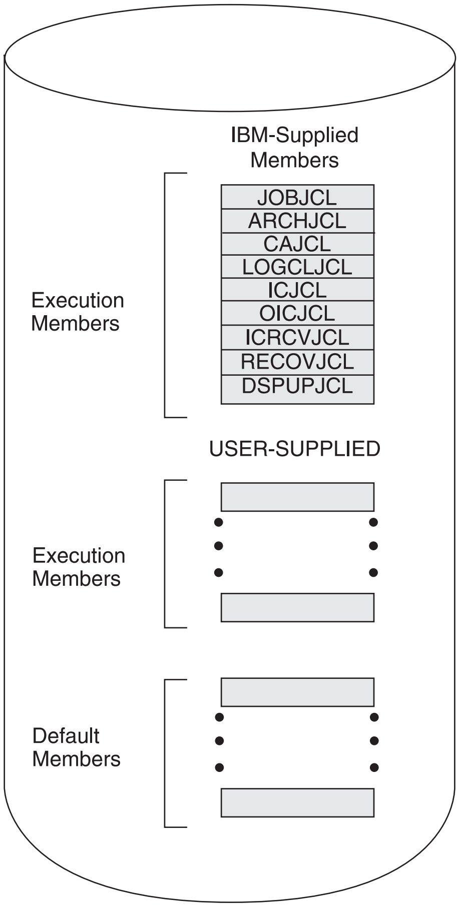


Figure 14. Skeletal JCL Data Set Contents

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 92
- "Writing Control Keywords" on page 98
- "Understanding Skeletal JCL Default Members" on page 107

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 429 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_type*s:

- 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 98.

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 7 on page 94

Table 7. Records That Can Be Selected Using the %SELECT Keyword

record_type	What Is Selected
OLDS	Specifies that OLDSs are to be selected. If dual logging is in effect, both PRIOLDS and SECOLDS can be selected.
SLDS	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 SECSLD 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.
SSLDS	Specifies that SECSLDs are to be selected.
RLDS	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.
SRLDS	Specifies that SECRLDSs are to be selected.
IC	Specifies that image copy data sets are to be selected.
CA	Specifies that change accumulation data sets are to be selected.
ALLOC	Specifies that DBDS allocation records are to be selected.
DBDS	Specifies that DBDS records are to be selected.

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

Specifies the DBDS with which the selected records are to be associated. The DBDS can be specified as dbname, ddname, 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

Specifies 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 130 in Chapter 7, “DBRC Commands,” on page 125

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 130 in Chapter 7, “DBRC Commands,” on page 125
- `%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

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 %CAODSN keyword, which is the name of the input change accumulation data set.

The syntax of the %SET MEMBER keyword is:

```
►►──%SET_MEMBER=newmbrname──────────────────────────────────────────►◄
```

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 132 for more information on time stamps.

For GENJCL.USER, the default is:

```
TIMEFMT(O,O,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(O,O,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:

```
►►──%SET_TIMEFMT(subparm,[subparm],...)──────────────────────────────►◄
```

Related Reading: For detailed information about the TIMEFMT keyword, its parameters, and format, see "TIMEFMT Parameter" on page 132 Remember that the TIMEFMT keyword affects GENJCL output only if it is issued via the GENJCL command or from a %SET statement in the skeletal JCL.

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:

```
►►──%SELECT──OLDS(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 yydddhhmmsst{offset}.

%OLDCTIM The time the OLDS was closed. DBRC sets %OLDCTIM in the form yydddhhmmsst{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.

```
%SELECT     OLDS(IMSA,UNARCH)
//%OLDSDDN  DD DSN=%OLDSDSN,DISP=SHR
%ENDSEL
```

The JCL generated by this select group might be:

```
//DFSOLP00  DD  DSN=IMS.OLDSP00,DISP=SHR
//DFSOLS00  DD  DSN=IMS.OLDSS00,DISP=SHR
//DFSOLP01  DD  DSN=IMS.OLDSP01,DISP=SHR
//DFSOLS01  DD  DSN=IMS.OLDSS01,DISP=SHR
```

Example 2: The following select group generates a list of all OLDSs belonging to subsystem IMSA:

```
%SELECT       OLDS(IMSA,ALL)
%OLDSTYPOLDS  DD NAME=%OLDSDDN
              DSN=%OLDSDSN
              CLOSE TIME=%OLDSCTIM
%ENDSEL
```

The output generated by this select group might be:

```
POLDS  DD NAME=DFSOLP00
       DSN=IMS.POLDS00
       CLOSE TIME=842351638193+0012
POLDS  DD NAME=DFSOLP01
       DSN=IMS.POLDS01
       CLOSE TIME=842351712246+0055
POLDS  DD NAME=DFSOLP02
       DSN=IMS.POLDS02
       CLOSE TIME=000000000000+0000
```

Selecting SLDSs

The syntax of the %SELECT keyword to select SLDS is:

```
►►──%SELECT──slds_type(ssid,time_qualifier)────────────────────────────►◄
```

slds_type
Can be specified as SLDS (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 if 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 101

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 94

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 yydddhhmmsst{offset}.

%SLDETIM The stop time of the SLDS. DBRC sets the %SLDETIM in the form yydddhhmmsst{offset}.

%SLDOTIM The start time of the PRISLDS/SECSLDS. DBRC sets %SLDOTIM in the form *yydddhhmmsst{offset}*.

%SLDSSEL 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.

```
%SELECT       SLDS(IMSA,LAST)
LATEST SLDS:  DSN=%SLDSDSN
               STOP TIME=%SLDETIM
%ENDSEL
```

The output generated by this select group might be:

```
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:

```
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.

```
%SELECT   SLDS(IMSA,FROM(840031903298))
%DELETE   (%SLDSDSN EQ '')
//SLDS    DD  DSN=%SLDSDSN,DISP=OLD,
//            UNIT=%SLDUNIT,
//            VOL=SER=(%SLDVOLS),
//            LABEL=(1,SL)
%ENDDEL
%ENDSEL
```

The generated DD statements might be:

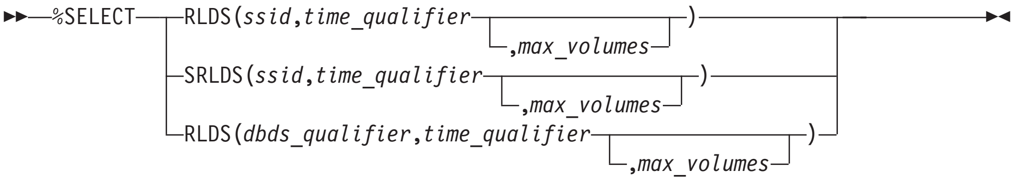
```
//SLDS    DD  DSN=IMS.SLDS1,DISP=OLD,
//            UNIT=3400,
//            VOL=SER=(VOLUM1,VOLUM2,VOLUM3),
//            LABEL=(1,SL)
//        DD  DSN=IMS.SLDS2,DISP=OLD,
//            Unit=3400,
//            VOL=SER=(VOLUM4,VOLUM5,VOLUM6,
//            VOLUM7,                               C
// 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:



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 94

dbds_qualifier
DBDS qualifier as specified in "Understanding the Selection Criteria Parameter" on page 94 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 yydddhhmmsst{offset}.

%LOGETIM The stop time of the RLDS. DBRC sets %LOGETIM in the form yydddhhmmsst{offset}. If the data set is still open, the time is set to 000000000000+0000.

%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.

%LOGFRID The log record sequence number of the first log record of the RLDS.

%LOGLRID The log record sequence number of the last log record of the RLDS.

%LOGOTIM 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.

```
%SELECT   RLDS(BATCHJOB,LAST)
%DELETE   (%LOGETIM NE '000000000000+0000')
//LOGDD   DD  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:

```
//LOGDD   DD  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:

```
►►──%SELECT──IC(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 94

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 ICSTOP is null.

%ICTIME The run time of the image copy. DBRC sets %ICTIME in the form yydddhhmmsst{offset}.

%ICUNIT The unit type of the image copy data set if it is a NONHSSP type; otherwise, ICUNIT is null.

%ICVCNT The number of volumes of the image copy data set if it is a NONHSSP type; otherwise, ICVCNT is null.

%ICVOLS The volume serial number list of the image copy data set if it is a NONHSSP type; otherwise, `ICVOLS` is null.

%ICUSID The update set identifier (USID).

%ICCAT Set to `YES` if the image copy is cataloged (HSSP CICs only). Otherwise, `ICCAT` is set to `NO`.

%IC2SEL 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:

%IC2DSN The data set name of the duplicate image copy data set.

%IC2FSEQ The file sequence number of the duplicate image copy data set. If the IC was created by HSSP, `IC2FSEQ` is set to null.

%IC2UNIT The unit type of the duplicate image copy data set. If the IC was created by HSSP, `IC2UNIT` is set to null.

%IC2VCNT The number of volumes of the duplicate image copy data set. If the IC was created by HSSP, `IC2VCNT` is set to null.

%IC2VOLS 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 ddname 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)
```

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 94

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:

%CADSN 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 `yydddhhmmsst{offset}`.

%CATIME The change accumulation data set time in the form `yydddhhmmsst{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,                            C
//  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:

```
►►──%SELECT──ALLOC(dbds_qualifier,time_qualifier)──────────────────────────►◄
```

```
►►──%SELECT──ALLOC(PRILOG,time_qualifier)──────────────────────────────────►◄
```

dbds_qualifier and time_qualifier

DBDS qualifier and time qualifier are as specified in “Understanding the Selection Criteria Parameter” on page 94 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:

```
%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 94 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, the update set identifier (USID) of the area. 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')
//%DBADDN   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
```

Using the DEFAULTS parameter and assuming the values are not overridden, the GENJCL command generates the following:

```
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.OIC, 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

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(O,O,P,4,PERM)
```

or

```
CHANGE.RECON TIMEFMT(O,O,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 USER01:

```
%SELECT SLDS(%USSID,LAST)
%ENDSEL
 SLDSDD  BEG=%SLDSTIM
         END=%SLDETIM
         VOL=%SLDVOLS
         UNT=%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 USER01:

```
GENJCL.USER MEMBER(USER01),-
            NOJOB, USERKEYS((%USSID,'SYS3')),-
            JCLOUT(SYSPRINT)
```

3. Below is the SLDS listing:

```
PRISLD
 START = 96.296 11:51:21.8              *    SSID=SYS3     VERSION=6.1
 STOP  = 96.296 13:08:18.4                   #DSN=4
 GSGNAME=**NULL**
 FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 0

 DSN=IMSVS.SLDSP.SYS3.D96296.T1151218.V06                 UNIT=SYSDA
 START = 96.296 11:51:21.8                FIRST DS LSN= 0000000000000001
 STOP  = 96.296 11:52:08.7                LAST  DS LSN= 00000000000002B5
 FILE SEQ=0001    #VOLUMES=0001

  VOLSER=000000 STOPTIME = 96.292 11:52:08.7
    CKPTCT=1    CHKPT ID = 96.292 11:51:26.8

 DSN=IMSVS.SLDSP.SYS3.D96296.T1152087.V00                 UNIT=SYSDA
 START = 96.296 11:52:08.7                FIRST DS LSN= 00000000000002B6
 STOP  = 96.296 11:52:51.6                LAST  DS LSN= 000000000000036D
 FILE SEQ=0001    #VOLUMES=0001

  VOLSER=000000 STOPTIME = 96.296 11:52:51.6
    CKPTCT=0    CHKPT ID = 96.296 11:51:26.8

 DSN=IMSVS.SLDSP.SYS3.D96296.T1152516.V00                 UNIT=SYSDA
 START = 96.296 11:52:51.6                FIRST DS LSN= 000000000000036E
 STOP  = 96.296 13:08:16.9                LAST  DS LSN= 00000000000004E5
 FILE SEQ=0001    #VOLUMES=0001

  VOLSER=000000 STOPTIME = 96.296 13:08:16.9
    CKPTCT=2    CHKPT ID = 96.296 11:52:52.0

 DSN=IMSVS.SLDSP.SYS3.D96296.T1152516.V01                 UNIT=SYSDA
 START = 96.296 13:08:16.9                FIRST DS LSN= 00000000000004E6
 STOP  = 96.296 13:08:18.4                LAST  DS LSN= 0000000000000571
 FILE SEQ=0001    #VOLUMES=0001

  VOLSER=000000 STOPTIME = 96.296 13:08:18.4
    CKPTCT=0    CHKPT ID = 96.296 13:08:16.8
```

Below is the output that results from issuing the previous sample GENJCL.USER command:

```
SLDSDD  BEG=962961308169
        END=962961308184
        VOL=000000
        UNT=SYSDA
        DSN=IMSVS.SLDSP.SYS3.D96296.T1152516.V01
```

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 copies exist that can be used.
  - Message DSP0063I was displayed.

  See “Solutions for Situation #1” on page 111 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 for Situation #2” on page 112 on how to correct this.

Solutions for Situation #1

Either of the following will solve situation number one above:

- 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 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) -
                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 for Situation #2

- 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(IMSA) DSN(DUMMY) FIRSTREC(11)
  -STARTIME(020011314544) VOLSER(SC3390)
NOTIFY.PRILOG SSID(IMSA)
  STARTIME(020011314544) - RUNTIME(020021315001) LASTREC(15)
CHANGE.PRILOG STARTIME(020011314544)
  ERROR - DSSTART(020021315000)
```

Below is the PRILOG as it appears in the RECON data set before issuing the NOTIFY.PRILOG command to close it:

```
PRILOG                                       RECORD SIZE=     464
 START = 02.001 13:14:54.4              *    SSID=IMSA     VERSION=8.1
 STOP  = 00.000 00:00:00.0                   #DSN=2
 GSGNAME=**NULL**
 FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 0
 DSN=LOG1                                                 UNIT=3400
 START = 02.001 13:14:54.4                FIRST DS LSN= 0000000000000001
 STOP  = 02.001 13:15:00.0                LAST  DS LSN= 0000000000000005
 FILE SEQ=0001    #VOLUMES=0001
 VOLSER=SC3390 STOPTIME = 02.001 13:15:00.0
      CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
          LOCK SEQUENCE#=  000000000000
 DSN=LOG2                                                 UNIT=3400
 START = 02.001 13:15:00.0                FIRST DS LSN= 0000000000000006
 STOP  = 02.002 13:15:00.0                LAST  DS LSN= 000000000000000A
 FILE SEQ=0001    #VOLUMES=0001
     VOLSER=SC3390 STOPTIME = 02.002 13:15:00.0
       CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
          LOCK SEQUENCE#=  000000000000
```

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 = 02.001 13:14:54.4              *    SSID=IMSA     VERSION=8.1
 STOP  = 02.002 13:15:00.1                   #DSN=3
 GSGNAME=**NULL**
 FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 0
 DSN=LOG1                                                 UNIT=3400
 START = 02.001 13:14:54.4                FIRST DS LSN= 0000000000000001
 STOP  = 02.001 13:15:00.0                LAST  DS LSN= 0000000000000005
 FILE SEQ=0001    #VOLUMES=0001
     VOLSER=SC3390 STOPTIME = 02.001 13:15:00.0
       CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
         LOCK SEQUENCE#=  000000000000
 DSN=LOG2                                                 UNIT=3400
 START = 02.001 13:15:00.0                FIRST DS LSN= 0000000000000006
 STOP  = 02.002 13:15:00.0                LAST  DS LSN= 000000000000000A
 FILE SEQ=0001    #VOLUMES=0001
```

```
         VOLSER=SC3390 STOPTIME = 02.002 13:15:00.0
            CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
         LOCK SEQUENCE#=  000000000000

DSN=DUMMY                                         ERROR  UNIT=3400
START = 02.002 13:15:00.0                FIRST DS LSN= 000000000000000B
STOP  = 02.002 13:15:00.1                LAST  DS LSN= 000000000000000F
FILE SEQ=0001     #VOLUMES=0001
    VOLSER=SC3390 STOPTIME = 02.002 13:15:00.1
    CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
          LOCK SEQUENCE#=  000000000000
```

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 212 and “DELETE.LOG (for RLDS and SLDS)” on page 212 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(XXXXX) STARTRCV
CHANGE.SUBSYS SSID(XXXXX) ENDRECOV  <---(removes auth from DB)
DELETE.SUBSYS SSID(XXXXX)
```

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(areaname) 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:

```
T1--T2---T3--T4-T5--------T6--------T7---------T8----T9---


|---A1---D1--|--------------------A4-----------------// (open)
Log1 (DSN1)    (OPEN OLDS)

                |--A2--|  |--A3-|
                Log2      Log3
```

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 171 and the CHANGE.SECLOG commands starting with "CHANGE.SECLOG (for OLDS)" on page 191

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 per RECON data set.

To specify that these data sets are cataloged specify:

```
   //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-

```
   //INITRCON JOB
⋮
   //SYSIN   DD  *
        INIT.RECON  NOCATDS
   /*
```

Related Reading: See "INIT.RECON" on page 280 and "CHANGE.RECON" on page 181 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(960031314544) RUNTIME(960031314545) -
 LASTREC(015) OLDS(DFSOLP02) SSID(IMSA)
CHANGE.PRILOG OLDS(DFSOLP02) ARCHIVED SSID(IMSA)
```

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=IMSA            # DD ENTRIES=3
 EARLIEST CHECKPOINT = 96.001 14:22:22.2

 DDNAME=DFSOLP00   DSN=IMSA.OLDP00
 START = 96.001 13:14:54.4                FIRST DS LSN= 0000000000000001
 STOP  = 96.002 13:14:54.4                LAST  DS LSN= 0000000000000005
 STATUS=ARC COMPLT                                FEOV=NO    AVAIL
 PRILOG TIME=96.001 13:14:54.4                ARCHIVE JOB NAME=JT185209
 VERSION=6.1

 DDNAME=DFSOLP01   DSN=IMSA.OLDP01
 START = 96.002 13:14:54.4                FIRST DS LSN= 0000000000000006
 STOP  = 96.003 13:14:54.4                LAST  DS LSN= 000000000000000A
 STATUS=ARC COMPLT                                FEOV=NO    AVAIL
 PRILOG TIME=96.001 13:14:54.4                ARCHIVE JOB NAME=JT185210
 VERSION=6.1

 DDNAME=DFSOLP02   DSN=IMSA.OLDP02
 START = 96.003 13:14:54.4                FIRST DS LSN= 000000000000000B
 STOP  = 00.000 00:00:00.0                LAST  DS LSN= 0000000000000000
 STATUS=ACTIVE                                    FEOV=NO    AVAIL
 PRILOG TIME=96.001 13:14:54.4                ARCHIVE JOB NAME=JT185211
 VERSION=6.1
```

The PRIOLD record after issuing the commands to close and mark DFSOLP02 as archived (with the differences highlighted):

```
PRIOLD
 SSID=IMSA            # DD ENTRIES=3
 EARLIEST CHECKPOINT = 96.001 14:22:22.2

 DDNAME=DFSOLP00   DSN=IMSA.OLDP00
 START = 96.001 13:14:54.4                FIRST DS LSN= 0000000000000001
 STOP  = 96.002 13:14:54.4                LAST  DS LSN= 0000000000000005
 STATUS=ARC COMPLT                                FEOV=NO    AVAIL
 PRILOG TIME=96.001 13:14:54.4                ARCHIVE JOB NAME=JT185209
 VERSION=6.1

 DDNAME=DFSOLP01   DSN=IMSA.OLDP01
 START = 96.002 13:14:54.4                FIRST DS LSN= 0000000000000006
```

```
STOP  = 96.003 13:14:54.4                LAST  DS LSN= 000000000000000A
STATUS=ARC COMPLT                                FEOV=NO    AVAIL
PRILOG TIME=96.001 13:14:54.4                ARCHIVE JOB NAME=JT185210
VERSION=6.1

DDNAME=DFSOLP02   DSN=IMSA.OLDP02
START = 96.003 13:14:54.4                FIRST DS LSN= 000000000000000B
STOP  = 96.003 13:14:54.5                LAST  DS LSN= 000000000000000F
STATUS=ARC COMPLT                                FEOV=NO    AVAIL
PRILOG TIME=96.001 13:14:54.4                ARCHIVE JOB NAME=JT185211
VERSION=6.1
```

Avoiding Some Causes of RECON Data Set Enqueue Problems

The following sections describe some scenarios that can cause RECON data set enqueue problems and contain suggestions for avoiding these problems.

Related Reading: See “RECON Data Set Serialization” on page 59 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 Version 9: 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

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 Online Commands" on page 126
- "Command Syntax" on page 126

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.GSG` 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 126).

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 Version 9: 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.
- See "How to Read Syntax Diagrams" on page 128 for information about the syntax diagrams and notations used in this book.
- See "Data Set Naming Conventions" on page 35 for information on naming conventions.

DBRC Online Commands

Certain DBRC commands can be issued from an online IMS environment (/RMxxxxxx commands). The online DBRC commands are:

- /RMCHANGE
- /RMDELETE
- /RMGENJCL
- /RMINIT
- /RMLIST
- /RMNOTIFY

Related Reading: See *IMS Version 9: 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 83.
- For the details of using the DBRC Command Authorization Exit (DSPDCAX0), see the *IMS Version 9: Customization Guide*.

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 127
- “Continuation Characters” on page 127
- “Comments” on page 127
- “Commands” on page 127
- “Parameters” on page 127
- “How to Read Syntax Diagrams” on page 128
- “DBRC Time Stamps” on page 130

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.

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. Such a character string is assumed 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:

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

Each syntax diagram in this book begins with a double right arrow and ends with a right and left arrow pair. Lines that begin with a single right arrow are continuation lines. You read a syntax diagram from left to right and from top to bottom, following the direction of the arrows.

Table 8 describes the conventions that are used in syntax diagrams in this information:

Table 8. How to Read Syntax Diagrams

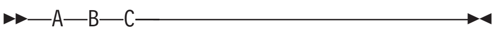
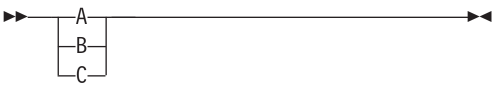
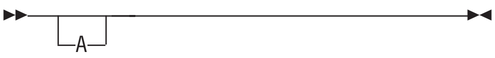
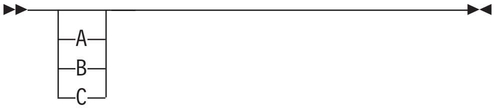
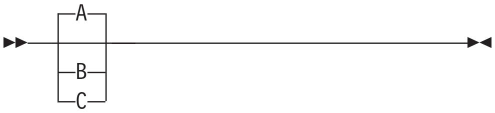
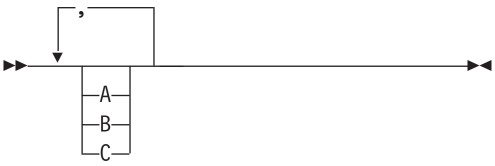
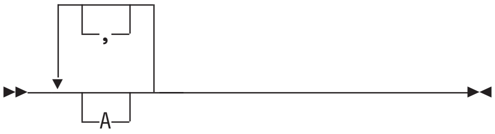
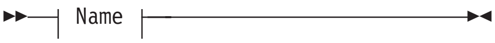
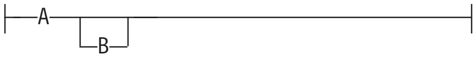
Convention	Meaning
	You must specify values A, B, and C. Required values are shown on the main path of a syntax diagram.
	You must specify value A, B, or C.

Table 8. How to Read Syntax Diagrams (continued)

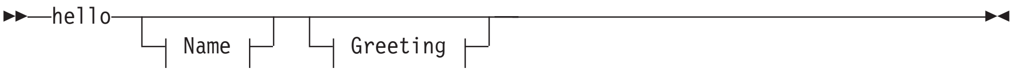
Convention	Meaning
	You have the option to specify value A. Optional values are shown below the main path of a syntax diagram.
	You have the option to specify A, B, C, or none of these values.
	You have the option to specify A, B, C, or none of these values. If you don't specify a value, A is the default.
	You have the option to specify one, more than one, or none of the values A, B, or C. Any required separator for multiple or repeated values (in this example, the comma) is shown on the arrow.
	You have the option to specify value A multiple times. The separator in this example is optional.
 **Name:** 	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.
Punctuation marks and numbers	Enter punctuation marks (slashes, commas, periods, parentheses, quotation marks, equal signs) and numbers exactly as shown.
Uppercase values	Keywords, their allowable synonyms, and reserved parameters appear in uppercase letters for z/OS. Enter these values exactly as shown.
Lowercase values	Keywords, their allowable synonyms, and reserved parameters appear in lowercase letters for UNIX. Enter these values exactly as shown.
Lowercase values in italic (for example, *name*)	Supply your own text or value in place of the *name* variable.
ƀ	A ƀ symbol indicates one blank position.

Other syntax conventions include the following:

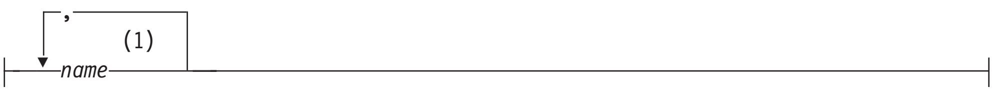
- When you enter commands, separate parameters and keywords by at least one blank if there is no intervening punctuation.
- Footnotes are shown by a number in parentheses, for example, (1).
- Parameters with number values end with the symbol #.
- Parameters that are names end with 'name'.
- Parameters that can be generic end with the symbol *.

Syntax Diagram Example

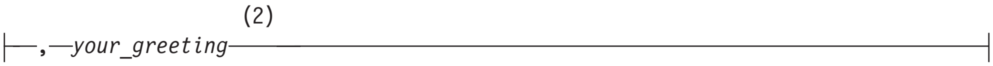
Here is an example syntax diagram that describes the `hello` command.



Name:



Greeting:



Notes:

1 You can code up to three names.

2 Compose and add your own greeting (for example, how are you?).

According to the syntax diagram, these commands are all valid versions of the `hello` command:

```
hello
hello name
hello name, name
hello name, name, name
hello, your_greeting
hello name, your_greeting
hello name, name, your_greeting
hello name, name, name, your_greeting
```

The space before the *name* value is significant. If you do not code *name*, you must still code the comma before *your_greeting*.

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 181 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 134

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:

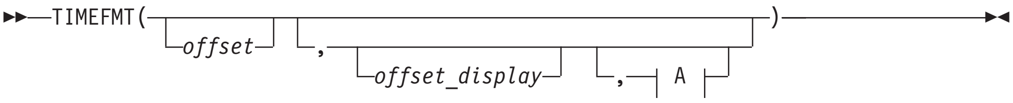
```
'94.252 08:24:45.7 -8'            '94/252-08.24.45.7 -8:00'
'94,252 08:24:45.7 PST'           '94/252-16.24.45.7 UTC'
'1994 252 16.24.45.7 +0'
```

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.

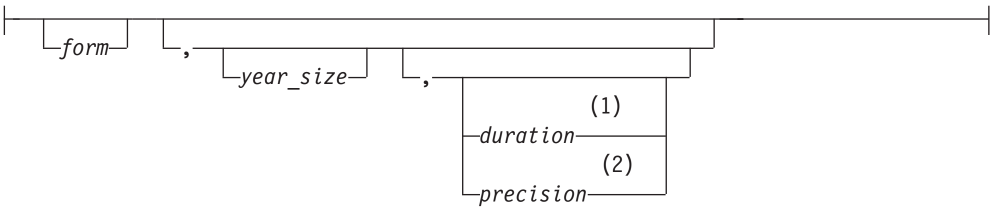
The TIMEFMT parameter can be used on `CHANGE.RECON`, `LIST.xxxx`, and `GENJCL.xxxx` commands or can be specified in a skeletal JCL member.

Restriction: The parameters that are associated with TIMEFMT are positional.

The format of the TIMEFMT parameter sublist is as follows:



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:

```
%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
```

Related Reading: See “Using the %SET TIMEFMT Keyword” on page 97 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 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 INIT.RECON or the upgrade utility 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 - yydddhhmmsst) is recorded in 12 bytes in the RECON as `1998030F|18000012|3456032d|` (yyyyddd hhmmsstx xxxxfqqs), where:

hh
UTC, 8 hours later than local time

x xxxx
Low-order digits ignored by DBRC (except for GENJCL.USER output with precision)

fqqs
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 may not be correct if a clock change has occurred since the time when the record was written in RECON.

With the wrong offset supplied in a command, DBRC does not find the record.

Time stamp supplied and wrong offset is used:

```
'1998.030 16:00:00.1 -02:00'
```

DBRC determines an incorrect search time:

```
1998030F 18000012 3456008D
```

Time stamps from different RECON listings, even with time stamps displayed in different formats, may 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
- LIST.LOG
- NOTIFY.ALLOC
- NOTIFY.BKOUT
- NOTIFY.CA
- NOTIFY.IC
- NOTIFY.PRILOG
- NOTIFY.RECOV
- NOTIFY.REORG
- NOTIFY.SECLOG
- NOTIFY.UIC

Related Reading: See “Standard Time Stamp Format” on page 130 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.

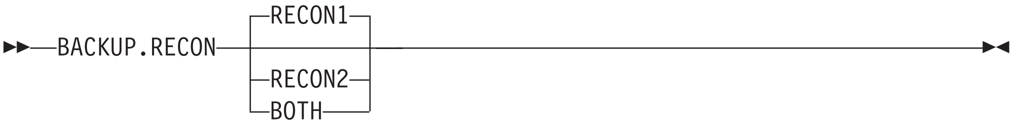
BACKUP.RECON Usage

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



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

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)─┘

                          ┌─UNAVAIL─┐
►──┬──────────────────┬───┼─────────┼─────────────────────────────────────►◄
   └─ADSN──(──name──)─┘   └─AVAIL───┘
```

Parameters for CHANGE.ADS

ADDN(*name*)
Required parameter you use to specify the area data set ddname of the ADS being changed.

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 ddname.

ADSN(*name*)
Optional parameter you use to identify the ADS being changed, by new data set 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.

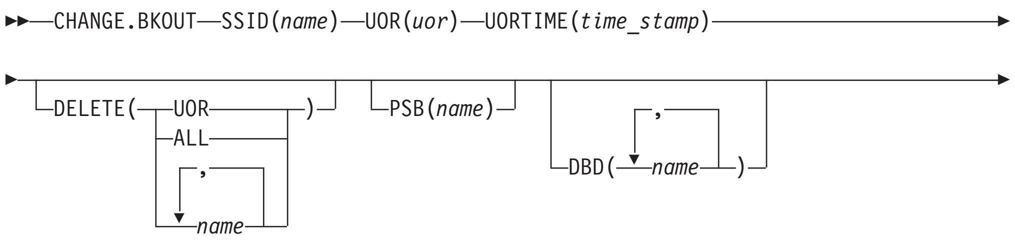
```
 //CHGADS   JOB

⋮
 //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



```
►─┬──────────────────┬──────────────────────────────────────────►◄
  │       ┌─,────┐   │
  │       ▼      │   │
  └─BKO(────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(E2E8E2F34040400000000600000003)-
                    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)──┬──────────────┬──────────────►
                                                   └─CADSN(name)──┘
```

```
►──┬────────────────┬──┬─────────┬──┬────────────────────────────┬──────────►
   └─FILESEQ(value)─┘  ├─INVALID─┤  │        ┌─3400─────┐        │
                       └─VALID───┘  └─UNIT(──┴─unittype─┴──)─────┘

►──┬─────────────────────────────┬──┬────────┬──────────────────────────────►◄
   │             ┌─,──────┐      │  ├─COMP───┤
   │             ▼        │      │  └─SUBSET─┘
   └─VOLLIST(────┴─volser─┴──)───┘
```

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 130).

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.

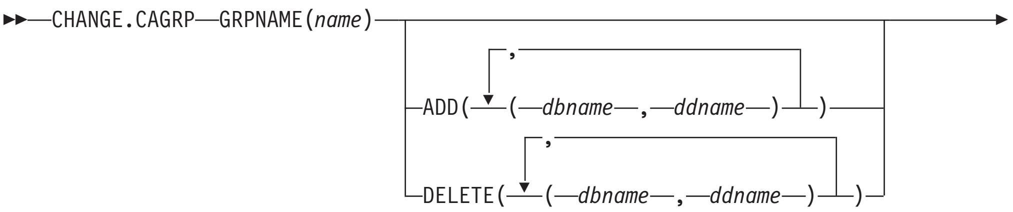
```
//CHGCA    JOB
 
⋮
//SYSIN    DD   *
     CHANGE.CA  GRPNAME(CAGRP2)  RECTIME(820650204335)   -
       INVALID  CADSN(IMS.CAGRP2.CA.CADSN2)  -
       VOLLIST(VOLCA1) FILESEQ(4)
/*
```

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



```
►─┬──────────────┬──┬──────────────────┬──┬───────────────┬──┬─────────┬──────────►◄
  └─CAJCL(member)─┘  ├─DEFLTJCL(member)─┤  └─GRPMAX(value)─┘  ├─NOREUSE─┤
                     └─NODEFLT──────────┘                     └─REUSE───┘
```

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 225

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.

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 GRPNAME(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 GRPNAME(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  GRPNAME(CAGRP3)  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.

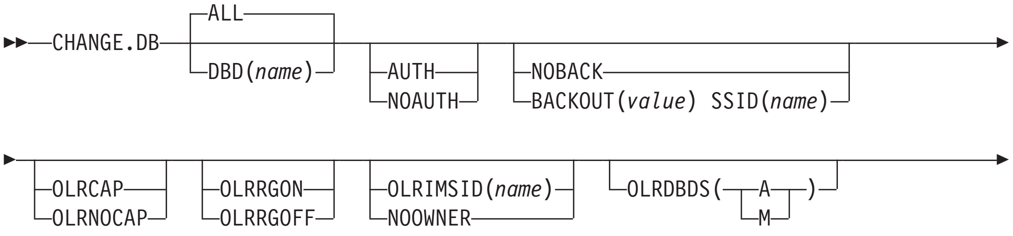
You can also use `CHANGE.DB` to remove a rare authorization inconsistency between the subsystem (SSYS) record and the database or area (DB/AREA) record. This inconsistency occurs when either the SSYS record still has an entry for the DB/AREA in its authorized databases/areas list but the DB/AREA record 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 only time that the ALL parameter is allowed is for `CHANGE.DB ALL OLRCAP`. 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 156 for a the description of MAS and NOMAS.

CHANGE.DB Syntax

Use one of the following two diagrams for the `CHANGE.DB` command.



```
►─┬─────────┬──┬─────────┬──┬─────────────────────┬──┬──────────┬──┬─────────┬──────────►
  ├─PINIT───┤  ├─READON──┤  └─SHARELVL(─┬─0─┬─)─┘  ├─NONRECOV─┤  ├─ICREQ───┤
  └─NOPINIT─┘  └─READOFF─┘              ├─1─┤      ├─RECOVABL─┤  └─NOICREQ─┘
                                        ├─2─┤      └─USERRCOV─┘
                                        └─3─┘

►─┬─────────┬──┬───────────────────┬──┬──────────┬──┬───────────────────┬──────────────►◄
  ├─TYPEFP──┤  ├─GSGNAME(gsgname)──┤  ├─RCVTRACK─┤  ├─PARTSEL(pgmname)──┤
  └─TYPEIMS─┘  └─NOTCOVER──────────┘  └─DBTRACK──┘  └─HIKEY─────────────┘
```

Or:

```
►►──CHANGE.DB──UNAUTH──DBD(name)─┬─────────────────────┬─SSID(name)──────────►
                                 │                 (1) │
                                 └─AREA(name)──────────┘

   ┌─ACTIVE───┐
►──┼──────────┼──────────────────────────────────────────────────────────────►◄
   └─TRACKING─┘
```

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, 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 148

Table 9. Parameter Definitions for HALDB Partition and Master

CHANGE.DB	*name* = TYPE=HALDB	*name* = TYPE=PART
AUTH | NOAUTH	No	Yes
NOBACK | BACKOUT	No	Yes
READON | READOFF	No	Yes
NORECOV | RECOVABL	Yes	No
OLRCAP | OLRNOCAP	Yes	No
OLRRGON | OLRRGOFF	No	Yes
OLRIMSID(name)NOOWNER	No	Yes
OLRDBDS(A | M)	No	Yes
SHARELVL	Yes	No
TYPEFP | TYPEIMS	No	No
GSGNAME | NOTCOVER	Yes	No

Table 9. Parameter Definitions for HALDB Partition and Master (continued)

CHANGE.DB	*name* = TYPE=HALDB	*name* = TYPE=PART
RCVTRACK | DBTRACK	Yes	No
UNAUTH	No	Yes
PINIT | NOPINIT	Yes	Yes
PARTSEL|HIKEY	Yes	No
ICREQ|NOICREQ	Yes	No

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 DBs 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 REORG 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 REORG of the DBDS. The IC NEEDED flag can still be set manually with the 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 | BACKOUT(*value*)

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 either the BACKOUT or UNAUTH parameters If coded without either 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. These parameters are meaningful for TYPEIMS and TYPEHALDB databases only, and are rejected for TYPEFP databases and HALDB partitions.

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. Therefore, they are not usable for recovery.

NONRECOV
Specifies that no recovery is to be performed on the database. NONRECOV cannot be specified for an RSR-covered database.

RECOVABL
Specifies that the database is recoverable and all updates performed for the DBDSs are to be registered 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:

- The NONRECOV, RECOVABL, and USERRCOV parameters are not valid with ALL.
- You cannot take concurrent image copies of nonrecoverable or user-recoverable databases.

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.

OLRRGON | OLRRGOFF
Mutually exclusive, optional parameters that specify whether a HALDB OLR is in progress. These parameters are valid only for a HALDB master.

OLRRGON Specifies that a HALDB OLR is in progress. Both the A-through-J and the M-through-V data sets contain active data.

OLRRGOFF Specifies that a HALDB OLR has completed and that either the A-through-J set of data sets or the M-through-V set of data sets contain active data. The flag is changed to indicate that the inactive data sets are now the active data sets. `OLREORG CURSOR ACTIVE = NO` is set in the DB record. The OLRIMSID is cleared.

Note: You might need to manually delete either the DB (HALDB) records or to add a stop time to them using the `DELETE.REORG` and `NOTIFY.REORG` commands. For more information, see "DELETE.REORG" on page 216 and "NOTIFY.REORG" on page 328.

OLRIMSID(*name*) | NOOWNER
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 Version 9: 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 Version 9: 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 155 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. Adding, changing, or deleting the partition selection exit program name causes all of the partition high keys or partition selection strings to be cleared and all of the partitions to be 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.

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    *
     CHANGE.DB   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.

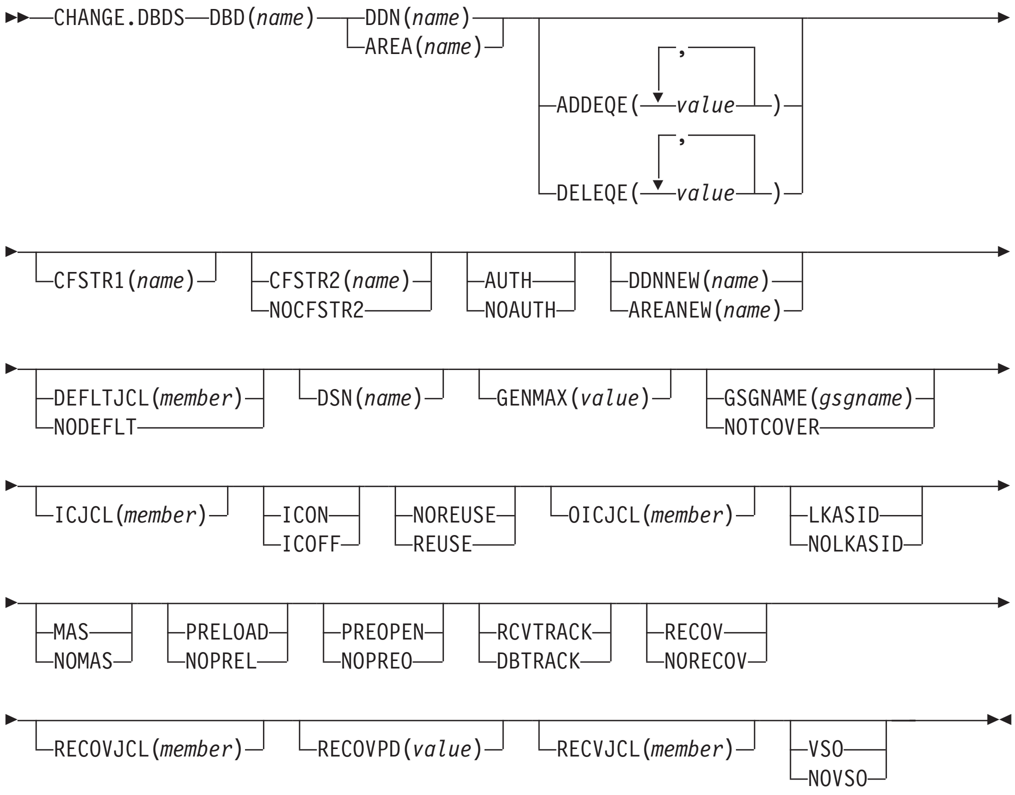
Some attributes of a HALDB DBDS may be changed with this command. See "CHANGE.PART" on page 167 if you want to change all the DBDSs of a HALDB partition.

Restrictions for HALDBs: For HALDBs, you can use this command only as indicated in Table 10 on page 155

Table 10. CHANGE.DBDS for HALDBs

CHANGE.DBDS	Partition Data DBDS	Partition Index/ILDS DBDS
ADDEQE ǀ DELEQE	Yes	Yes
CFSTR1	N/A	N/A
CFSTR2 ǀ NOCFSTR2	N/A	N/A
AUTH ǀ NOAUTH	N/A	N/A
DDNEW ǀ AREANEW	No	No
DEFLTJCL ǀ NODEFLT	Yes	No
DSN	No	No
GENMAX	Yes	No
GSGNAME ǀ NOTCOVER	N/A	N/A
ICJCL	Yes	No
ICON ǀ ICOFF	Yes	No
NOREUSE ǀ REUSE	Yes	No
OICJCL	Yes	No
LKASID ǀ NOLKASID	N/A	N/A
PRELOAD ǀ NOPREL	N/A	N/A
PREOPEN ǀ NOPREO	N/A	N/A
RCVTRACK ǀ DBTRACK	N/A	N/A
RECOV ǀ NORECOV	Yes	Yes
RECOVJCL	Yes	No
RECOVPD	Yes	No
RECVJCL	Yes	No
VSO ǀ NOVSO	N/A	N/A

CHANGE.DBDS Syntax



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 ddname 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 Version 9: 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.IC, GENJCL.OIC, 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.IC` 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 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 273 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 of the image copies 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 an 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

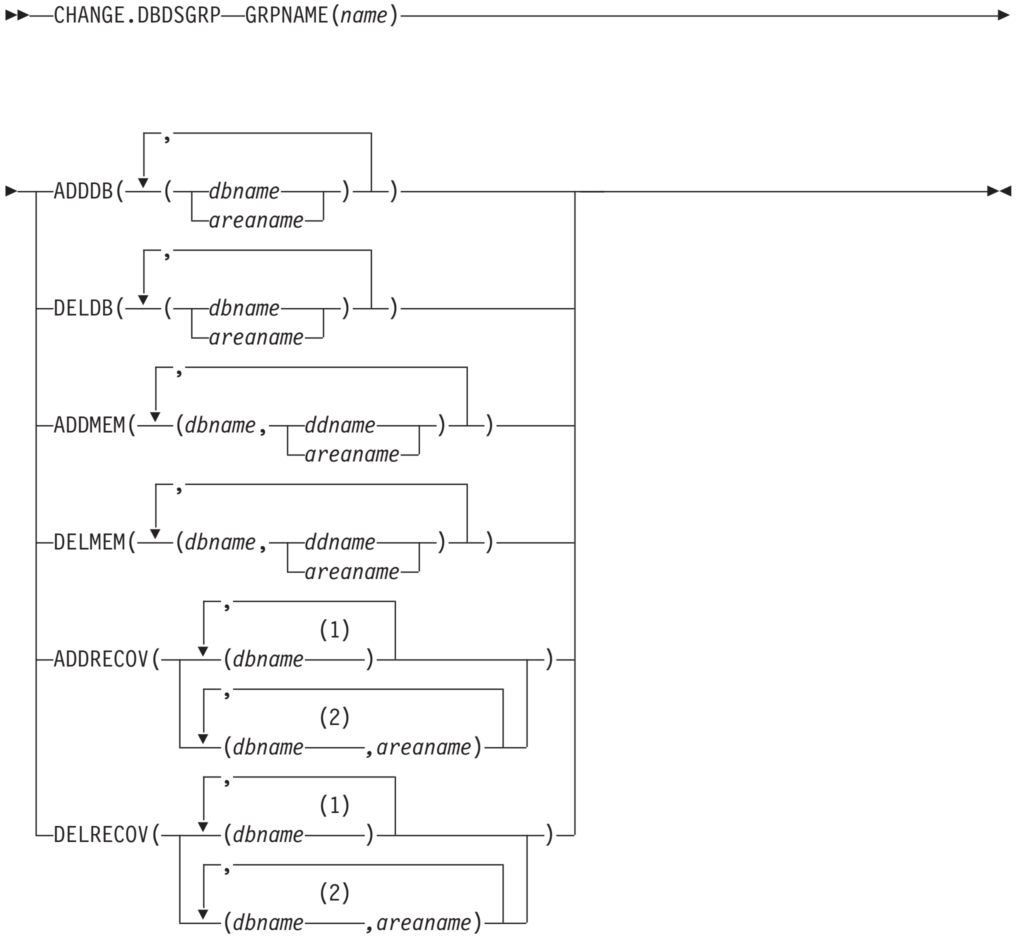
This example specifies changes to be made to the record in RECON for the Fast Path DEDB that is identified by the DBD and AREA parameters. The image copy data sets for the specified DEDB area are not reused, and the maximum number of images that DBRC maintains is two. In addition, the image copy data sets for the specified DBDS are kept in RECON for at least 15 days.

```
//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



Notes:

1 Full-function database name

2 Fast Path DEDB name

Parameters for CHANGE.DBDSGRP

GRPNAME(*name***)**
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***) |**
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.

ADDDB(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.

DELDB(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 DBs 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 11.

Table 11. Specifying HALDB Master or Partition Database Names for the CHANGE.DBDSGRP Command

CHANGE.DBDSGRP	*dbname* = HALDB Master	*dbname* = Partition
ADDDB	Yes	Yes
DELDB	Yes	Yes
ADDMEM	No	Yes
DELMEM	No	Yes
ADDRECOV	Yes	No
DELRECOV	Yes	No

Example of Changing a Group of DBDSs

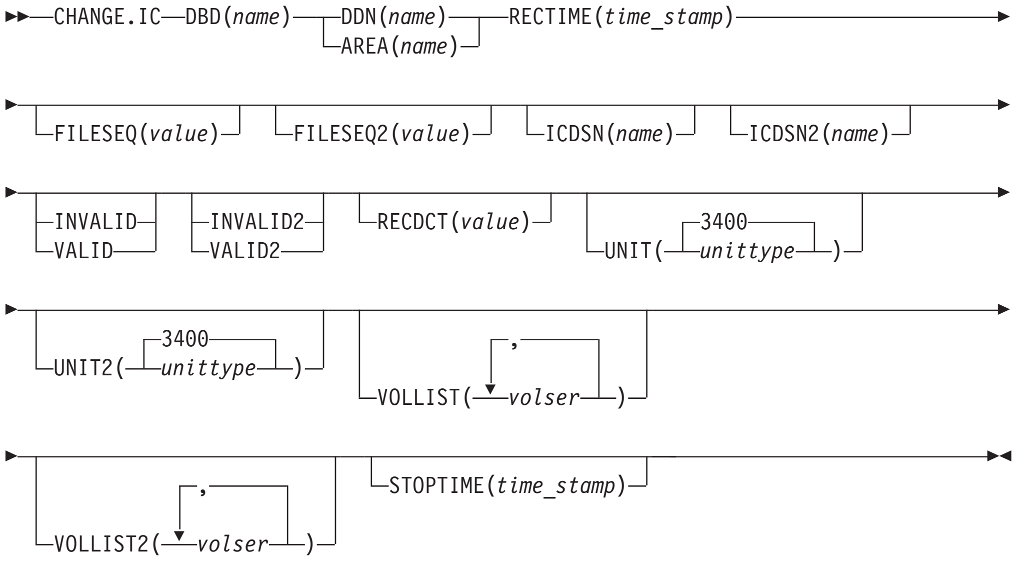
In this example, a group of DBDSs is changed.

```
//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



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.

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.

RECTIME(*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 130

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 130) 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(DBDKSDS1)  DDN(DDNKSDS1)  -
     ICDSN(IMS.DBDKSDS1.DDNKSDS1.IC.ICDSN02) -
```

```
        ICDSN2(IMS.DBDKSDS1.DDNKSDS1.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 155 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 KEYSTRNG 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 215), 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 Version 9: 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.

```
►►──CHANGE.PART──DBD(name)──PART(name)──┬─────────────────────────────┬──────────►
                                         └─KEYSTRNG(─┬─string─────┬─)─┘
                                                     └─hex_string─┘

►──┬──────────────────┬──┬────────────────┬──┬───────────────┬────────────────────►
   └─DSNPREFX(string)─┘  └─RANDOMZR(name)─┘  └─ANCHOR(value)─┘

►──┬────────────────┬──┬──────────────┬──┬─────────────┬──┬─────────────┬────────►
   └─HIBLOCK(value)─┘  └─BYTES(value)─┘  └─FBFF(value)─┘  └─FSPF(value)─┘

►──┬─────────────────────────┬──┬──────────────┬──┬───────────────────┬──────────►
   │          ┌─,─────┐      │  └─GENMAX(value)─┘  └─DEFLTJCL(member)─┘
   │          ▼       │      │
   └─BLOCKSZE(───nnnnn─┴─)───┘

►──┬──────────────┬──┬───────────────┬──┬─────────┬──┬─────────────────┬─────────►
   └─ICJCL(member)─┘  └─OICJCL(member)─┘  ├─REUSE───┤  └─RECOVJCL(member)─┘
                                          └─NOREUSE─┘
```

```
►─┬───────────────┬──┬─────────────────┬─────────────────────────────────►◄
  └─RECOVPD(value)─┘  └─RECVJCL(member)─┘
```

Or:

```
►►──CHANGE.PART──DBD(name)──┬─DISABLE─┬────────────────────────────────►◄
                            └─ENABLE──┘
```

Parameters for CHANGE.PART

DBD(*name*)
Required parameter used to identify the HALDB.

PART(*name*)
Required parameter you use to identify the partition.

KEYSTRNG(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 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 upper case. Hexadecimal values must be enclosed by single quotes and preceded by the letter X, for example: `KEYSTRNG(X'D7C1D9E3D2C5E8')`.

If no partition selection routine is specified for the HALDB, KEYSTRNG 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, KEYSTRNG 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 Version 9: Utilities Reference: Database and Transaction Manager*, ZES1-2358).

DSNPREFX(*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 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.

BLOCKSZE(*nnnnn***)**
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 of the image copies for a specified partition DBDS. Specify a numeric value from 0 to 999 that represents the number of days the image copies are to be kept in RECON. 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.

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.

```
 //CHGPART JOB
.
.
.
//IMS      DD   DSN=IMS.DBDLIB,DISP=SHR
...
//SYSIN DD *
  CHANGE.PART DBD(DB3) PART(PART3) KEYSTRNG(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─┤  ├─AVAIL───┤  └─DSN(name)─┘
                                 ├─ARSCHED──┤  └─UNAVAIL─┘
                                 ├─ARSTART──┤
                                 └─ARCHIVED─┘

►──┬────────┬──┬────────────┬────────────────────────────────────────────────────►◄
   ├─ERROR──┤  └─SSID(name)─┘
   └─NORMAL─┘
```

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.

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.

```
//CHGPRILG JOB

⋮
//SYSIN    DD   *
     CHANGE.PRILOG  OLDS(DFSOLP02) -
                    DSN(IMS.NEWLOG) SSID(IMSA)
/*
```

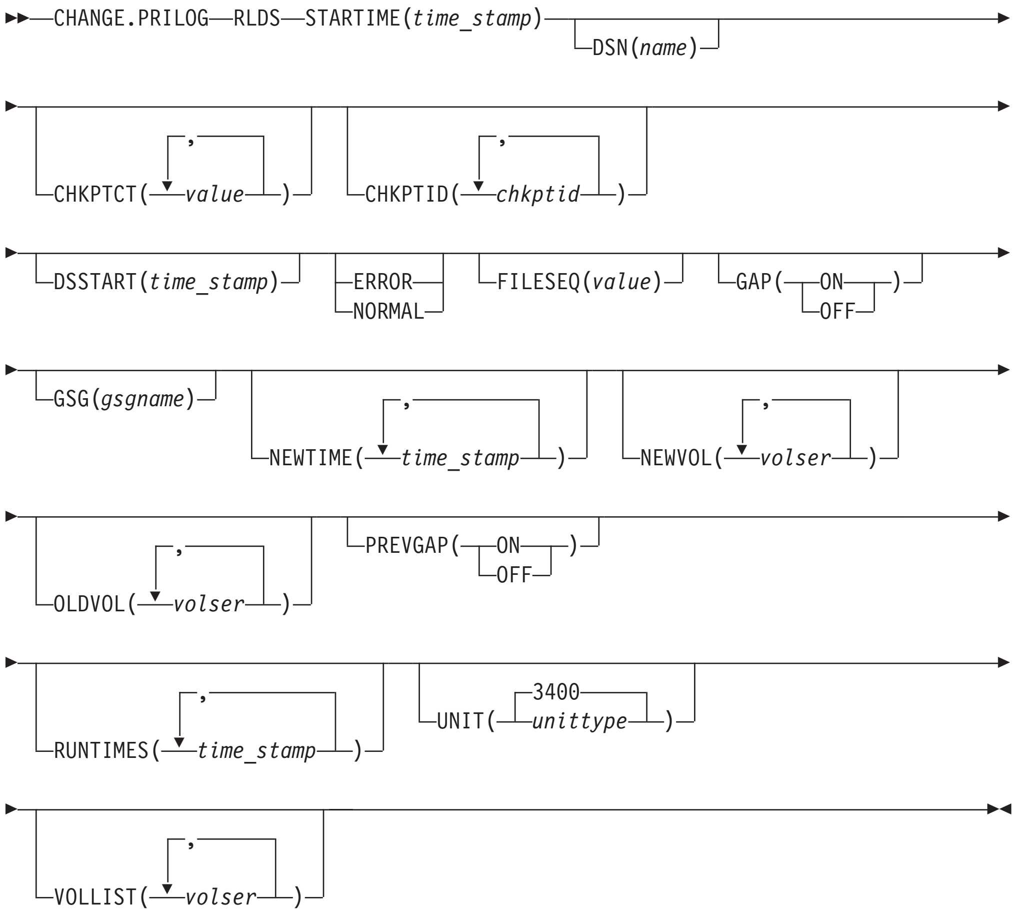
CHANGE.PRILOG (for RLDS)

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



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 130).

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 130). 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 130).

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 130).

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 that 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 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 parameter sets, NEWTIME OLDVOL NEWVOL and RUNTIMES VOLLIST, are mutually exclusive.

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 130).

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 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 174 for an explanation of how the two parameters interact. The parameter sets, NEWTIME OLDVOL NEWVOL and RUNTIMES VOLLIST, are mutually exclusive.

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
/*
```

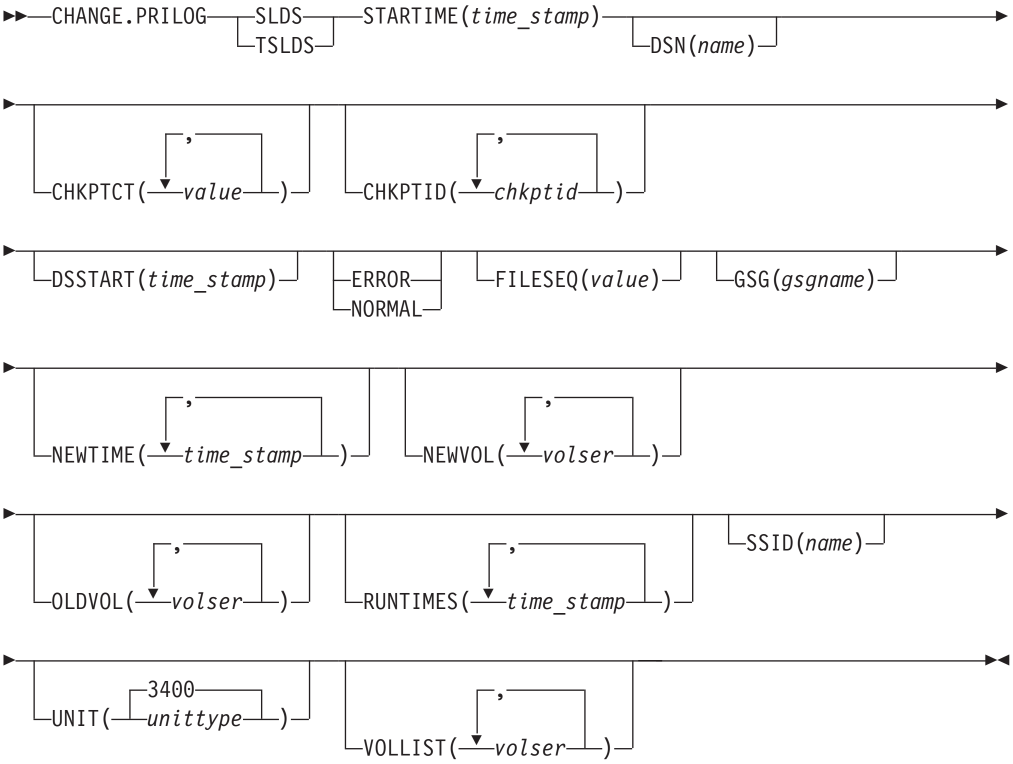
CHANGE.PRILOG (for SLDS and TSLDS)

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



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 130).

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 130). 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 130).

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 321 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 130).

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 NEWTIME OLDVOL NEWVOL to change the stop times of log volumes. If you do specify RUNTIMES, you must also specify VOLLIST. The parameter sets, NEWTIME OLDVOL NEWVOL and RUNTIMES VOLLIST, are mutually exclusive.

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 130).

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 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 321

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 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.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)
/*
```

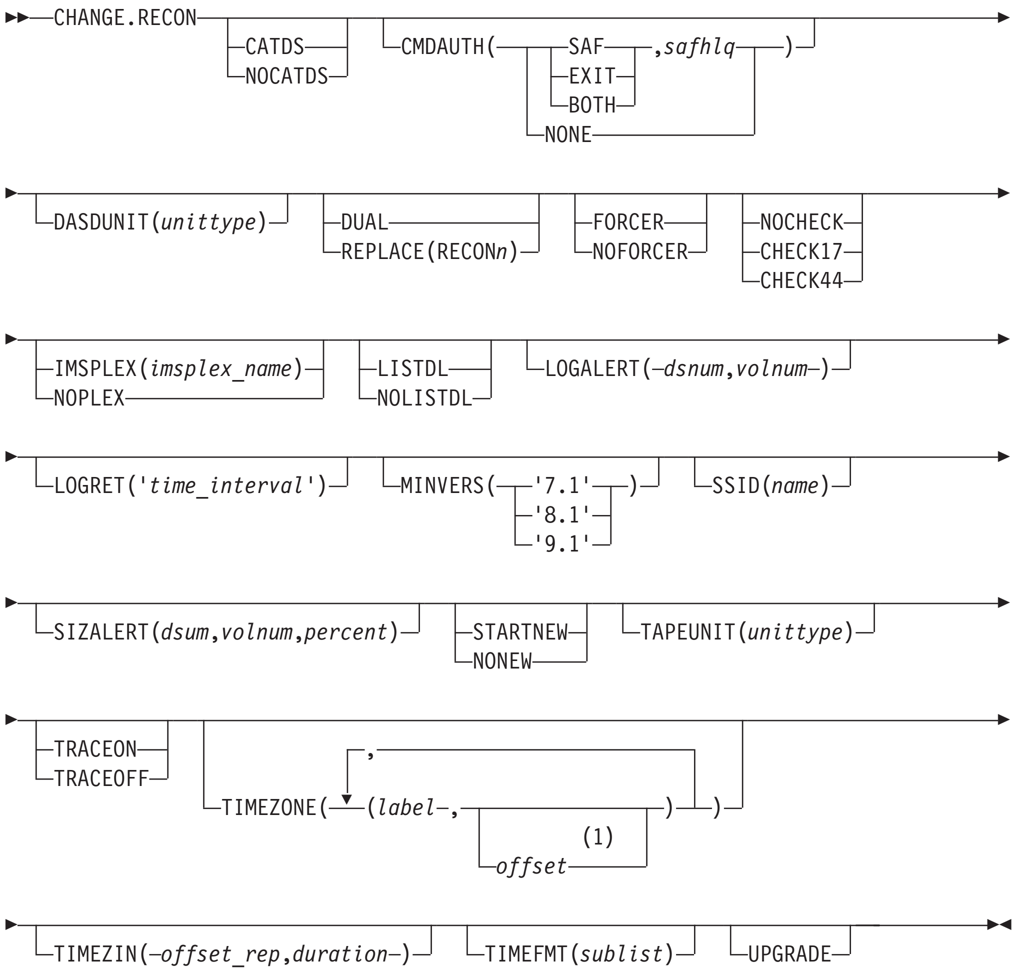
Example of Marking Primary SLDS as Normal: In this example, the first data set of a primary SLDS is marked as normal.

```
//CHGPRILG JOB
⋮
//SYSIN    DD   *
     CHANGE.PRILOG  SLDS STARTTIME(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.

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(RECON*n*)

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 57 for more information about using REPLACE.

For RECON*n*, 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.

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 Version 8 DBRC instances 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.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 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.
- 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 130 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, the 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')  x= 10 days, 12 hours, 30 minutes
```

Related Reading: See “DELETE.LOG (for RLDS and SLDS)” on page 212 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.

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 the16M 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 190 for more information about the TIMEZONE parameter.

label
An alphanumeric value of up to five characters, the first of which must be alphabetic. Lower-case characters are translated to upper case.

offset
A signed-decimal value in the form of ± [h]h[:mm] that meets the requirements of a valid time stamp offset. See “Standard Time Stamp Format” on page 130 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 190

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 130 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 132

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 `DSP0250I` 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 DSP0251I 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

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

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
 ⋮
//SYSIN    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

```
►►──CHANGE.SECLOG──OLDS(ddname)──┬─────────┬──┬───────────┬──┬────────┬──────────►
                                 ├─AVAIL───┤  └─DSN(name)─┘  ├─ERROR──┤
                                 └─UNAVAIL─┘                 └─NORMAL─┘

►──┬────────────┬──────────────────────────────────────────────────────────────────►◄
   └─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
/*
```

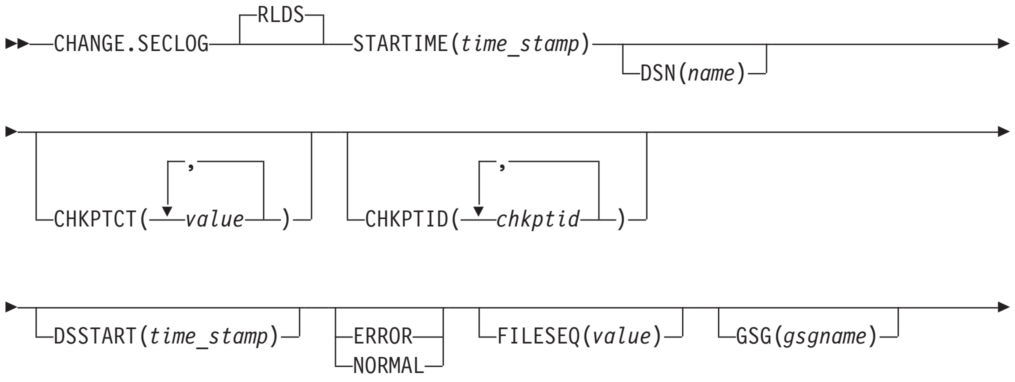
CHANGE.SECLOG (for RLDS)

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



```
►──┬────────────────────────────────┬──┬───────────────────────────┬────────────────────────►
   │             ┌─,──────────┐     │  │          ┌─,──────┐       │
   │             ▼            │     │  │          ▼        │       │
   └─NEWTIME(────time_stamp───┴─)───┘  └─NEWVOL(───volser───┴─)───┘

►──┬────────────────────────────┬──┬────────────────────────────────┬───────────────────────►
   │           ┌─,──────┐       │  │               ┌─,──────────┐   │
   │           ▼        │       │  │               ▼            │   │
   └─OLDVOL(───volser───┴─)─────┘  └─RUNTIMES(─────time_stamp───┴─)─┘

►──┬─────────────────────────────┬──┬──────────────────────────────┬─────────────────────►◄
   │        ┌─3400───────┐       │  │             ┌─,──────┐       │
   └─UNIT(──┴─unittype───┴─)─────┘  │             ▼        │       │
                                    └─VOLLIST(────volser───┴─)─────┘
```

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 130).

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 130). 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 130).

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 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 130).

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. You should use the new parameter set, NEWTIME OLDVOL NEWVOL, to change the stop times of log volumes. If you do specify RUNTIMES, you must also specify VOLLIST. The parameter sets, NEWTIME OLDVOL NEWVOL and RUNTIMES VOLLIST, are mutually exclusive.

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 130).

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 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.

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 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 174 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 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

```
►►──CHANGE.SECLOG──┬─SLDS──┬──STARTIME(time_stamp)──┬────────────┬──────────────►
                   └─TSLDS─┘                        └─DSN(name)──┘
```

```
►──┬──────────────────────────┬──┬────────────────────────────┬──┬──────────────────────────┬──────►
   │             ┌─,──────┐   │  │             ┌─,────────┐   │  └─DSSTART(time_stamp)──────┘
   │             ▼        │   │  │             ▼          │   │
   └─CHKPTCT(──value──────┴─)─┘  └─CHKPTID(──chkptid──────┴─)─┘

►──┬──────────┬──┬─────────────────┬──┬──────────────┬──┬─────────────────────────────────┬──────►
   ├─ERROR────┤  └─FILESEQ(value)──┘  └─GSG(gsgname)─┘  │              ┌─,───────────┐   │
   └─NORMAL───┘                                         │              ▼             │   │
                                                        └─NEWTIME(──time_stamp───────┴─)─┘

►──┬──────────────────────────┬──┬──────────────────────────┬──┬──────────────────────────────────┬──►
   │            ┌─,───────┐   │  │            ┌─,───────┐   │  │               ┌─,───────────┐    │
   │            ▼         │   │  │            ▼         │   │  │               ▼             │    │
   └─NEWVOL(──volser──────┴─)─┘  └─OLDVOL(──volser──────┴─)─┘  └─RUNTIMES(──time_stamp──────┴─)──┘

►──┬──────────────┬──┬─────────────────────────┬──┬───────────────────────────┬──────────────────►◄
   └─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 130).

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 130). 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 130).

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 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 321 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 130).

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. You should use the new parameter set NEWTIME OLDVOL NEWVOL to change the stop times of log volumes. If you do specify RUNTIMES, you must also specify VOLLIST. The parameter sets, NEWTIME OLDVOL NEWVOL and RUNTIMES VOLLIST, are mutually exclusive.

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 130).

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 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.PRILOG (for SLDS and TSLDS)” on page 321

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 832331243299. 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───┤  ├─LOCAL────┤  └─NORTA─┘
                                                 └─TRACKING─┘  └─NONLOCAL─┘
```

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.

```
//CHGSG    JOB
⋮
//SYSIN DD *
CHANGE.SG SGNAME() GSGNAME(GSG1) ACTIVE
/*
```

In this example, SG1 is changed to a tracking role:

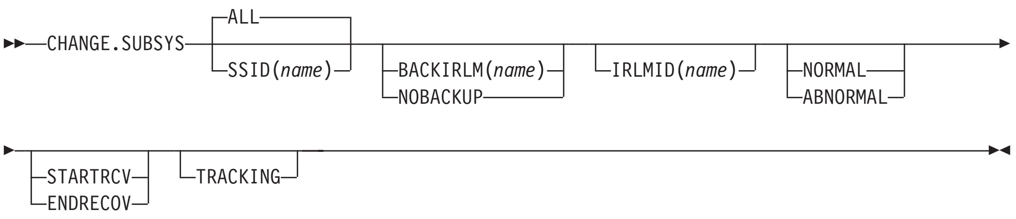
```
CHANGE.SG SGNAME(SG1) GSGNAME(GSG1) TRACKING
```

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



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.

```
//CHGSBSYS   JOB

⋮
//SYSIN      DD    *
    CHANGE.SUBSYS  IRLMID(IRLM2)  SSID(ISM34)  ABNORMAL
/*
```

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)──┬──RECTIME(time_stamp)──UDATA('string')──────────►◄
                           └─AREA(name)─┘
```

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 130).

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.

```
//CHGUIC   JOB
 ⋮
//SYSIN    DD   *
     CHANGE.UIC   DBD(DBDKSDS1)  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

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.

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.

```
//DELADS   JOB  ('LEOPARD,IMS'),'LEOPARD',REGION=880K,
⋮
//SYSIN    DD   *
     DELETE.ADS  DBD(DBD00001)  AREA(AREA0001)  -
                 ADDN(AREA0002)
/*
```

DELETE.ALLOC

Use a `DELETE.ALLOC` 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 caution, 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.ALLOC Syntax

```
►►──DELETE.ALLOC──DBD(name)──┬─DDN(name)──┬──RECTIME(time_stamp)──────────────────────►◄
                             └─AREA(name)─┘
```

Parameters for DELETE.ALLOC

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.ALLOC` 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 ALLOC record. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 130).

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.

```
//DELALLOC JOB
⋮
//SYSIN    DD   *
     DELETE.ALLOC  DBD(DBDKSDS1) DDN(DDNKSDS1)  -
       RECTIME(840231102234)
/*
```

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

```
►►──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.

```
//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

```
►►──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 130).

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 Version 9: 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.

```
//DELDB    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.

```
//DELDBDGP   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.

```
//DELDBDGP   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)──┬──RECTIME(time_stamp)──ICDSN2(name)────────────────►◄
                          └─AREA(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 130).

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)

Use this command to delete a data set entry from an OLDS record.

DELETE.LOG (for OLDS) Syntax

```
►►──DELETE.LOG──OLDS(ddname)──┬─────────┬──┬──────────┬──┬────────────┬──────────────────►◄
                              └─INTERIM─┘  └─LASTCLOS─┘  └─SSID(name)─┘
```

Parameters for DELETE.LOG (for OLDS)

OLDS(*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 *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(*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.

```
//DELLOG   JOB

⋮
//SYSIN    DD   *
     DELETE.LOG  SSID(IMSA) OLDS(DFSOLP03) -
       INTERIM
/*
```

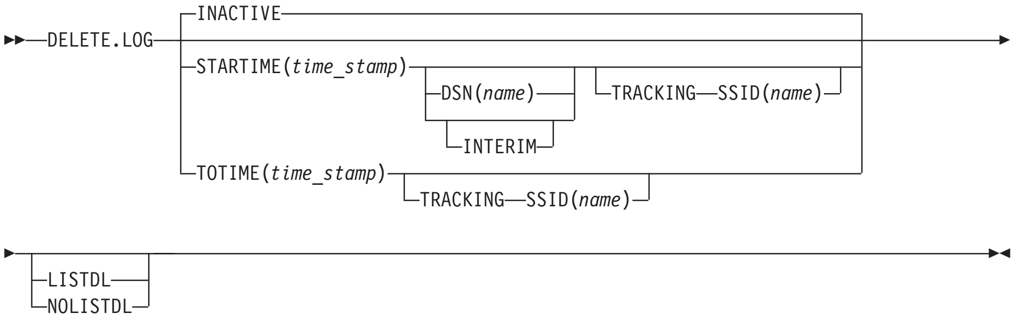
DELETE.LOG (for RLDS and SLDS)

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.
- 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



Parameters for DELETE.LOG (for RLDS and SLDS)

INACTIVE | STARTIME(*time_stamp*) | TOTIME(*time_stamp*)

Mutually exclusive, optional parameters you use to specify the records that are to be deleted.

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

Deletes an inactive 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 114 and “Getting PRILOG Compression to Work” on page 112 for more information about deleting log records and compression.

STARTIME(*time_stamp*)

Specifies the START time of the log records to be deleted. Use the 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 130). 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, TSSLD, TIPSL, and TISSL 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 TISSL 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.DPDS` 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 Version 9: 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 *
DELETE.PART  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)──┬──RECTIME(time_stamp)──────────────────────►◄
                             └─AREA(name)─┘
```

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 130).

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(DBDESDSB)     DDN(DDNESDSB)  -
        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

```
►►──DELETE.REORG──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 130).

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

```
►►──DELETE.SG──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.

```
//DELDBDGP   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 201.

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.

```
//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.

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.

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 130).

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

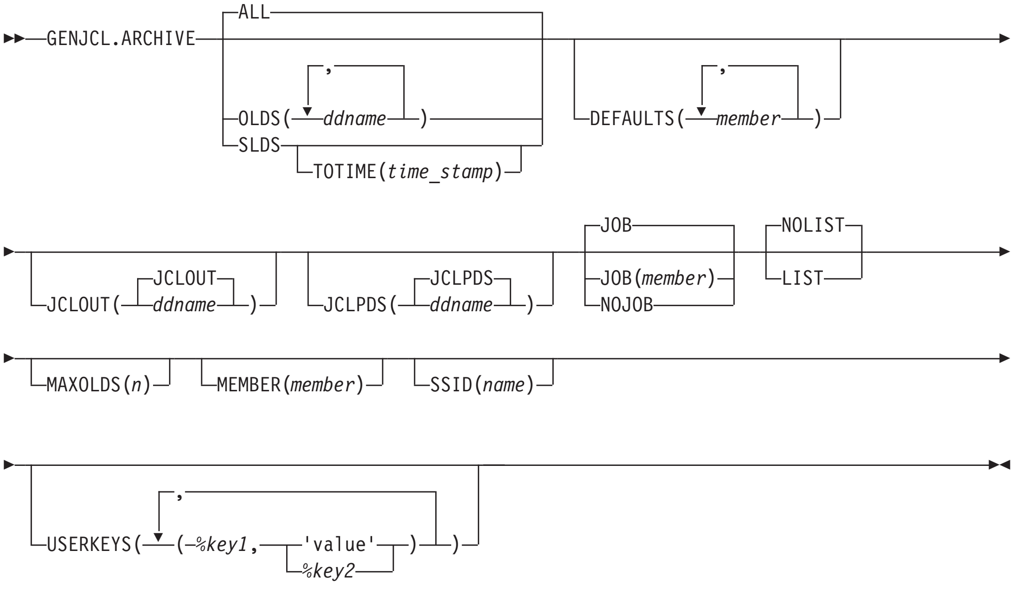
Use the GENJCL commands to generate JCL and utility control statements that run the various IMS recovery utilities.

GENJCL Commands In This Chapter

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 87

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 | *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 (PDS) as long as 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 that 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 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 87

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.

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 90

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.

```
//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 OLDSs: 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'.

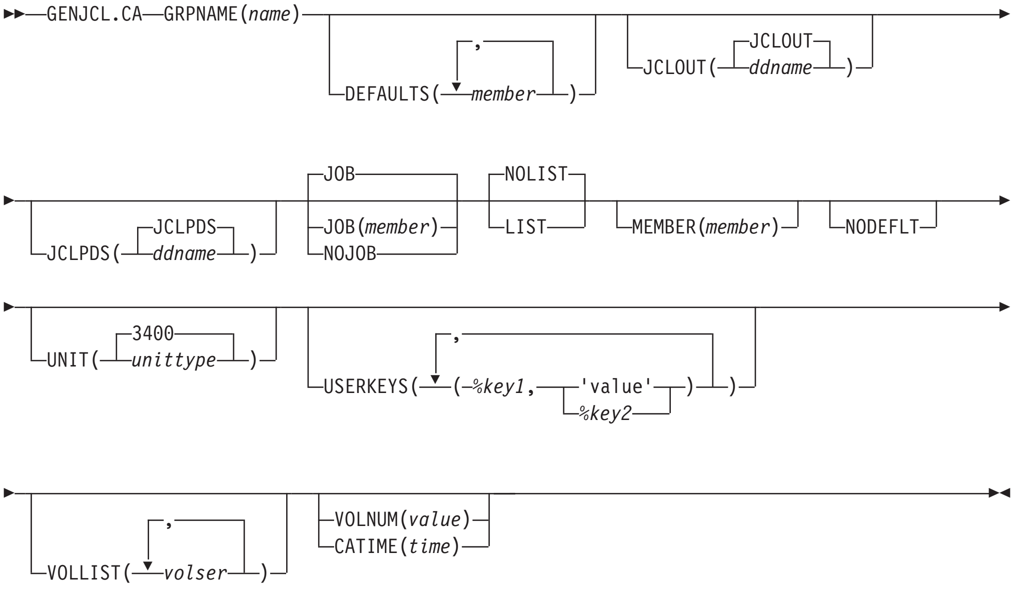
The values specified in the USERKEYS parameter for a keyword overrides the values found in the DEFAULTS member. JOB1 is a member in the JCLPDS that produces a job statement.

```
//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 87

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 87

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.

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 90

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 130).

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 CAJCLA 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 CAJCLB 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 CAGRP, 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'.

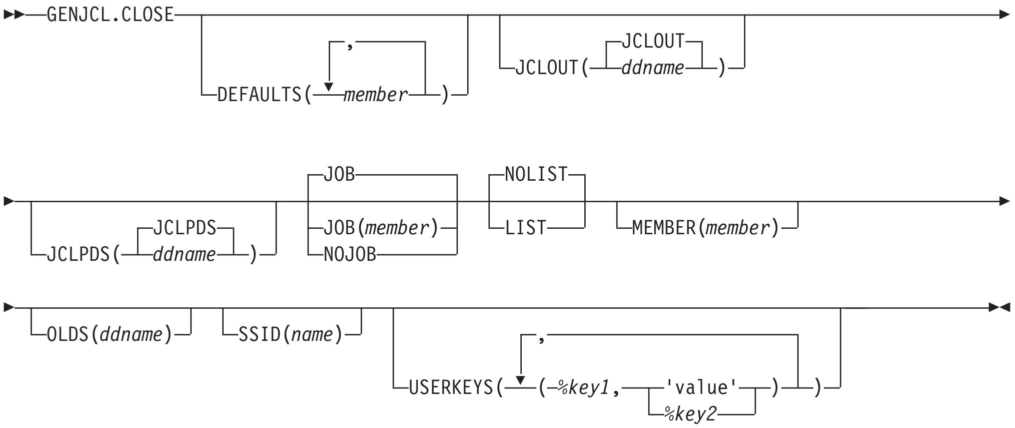
The values in the explicitly defined DEFAULTS member overrides values in the predefined DEFLTJCL member. The values specified in the USERKEYS parameter for a keyword overrides the values found in the DEFAULTS member. JCL is generated with no job statement. All volumes that have stop times less than or 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 87

GENJCL.CLOSE Syntax



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.

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 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 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 87

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.

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 90

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 87

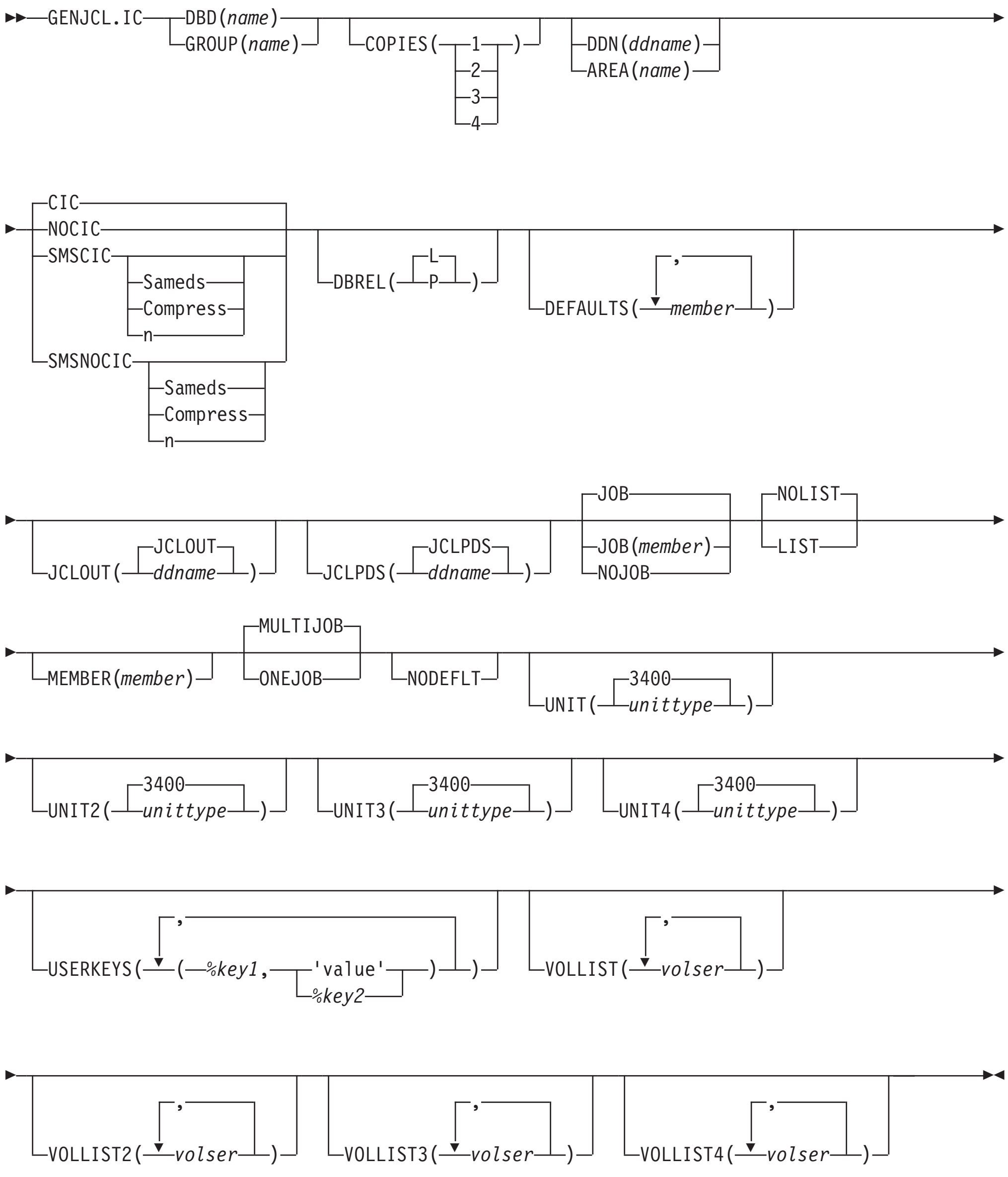
Important:

- 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.

Related Reading: See the *IMS Version 9: Utilities Reference: Database and Transaction Manager* for an overview of the Database Image Copy utility and the Database IMage Copy 2 utility.

GENJCL.IC Syntax



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*) | SMSNOCIC(Sameds | Compress | *n*)

Optional, mutually exclusive, 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*)

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(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 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 DFSMSdss OPTIMIZE parameter. *n* is a number from 1 to 4 where:

- If *n* is 1, DFSMSdss reads one track at a time.
- If *n* is 2, DFSMSdss reads two tracks at a time.
- If *n* is 3, DFSMSdss reads five tracks at a time.
- If *n* is 4, DFSMSdss 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 | *n*)

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 DFSMSdss 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 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 DFSMSdss OPTIMIZE parameter. *n* is a number from 1 to 4 where:

- If *n* is 1, DFSMSdss reads one track at a time.
- If *n* is 2, DFSMSdss reads two tracks at a time.
- If *n* is 3, DFSMSdss reads five tracks at a time.
- If *n* is 4, DFSMSdss 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*)
Applicable **only** when SMSNOCIC is also specified and it indicates when the database is to be made available for update processing. It is ignored when SMSNOCIC is not specified.

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 87 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.

UNIT(3400 | *unittype*)
UNIT2(3400 | *unittype*)
UNIT3(3400 | *unittype*)

UNIT4(3400 | *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 90

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(DBDKSDS1) DDN(DDNKSDS1)
```

The EXEC and SYSIN statements for the generated JCL are shown below:

```
⋮
 //IC1   EXEC  PGM=DFSUDMP0,REGION=nnnK,
 //               PARM='CIC,GSGNAME='
 //D1OU1  DD  DSN=...

⋮
 //SYSIN   DD *
 D1 DBDKSDS1 DDNKSDS1 D1OU1
 /*
```

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.

```
 //GENJIC2 JOB
 //JCLOUT  DD   . . .
 //PDS4    DD   . . .
 //SYSIN   DD   *
   GENJCL.IC GROUP(GROUP1) JOB MEMBER(ICJCL1) JCLPDS(PDS4) ONEJOB -
   NOCIC USERKEYS((%DEFIC,'1ST USERKEYS PARM')) NODEFLT
```

The following statements are examples of one of the EXEC statements and one of the SYSIN statements for the generated JCL:

```
⋮
 //IC1   EXEC  PGM=DFSUDMP0,REGION=nnnK,
 //               PARM=',GSGNAME='
 //D1OU1  DD  DSN=...

⋮
 //SYSIN   DD *
 D1 DBD1GRP1 DDN1GRP1 D1OU1
 /*
 //IC2   EXEC  PGM=DFSUDMP0,REGION=nnnK,
 //               PARM=',GSGNAME='

⋮
```

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.

```
 //GENJIC3  JOB
 //JCLOUT   DD   . . .
 //JCLPDS   DD   . . .

⋮
 //SYSIN    DD   *
   GENJCL.IC   DBD(DBDVSAM1) DDN(DDNVSAM1) COPIES(4) SMSCIC(1)      -
               VOLLIST(IC2001) VOLLIST2(IC2002) VOLLIST3(IC2003) -
               VOLLIST4(IC2004)
 /*
```

The EXEC and SYSIN statements for the generated JCL are shown below:

```
⋮
 //IC1   EXEC  PGM=DFSRRC00,REGION=nnnK,
 //               PARM='ULU,DFSUDMT0,,,,,,,,,,,,Y,,,,,,,,'

⋮
 //SYSIN   DD *
  4 DBDVSAM1 DDNVSAM1 D1OU1    D2OU1    D3OU1    D4OU1   S  1
 /*
```

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.

```
 //GENJIC4  JOB
 //JCLOUT   DD   . . .
 //JCLPDS   DD   . . .

⋮
 //SYSIN    DD   *
   GENJCL.IC   DBD(DBDVSAM2) DDN(DDNVSAM2) DBREL(P) SMSNOCIC
 /*
```

The EXEC and SYSIN statements for the generated JCL are shown below:

```
⋮
 //IC1   EXEC  PGM=DFSRRC00,REGION=nnnK,
 //               PARM='ULU,DFSUDMT0,,,,,,,,,,,,Y,,,,,,,,GSGN4IC2'

⋮
 //SYSIN   DD *
  1 DBDVSAM2 DDNVSAM2 DATAOUT1                           XP
 /*
```

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.

```
//GENJIC5    JOB
//JCLOUT     DD   . . .
//JCLPDS     DD   . . .
```

```
 .
 .
 .
//SYSIN      DD   *
  GENJCL.IC   DBD(DBDVSAM) SMSCIC(SAMEDS,COMPRESS) ONEJOB
/*
```

The following are the EXEC and SYSIN cards in the generated JCL:

```
:
//IC1  EXEC PGM=DFSRRC00,REGION=800K,
//             PARM='ULU,DFSUDMT0,,,,,,,,,,,,Y,,,,,,,,'
//D1OU0  DD  DSN=...
//SYSIN DD *
G  DHVNTZ02                                      S C1
 1 DHVNTZ02 HIDAM    D1OU0
S  DHVNTZ02 HIDAM2                               /*
```

GENJCL.OIC

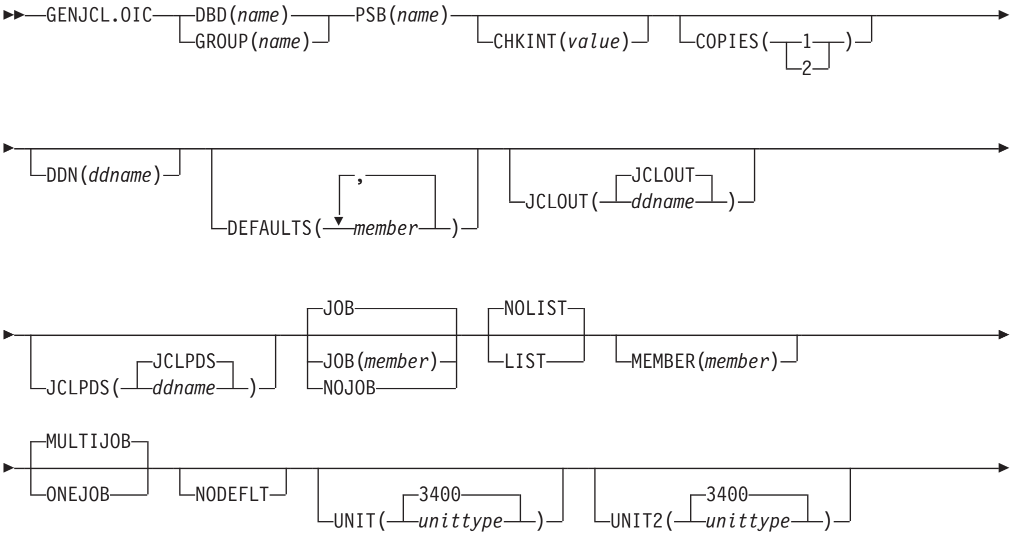
Use the GENJCL.OIC command to generate the JCL and utility control statements needed to run the Online Database Image Copy utility. For information about the IBM-supplied skeletal JCL execution member used by GENJCL.OIC, see “Generating JCL and User-Defined Output” on page 87

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



```
►──┬─────────────────────────────────────────────┬──┬───────────────────────────┬──►
   │            ┌─,──────────────────────────┐   │  │            ┌─,──────┐     │
   │            ▼                            │   │  │            ▼        │     │
   └─USERKEYS(──(─%key1,──┬─'value'─┬──)────┴─)─┘  └─VOLLIST(───volser───┴─)──┘
                          └─%key2───┘
```

```
►──┬─────────────────────────────┬───────────────────────────────────────────►◄
   │            ┌─,──────┐       │
   │            ▼        │       │
   └─VOLLIST2(───volser───┴─)───┘
```

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.OIC` 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.OIC` 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.IC` 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.OIC` command executes repeatedly, once for each DBDS of the specified database. If you specify a HALDB master name, the `GENJCL.OIC` command is performed for all data DBDSs for each partition in the HALDB master. If you specify a HALDB partition name, the `GENJCL.OIC` 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.OIC` command is performed for the identified DBDS of the partition. The `GENJCL.OIC` command fails if DDN does not identify a data DBDS in the partition.

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 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 87 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.

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 90

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.

```
 //GENJOIC  JOB
 //JCLPDS   DD   . . .
 //JCLOUT   DD   . . .

⋮
 //SYSIN    DD   *
      GENJCL.OIC     DBD(DBDKSDS1) DDN(DDNKSDS1)  -
                     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. `%DEFDBDS` 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.

```
 //GENJOIC1 JOB
 //OICOUT   DD  . . .
 //OICPDS   DD  . . .
 //SYSIN    DD  *
   GENJCL.OIC GROUP(GROUP1) JOB(JOBCARD) MEMBER(OICJCL2) -
          NODEFLT  JCLPDS(OICPDS) JCLOUT(OICOUT) PSB(PCBOIC6)      -
          USERKEYS((%DEFDBDS,'DATABASE DEFINED HERE'))     -
     DEFAULTS(DEF1,DEF2,DEF3)
```

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 87

GENJCL.RECEIVE Syntax

```
►►──GENJCL.RECEIVE──┬─DBD(name)───┬──┬───────────────────────────┬──┬───────────────┬──────►
                    └─GROUP(name)─┘  │            ┌─,──────┐     │  ├─DDN(ddname)─┤
                                     │            ▼        │     │  └─AREA(name)──┘
                                     └─DEFAULTS(───member──┴─)───┘

                                                          ┌─JOB──────────┐  ┌─NOLIST─┐
►──┬──────────────────────────┬──┬──────────────────────────┬──┼──────────────┼──┼────────┼──────►
   │          ┌─JCLOUT─┐      │  │          ┌─JCLPDS─┐      │  ├─JOB(member)──┤  └─LIST───┘
   └─JCLOUT(──┴─ddname─┴──)───┘  └─JCLPDS(──┴─ddname─┴──)───┘  └─NOJOB────────┘

►──┬──────────────────┬──┬──────────────────────────────────────────────────┬────────────►◄
   └─MEMBER(member)───┘  │             ┌─,───────────────────────────┐    │
                         │             ▼                             │    │
                         └─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 ddname 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 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 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 87 If this parameter is not specified, the default specified on the `INIT.DBDS` command is used.

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 90

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 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.RECEIVE GROUP(GROUP1) MEMBER(RCVJCL2) -
         JCLPDS(PDS) JCLOUT(OUT)       -
         USERKEYS((%DEFDBD1,'DEFINE DB1'),(%DEFDBD2,'DEFINE DB2'))   -
         DEFAULTS(DEF1,DEF2,DEF3)
```

GENJCL.RECOV

Use the GENJCL.RECOV 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.RECOV 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.RECOV 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.RECOV 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.RECOV 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.RECOV 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.

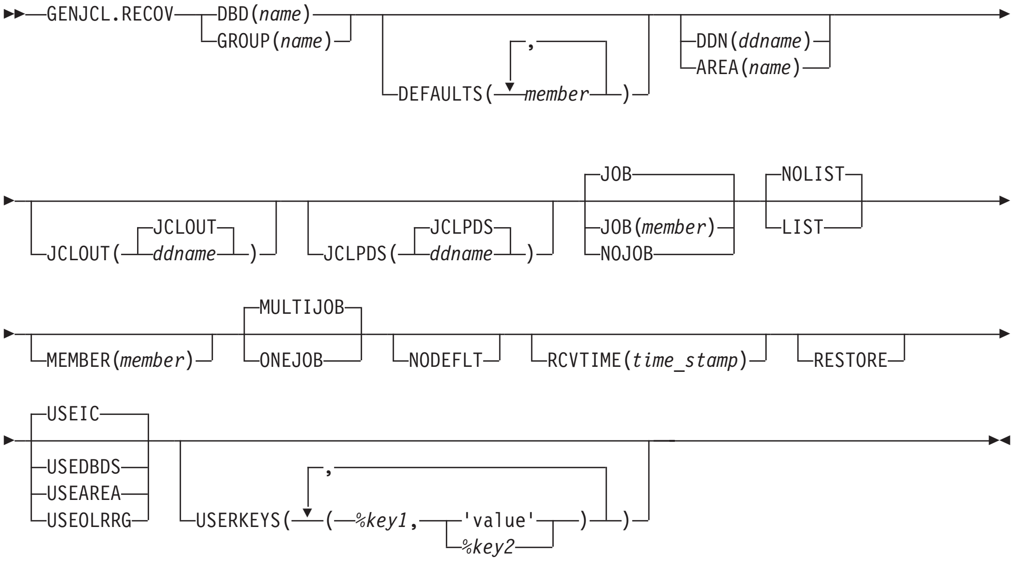
For information about the IBM-supplied skeletal JCL execution member used by GENJCL.RECOV, see "Generating JCL and User-Defined Output" on page 87

Use the RESTORE parameter to recover DBDSs that were designated as nonrecoverable.

Important: For HALDB partitions, the GENJCL.RECOV command treats ILDS and index data sets differently than data DBDSs. The GENJCL.RECOV 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.RECOV command fails.

Restriction: The GENJCL.RECOV 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 465 for information about IBM-supplied sample JCL that you can use.

GENJCL.RECOV Syntax



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
Specifies 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

Specifies 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 DEDB 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 87 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 97 for more information on full recovery.
- See *IMS Version 9: Operations Guide* for more information about time-stamp recoveries.

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.

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 90

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.

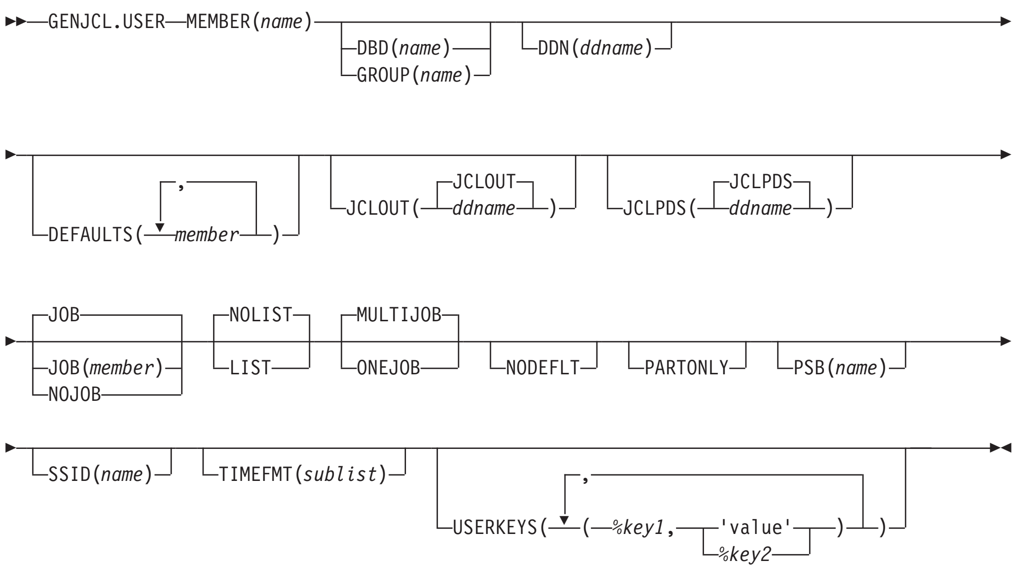
Skeletal member JOBJCL produces a job statement for each member of the group. `%DEFDBD1` and `%DEFDBD2` are user-defined values in member RECJCL2 which resolve to 'DEFINE DB1' and 'DEFINE DB2'. Default members DEF1, DEF2, and DEF3 are used to resolve user-defined keywords in RECJCL2. The default member 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((%DEFDBD1,'DEFINE DB1'),(%DEFDBD2,'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 87

GENJCL.USER Syntax



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 %mdbname 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.

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 132
- See “TIMEFMT Subparameter Order of Precedence” on page 134 for information on the precedence of the subparameters.
- See “Standard Time Stamp Format” on page 130 for examples of the different output forms.

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.

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 90

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.

```
//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:

```
/        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.DBDS COMMAND *
//* (DDNAME) SHOULD BE SUBSTITUTED IN LIST.DBDS 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

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 to 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

```
                                                           ┌─UNAVAIL─┐
►►──INIT.ADS──ADDN(name)──ADSN(name)──AREA(name)──DBD(name)──┼─────────┼──────────────────────►◄
                                                           └─AVAIL───┘
```

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 | UNAVAIL
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.

UNAVAIL
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.

```
//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.CAGRP` command. You can create change accumulation records in RECON up to the number specified in the GRPMAX parameter of the `INIT.CAGRP` command that you used to define the CA group.

INIT.CA Syntax

```
                                                   ┌─,──────┐
►►──INIT.CA──CADSN(name)──GRPNAME(name)──VOLLIST(──▼─volser─┴──)──┬─────────────────────────┬──►
                                                                   │          ┌─1─────┐      │
                                                                   └─FILESEQ(─┴─value─┴─)───┘
```

```
►─┬──────────────────────────┬──────────────────────────────────────────►◄
  │         ┌─3400─────┐     │
  └─UNIT(───┴─unittype─┴──)──┘
```

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.

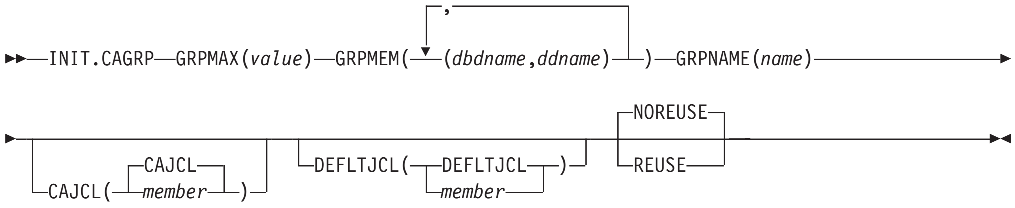
```
//INITCA   JOB
⋮
//SYSIN    DD   *
     INIT.CA  GRPNAME(CAGRP1)  -
              CADSN(IMS.CAGRP1.CA.CA001)  -
              VOLLIST(VOL001)      FILESEQ(4)
/*
```

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



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. *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. *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.

```
 //INITCAGP 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))
 /*
```

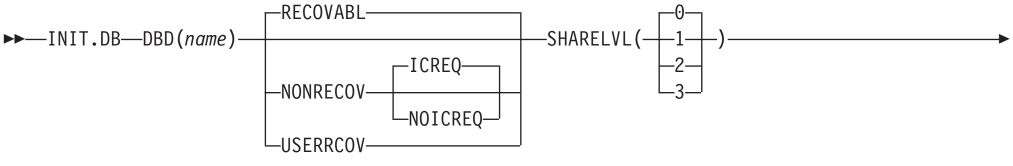
INIT.DB

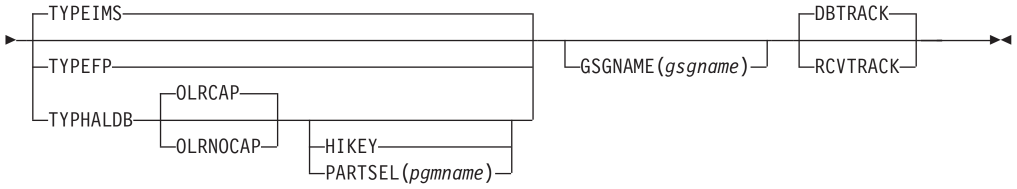
Use an INIT.DB command to register a database with DBRC and specify the level of database sharing. A database must be registered with DBRC before you can initialize a new DBDS, HALDB partition, or DEDB area with the INIT.DBDS command or INIT.PART.

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





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(*gsgname*)

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 19

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 Version 9: Administration Guide: System*.

TYPEFP | TYPEIMS | TYPEHALDB

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).

TYPEHALDB

Specifies that the database is a DL/I database (HALDB).

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 KEYSTRNG 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(pgmname)
Specifies the name of the Partition Selection Exit program to be used for the HALDB. The *pgmname* 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 Recovery Control utility with a ddname of IMS.

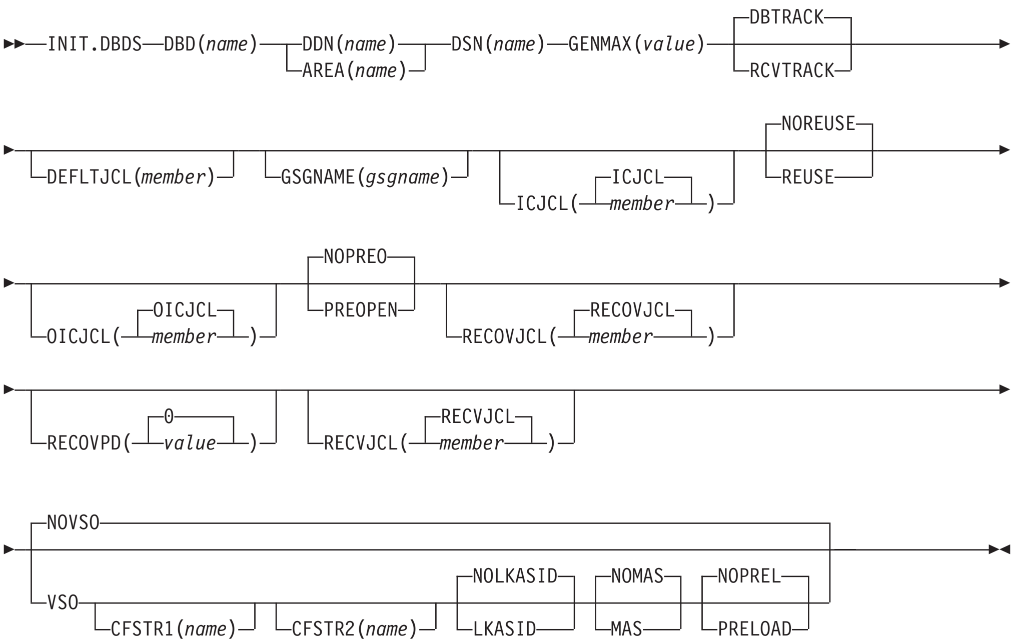
The `INIT.DBDS` command fails if you issue it while the database is in use.

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 155 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



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.OIC`, and `GENJCL.RECOV` commands in order to resolve keywords you have defined.

GSGNAME(*gsgname*)

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.OIC` 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(OICJCL | *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.OIC` 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 /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 pre-opened the next time the control region is started or a `/START AREA` command is processed. You cannot specify NOPREO with PRELOAD.

RECOVJCL(RECOVJCL | *member*)

Optional parameter you use to specify the name of a member of a partitioned data set of skeletal JCL. When you issue the `GENJCL.RECOV` command, DBRC uses this member to generate the JCL that runs the Database Recovery utility for the DBDS or area being identified.

RECOVPD(0 | *value*)

Optional parameter you use to specify the recovery period of the image copies for a specified DBDS or DEDB area.

The recovery period is 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 Version 9: 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 an 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 is SHARELVL(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 Version 9: 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 it is accessed only by an IMS system. The image copy data sets can be reused, and the maximum number of images that DBRC is to maintain for the image copy is 2. 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

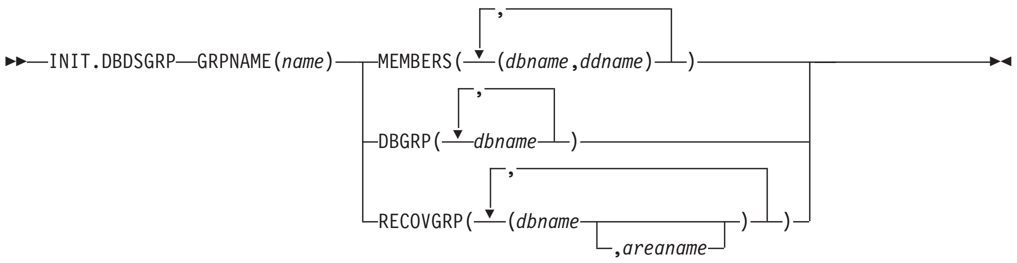
Use an INIT.DBDSGRP command to define a group of these types:

- 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 Syntax



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. Partitioned databases 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.DBDSGRP` command only as defined in Table 12.

Table 12. Specifying HALDB Master or Partition Database Names for the INIT.DBDSGRP Command

INIT.DBDSGRP	*name* = HALDB Master	*name* = Partition
MEMBERS	No	Yes

Table 12. Specifying HALDB Master or Partition Database Names for the INIT.DBDSGRP Command (continued)

INIT.DBDSGRP	*name* = **HALDB Master**	*name* = **Partition**
DBGRP	Yes	Yes
RECOVGRP	Yes	No

Example of Creating a Group of DBDSs

In this example, a group of DBDSs is defined.

```
 //INITDBGRP   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(gsgname)──┬─────────────────┬──────────────────────────────►◄
                                └─SEQNUM(number)─┘
```

Parameters for INIT.GSG

GSGNAME(*gsgname***)**
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

```
►►──INIT.IC──DBD(name)──┬─DDN(name)──┬──ICDSN(name)──┬──────────────────────────┬──────────────────►
                        └─AREA(name)─┘               │          ┌─1─────┐       │
                                                     └─FILESEQ(─┴─value─┴─)─────┘

►──┬───────────────────────────┬──┬─────────────────┬──┬───────────────────────────┬─────────────────►
   │           ┌─1─────┐       │  └─ICDSN2(name)─┘  │        ┌─3400─────┐       │
   └─FILESEQ2(─┴─value─┴─)─────┘                    └─UNIT(─┴─unittype─┴─)─────┘

►──┬───────────────────────────┬──┬─────────────────────────┬──┬──────────────────────────┬──────►◄
   │        ┌─3400─────┐       │  │          ┌─,──────┐     │  │           ┌─,──────┐     │
   └─UNIT2(─┴─unittype─┴─)─────┘  └─VOLLIST(─▼─volser─┴─)──┘  └─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.

ICDSN2(*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 ICDSN2 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.

```
//INITIC   JOB
⋮
//SYSIN    DD   *
     INIT.IC     DBD(DB1)  DDN(DD1)  -
                 ICDSN(IMS.*.ICDSN2)  -
                 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 Version 9: Administration Guide: Database Manager*, ZES1-2330 for information on designing databases, especially the following chapters:

- ″Designing Databases″
- ″Naming Conventions″

INIT.PART Syntax

```
►►──INIT.PART──DBD(name)──PART(name)──┬──────────────────────────────┬──────────────►
                                      └─KEYSTRNG──┬─string─────┬─────┘
                                                  └─hex_string─┘

►─DSNPREFX(string)──┬────────────────┬──┬──────────────┬──┬───────────────┬─────────►
                    └─RANDOMZR(name)─┘  └─ANCHOR(value)─┘  └─HIBLOCK(value)─┘

►──┬────────────┬──┬─────────────────────┬──┬─────────────────────┬─────────────────►
   └─BYTES(value)─┘  │      ┌─0─────┐      │  │      ┌─0─────┐      │
                   └─FBFF(─┴─value─┴─)─┘  └─FSPF(─┴─value─┴─)─┘

►──┬───────────────────────────────────┬──┬────────────────────────┬────────────────►
   │                ┌─,───────┐        │  │          ┌─2─────┐       │
   │                ▼         │        │  └─GENMAX(─┴─value─┴─)─┘
   └─BLOCKSZE──(───┬─4096──┬─┴──)──┘
                   └─value─┘

►──┬─────────────────┬──┬──────────────────────┬──┬────────────────────────┬────────►
   └─DEFLTJCL(member)─┘  │       ┌─ICJCL──┐      │  │        ┌─OICJCL─┐      │
                       └─ICJCL(─┴─member─┴─)─┘  └─OICJCL(─┴─member─┴─)─┘

   ┌─NOREUSE─┐
►──┼─────────┼──┬─────────────────────────────┬──┬──────────────────────┬───────────►
   └─REUSE───┘  │          ┌─RECOVJCL─┐         │  │          ┌─0─────┐    │
                └─RECOVJCL(─┴─member───┴─)─┘  └─RECOVPD(─┴─value─┴─)─┘

►──┬──────────────────────────┬─────────────────────────────────────────────────────►◄
   │         ┌─RECVJCL─┐        │
   └─RECVJCL(─┴─member──┴─)─┘
```

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.

KEYSTRNG (*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: KEYSTRNG(X’D7C1D9E3D2C5E8’).

If no partition selection routine was specified in the HALDB master definition, KEYSTRNG 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, KEYSTRNG 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.

DSNPREFX (*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 RANDOMIZR 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 Version 9: Utilities Reference: Database and Transaction Manager* 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
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.

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.

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 (_0_| *value***)**
Optional parameter you use to specify the recovery period of the image copies for a specified partition DBDS. Specify a numeric value from 0 to 999 that represents the number of days the image copies are to be kept in RECON. 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 KEYSTRNG 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.DBDLIB,DISP=SHR
```

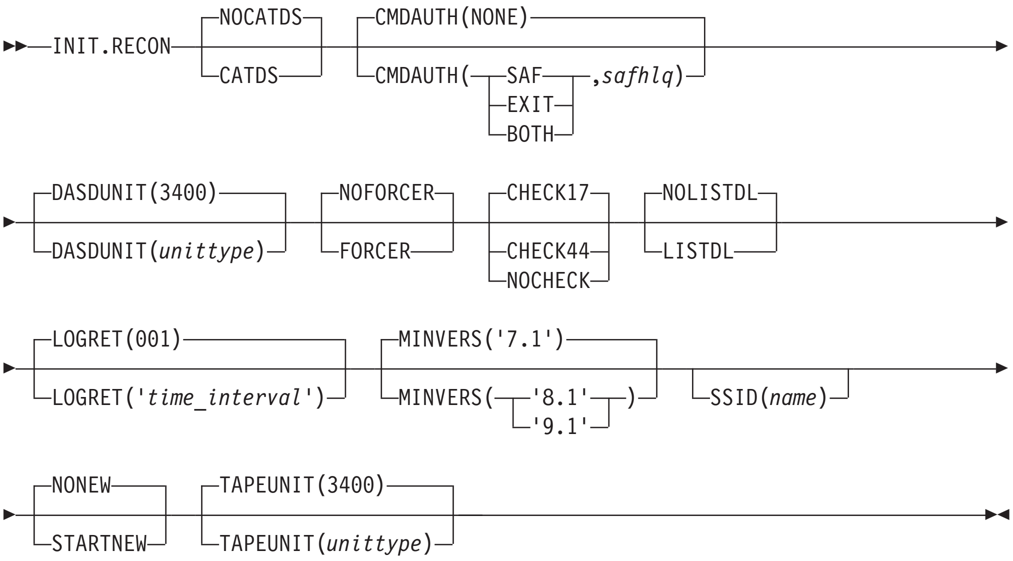
```
//SYSIN    DD  *
  INIT.PART DBD(DBHDOK01) PART(PDHDOKE) KEYSTRNG('ABCDEFGHIJKL--
MNOPQRSTUVWXYZ0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJ-
KLMNOPQRSTUVWXYZ0123456789- ABCD--
EFGHIJKLMNOPQRSTUVWXYZ0123456789') -
  DSNPREFX(TESTSYS1.PROJECT) -
  BLOCKSZE(32000,,4000)
/*
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
```

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



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 212 for more information on deleting inactive logs.
- See "Standard Time Stamp Format" on page 130 for more information on time stamps.

LISTDL | NOLISTDL

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.

NOLISTDL

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.

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.

```
//INITRCON JOB

⋮
//RECON1   DD  DSN=RECON7,DISP=SHR
//RECON2   DD  DSN=RECON8,DISP=SHR
//SYSIN    DD   *
     INIT.RECON  NOCHECK SSID(IMSB) 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

```
                                                          ┌─NONLOCAL─┐
►►──INIT.SG──GSGNAME(gsgname)──SGNAME(sgname)──┬─ACTIVE───┬──┼──────────┼──────────────────►◄
                                               └─TRACKING─┘  └─LOCAL────┘
```

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) SGNAME(STLSITE1) ACTIVE
INIT.SG  GSGNAME(IMSGSG1) SGNAME(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

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 of RECONs,” on page 469

LIST.BKOUT Syntax

```
                  ┌─ALL────────┐
►►──LIST.BKOUT────┼────────────┼──┬──────────────────┬──────────────────────────────────►◄
                  └─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 132 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 130 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 of RECONs," on page 469

LIST.CAGRP Syntax

```
                  ┌─ALL───────────┐
►►──LIST.CAGRP────┼───────────────┼──┬──────────────────┬──┬─────────┬──────────────►◄
                  └─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 132 for a description of the TIMEFMT parameter sublist format.
- See "Standard Time Stamp Format" on page 130 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   *
     LIST.CAGRP 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.

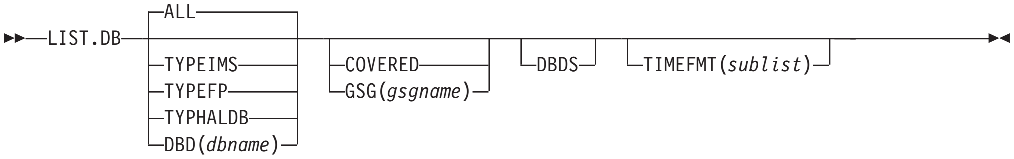
For the format of the records listed by this command, see the Appendix B, “Sample Listings of RECONs,” on page 469

The `LIST.DB` command displays the recoverable or nonrecoverable status of the database.

Related Reading:

- See “HALDB Records” on page 73 for detailed information about how HALDBs are represented in the RECON data set.
- See *IMS Version 9: 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 132 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 130 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 of RECONs,” on page 469

LIST.DBDS Syntax

```
►►──LIST.DBDS──┬─DBD(name)───┬──┬───────────┬──┬─────────────────┬──────────►◄
               └─GROUP(name)─┘  ├─DDN(name)─┤  └─TIMEFMT(sublist)─┘
                                └─AREA(name)┘
```

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 132 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 130 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   *
     LIST.DBDS DBD(FPEDBD02) AREA(AREA01)
/*
```

LIST.DBDSGRP

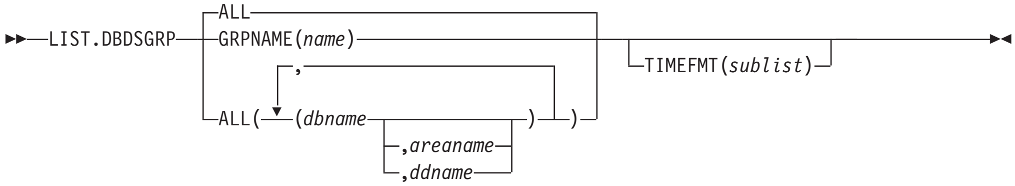
Use a LIST.DBDSGRP command to display a list of any of the following:

- 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 162

LIST.DBDSGRP Syntax



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 DBDS, DB, 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 DBDS, DB, 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 132 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 130 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

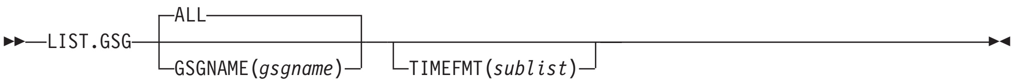
In this example, the members of a specified DBDS group are displayed.

```
//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 Appendix B, “Sample Listings of RECONs,” 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 132 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 130 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:

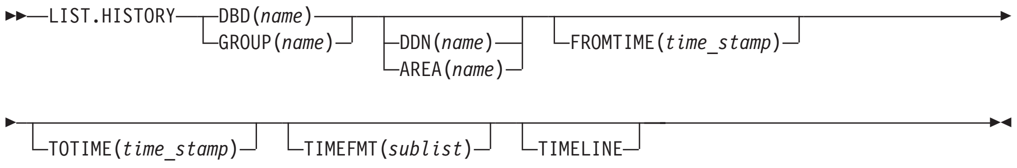
- The database record listing.
- The DBDS record listing. (For a DEDB area, the area authorization and area recovery records are combined to form a DBDS record listing.)
- The following records (if present), listed in time sequence order under each DBDS or area:
  - ALLOC records.
  - IMAGE records.
  - 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/or TOTIME 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 130). 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.

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 130). 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 132 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 130 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.

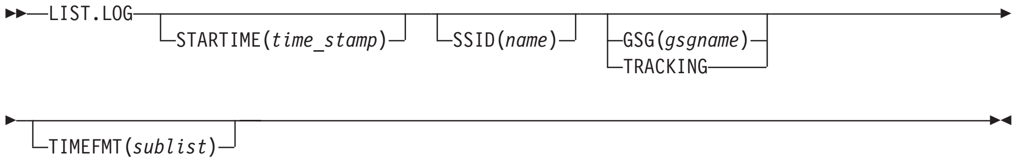
```
//LISTHIST    JOB

⋮
//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



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 132 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 130 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, SECSLD, 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)

Each command form is followed by a list of the records it displays.

LIST.LOG (for a Category of Records) Syntax

```
                ┌─ALL─┐
►►──LIST.LOG────┴─────┴──┬──────────────────────────┬─────────────────────────────►◄
                         └─┤ See "Optional Filters" ├─┘
```

- PRILOG
- LOGALL
- SECLOG
- PRISLD
- SECSLD
- PRIOLD
- SECOLD

```
              ┌─ALL─┐
►►──LIST.LOG──┴─────┴──INTERIM──┬──────────────────────────┬──────────────►◄
                                └─┤ See "Optional Filters" ├┘
```

- IPRI
- ISEC
- IPRISL
- ISECSL
- IPRIOL
- ISECOL

```
              ┌─ALL─┐
►►──LIST.LOG──┴─────┴──TRACKING──┬──────────────────────────┬─────────────►◄
                                 └─┤ See "Optional Filters" ├┘
```

- PRITSLDS
- SECTSLDS

```
              ┌─ALL─┐
►►──LIST.LOG──┴─────┴──TRACKING──INTERIM──┬───────────────────────┬───────►◄
                                          └─┤ See "Optoinal Filter" ├┘
```

- IPRITSLD
- ISECTSLD

```
►►──LIST.LOG──ALLOLDS──┬──────────────────────────┬───────────────────────►◄
                       └─┤ See "Optional Filters" ├┘
```

- PRIOLD
- SECOLD

```
►►──LIST.LOG──ALLOLDS──INTERIM──┬──────────────────────────┬──────────────►◄
                                └─┤ See "Optional Filters" ├┘
```

- IPRIOL
- ISECOL

```
►►──LIST.LOG──OLDS(ddname)──┬────────────┬──┬──────────────────────────┬──►◄
                            └─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:

```
▸▸──┬──────────────────────────┬──┬────────────────────────┬──┬────────────┬──┬──────────────┬──▸
    └─FROMTIME(time_stamp)─┘  └─TOTIME(time_stamp)─┘  └─SSID(name)─┘  └─GSG(gsgname)─┘

▸──┬──────┬──┬───────┬──┬────────┬──────────────────────────────────────────────────────────▸◂
   └─OPEN─┘  └─ERROR─┘  └─UNARCH─┘
```

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 130).

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 130).

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.

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 132 for a description of the TIMEFMT parameter sublist format.
- See "Standard Time Stamp Format" on page 130 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 SSID(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:

RECON status	Meaning
COPY1	PRIMARY ACTIVE RECON
COPY2	SECONDARY ACTIVE RECON
SPARE	AVAILABLE RECON

UNAVAILABLE	UNAVAILABLE RECON
DISCARDED	UNUSABLE RECON

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 132 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 130 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 of RECONs,” on page 469

Example of Displaying RECON Header and Status Information: In this example, only RECON header status information is displayed.

```
//LISTRCON JOB
⋮
//SYSIN    DD   *
   LIST.RECON STATUS
/*
```

Only the first segment of output shown in the Appendix B, “Sample Listings of RECONs,” on page 469 is produced in this case.

Related Reading: See Chapter 4, “Initializing and Maintaining the RECON Data Set,” on page 57 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 of RECONs,” on page 469

LIST.SUBSYS Syntax

```
                ┌─ONLINE─────┐
►►──LIST.SUBSYS─┤            ├─┬──────────────────┬─┬──────────┬──────────────────►◄
                ├─SSID(name)─┤ └─TIMEFMT(sublist)─┘ └─TRACKING─┘
                ├─ALL────────┤
                └─BATCH──────┘
```

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 132 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 130 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

NOTIFY.ALLOC

Use a NOTIFY.ALLOC 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 NOTIFY.ALLOC 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.

NOTIFY.ALLOC Syntax

```
►►──NOTIFY.ALLOC──ALLTIME(time_stamp)──DBD(name)──┬─DDN(name)──┬──────────────────────►
                                                  └─AREA(name)─┘

►──┬─DEALTIME(time_stamp)──┬──┬──────────────┬──┬─────────────┬──────────────────────►◄
   └─STARTIME(time_stamp)──┘  └─DSSN(value)──┘  └─USID(value)─┘
```

Parameters for NOTIFY.ALLOC

ALLTIME(*time_stamp*)

Required parameter you use to specify the time stamp of the allocation of the 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 130).

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 130).

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.

```
//NFYALLOC 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)──────────────────►

►─┬──────────────────┬─┬──────────────────┬───────────────────────────────────────────►◄
  │       ┌─,──┐     │ │       ┌─,──┐     │
  │       ▼    │     │ │       ▼    │     │
  └─DBD(──name─┴─)─┘ └─BKO(──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 130).

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.BACKOUT` 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(E2E8E2F340404040000000000600000003)
                   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──┴─)──┬────────────────────────────┬──────────────────────────────────────►
                         │              ┌─1─────┐     │
                         └─FILESEQ(──┴─value─┴──)─┘

►─┬──────────────────────────────────────────────────────────────────┬─┬─────────────────────────┬──►
  │          ┌─,─────────────────────────────────────────────┐      │ │        ┌─3400──────┐    │
  │          ▼                         ┌─NO──┐   ┌─COMPLETE───┐   │      │ └─UNIT(──┴─unittype─┴──)─┘
  └─PURGLIST(──(─time_stamp,──┴─YES─┴─,─┴─INCOMPLETE─┴──)──┴──)─┘

  ┌─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 130).

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 130).

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 130). 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   *
     NOTIFY.CA   GRPNAME(CAGRP2) -
                 STOPTIME(840240202020) -
                 RUNTIME(840250305029) CADSN(CADSN06) -
                 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)──┬──ICDSN(name)──RUNTIME(time_stamp)──────────────►
                          └─AREA(name)─┘

                                                                         ┌─BATCH────┐
►──┬─────────────────────────┬──┬──────────────────────────┬──┬─────────────────┬──┼──────────┼──►
   │           ┌─1─────┐     │  │            ┌─1─────┐     │  └─ICDSN2(name)────┘  ├─ONLINE───┤
   └─FILESEQ(──┴─value─┴──)──┘  └─FILESEQ2(──┴─value─┴──)──┘                       ├─CIC──────┤
                                                                                   ├─SMSCIC───┤
                                                                                   └─SMSNOCIC─┘

►──┬─────────────────┬──┬─────────────────────────┬──┬────────────────────────┬───────────────►
   └─RECDCT(value)───┘  └─STOPTIME(time_stamp)────┘  │       ┌─3400─────┐     │
                                                     └─UNIT(─┴─unittype─┴─)───┘

►──┬─────────────────────────┬──┬─────────────────────────┬──┬──────────────────────────┬─────►
   │        ┌─3400─────┐     │  │            ┌─,──────┐   │  │             ┌─,──────┐   │
   └─UNIT2(─┴─unittype─┴─)───┘  │            ▼        │   │  │             ▼        │   │
                                └─VOLLIST(───┴─volser─┴─)─┘  └─VOLLIST2(───┴─volser─┴─)─┘

►──┬───────────────┬───────────────────────────────────────────────────────────────────────►◄
   └─USID(value)───┘
```

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 ddname 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 130).

FILESEQ(1 | *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(1 | *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 | SMSNOCIC
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.

SMSNOCIC
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 SMSCIC is specified. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 130).

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(8520002020)
                 STOPTIME(8520004040)
                 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(gsgname)─┘  └─INTERIM─┘  └─LASTREC(number)─┘
```

```
►─┬──────────────────┬──┬──────────────────┬──┬──────────────┬──────────────────►◄
  └─LOGTOKEN(number)─┘  └─NXTOLDS(ddname)─┘  └─SSID(name)─┘
```

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 130).

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 13 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 13. Parameters of NOTIFY.PRILOG (for OLDS) Command for Open, Switch, and Close

Type of Log Entry	Required Keywords
OLDS Open	STARTIME, DSN, FIRSTREC
OLDS Switch	STARTIME, DSN, FIRSTREC, NXTOLDS
OLDS Close	LASTREC, STARTIME, RUNTIME

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 130

See Table 13 for a description of the use of this parameter with other parameters in the `NOTIFY.PRILOG` 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 PRILOG, it 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 that is 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')`.
- 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 OLDS.

GSG is required if LOGTOKEN is specified.

INTERIM
Optional parameter you use to specify that an interim log data set record is to be created.

Before you specify `NOTIFY.PRILOG INTERIM`, a corresponding primary log record must exist.

LASTREC(*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 `CLOSE` command. It is optional for the `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 `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: `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.

LOGTOKEN(*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.

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.OLDSP13) OLDS(DFSOLP13) -
                     FIRSTREC(001) -
                     SSID(IMSA)
/*
```

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.

```
NOTIFY.PRILOG SSID(IMSA) STARTIME(812171212120) OLDS(DFSOLP01) -
       DSN(IMS.OLDP01) LASTREC(4999)
NOTIFY.PRILOG SSID(IMSA) STARTIME(812181212120) OLDS(DFSOLP01) -
```

```
      DSN(IMS.OLDP02) NXTOLDS(DFSOLP02)
NOTIFY.PRILOG SSID(IMSA) STARTIME(812181212120) OLDS(DFSOLP02) -
      RUNTIME(932191010101)
/*
```

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)                                          LASTREC(4999)+
NXTOLDS(OLDS002) DSN(IMS.OLDS.A02) STARTIME(930140930000) FIRSTREC(5000)

NOTIFY.PRILOG SSID(IMSA)                                              +
   OLDS(OLDS002)                  STARTIME(930140930000)              +
                                   RUNTIME(930141030000)  LASTREC(9999)
```

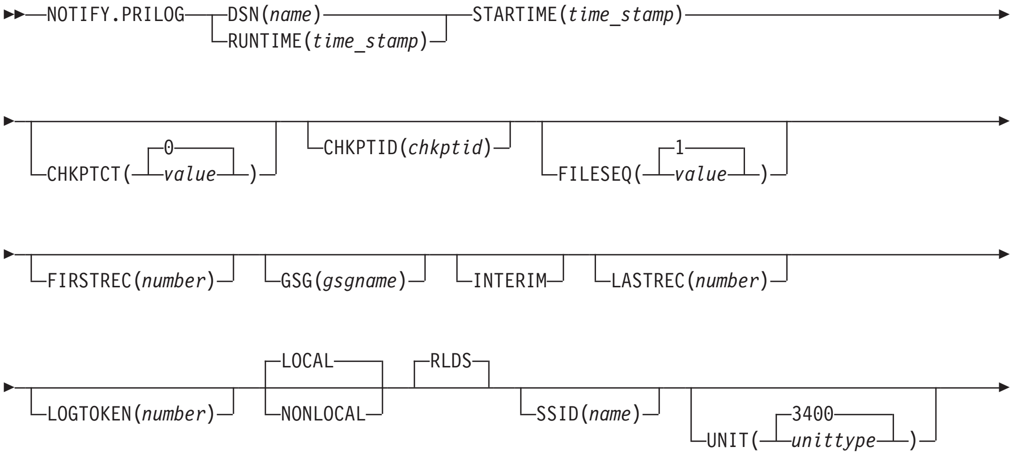
NOTIFY.PRILOG (for RLDS)

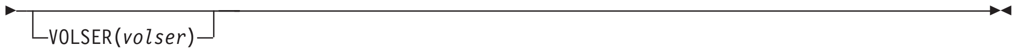
Use a `NOTIFY.PRILOG` command to add information about a primary RLDS (or a primary SLDS that a batch subsystem created) to RECON and to manually create interim-primary log data set records in RECON.

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





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 130).

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 14 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 14. Parameters of NOTIFY.PRILOG (for RLDS) Command for Open, EOV, and Close

Type of Log Entry	Required Keywords
RLDS Open	STARTIME, DSN, VOLSER, FIRSTREC
RLDS EOV	STARTIME, VOLSER, RUNTIME
RLDS Close	STARTIME, RUNTIME, LASTREC

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 130

See Table 14 for a description of the use of this parameter with other parameters in the `NOTIFY.PRILOG` command.

CHKPTCT(0 | *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 130).

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 `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.

LOGTOKEN(*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(*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 | *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(*volser*)
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──┬──┬─DSN(name)───────────┬──STARTIME(time_stamp)──────────►
                   └─TSLDS─┘  └─RUNTIME(time_stamp)─┘

►──┬────────────────┬──┬──────────────────┬──┬───────────────────────┬──────────────►
   └─CHKPTCT(value)─┘  └─CHKPTID(chkptid)─┘  │          ┌─1─────┐    │
                                             └─FILESEQ(─┴─value─┴─)──┘

                                                                          ┌─LOCAL────┐
►──┬──────────────────┬──┬──────────────┬──┬─────────┬──┬─────────────────┬──┼──────────┼──►
   └─FIRSTREC(number)─┘  └─GSG(gsgname)─┘  └─INTERIM─┘  └─LASTREC(number)─┘  └─NONLOCAL─┘

►──┬────────────┬──┬──────────────────────────┬──┬────────────────┬────────────────►◄
   └─SSID(name)─┘  │       ┌─3400─────┐       │  └─VOLSER(volser)─┘
                   └─UNIT(─┴─unittype─┴─)─────┘
```

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 130).

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 15 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 15. Parameters of NOTIFY.PRILOG (SLDS or TSLDS) Command for Open, EOV, and Close

Type of Log Entry	Required Keywords
SLDS Open	STARTIME, DSN, VOLSER, FIRSTREC
SLDS EOV	STARTIME, VOLSER, RUNTIME
SLDS Close	STARTIME, RUNTIME, LASTREC

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 130).

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 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 130).

FILESEQ(1 | *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.PRILOG` invocations for the PRISLD 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.PRILOG` invocation for the PRISLD 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 | ***unittype*)**
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.

```
//NFYPRILG JOB

⋮
//SYSIN    DD   *
     NOTIFY.PRILOG  SLDS STARTIME(820670201010)  -
                    VOLSER(VOL004)  DSN(PRILOG4) FIRSTREC(7000)
     NOTIFY.PRILOG  SLDS STARTIME(820670201010)  -
                    VOLSER(VOL005)  RUNTIME(820670202020)
     NOTIFY.PRILOG  SLDS STARTIME(820670201010)  -
                    RUNTIME(820670303030) LASTREC(8889)
/*
```

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 Version 9: 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)─┘

                                       ┌─CURRENT─────────────────────────┐  ┌─DBDS──┐
►──┬───────────────┬──┬────────────┬──┼─────────────────────────────────┼──┼───────┼──────────►◄
   └─RUNUSID(usid)─┘  └─ADDN(name)─┘  └─RUNTIME(time_stamp)─┬────────┬─┘  └─TRACK─┘
                                                             └─PITR───┘
```

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 130).

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 RUNTME parameter with the PITR parameter.

Restriction: Do not use the RCVTIME parameter when recovering from a nonstandard image copy at a tracking subsystem.

See 248 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

```
                                          ┌─CURRENT───────────┐
►►──NOTIFY.REORG──DBD(name)──DDN(name)────┤                   ├──────────────────────────►
                                          └─RUNTIME(time_stamp)┘
```

```
►──┬─────────────────────────┬──┬──────────────────────────┬──┬────────────────┬──┬─────────────────┬──►
   │           ┌─1─────┐     │  │            ┌─1─────┐     │  └─ICDSN(name)────┘  └─ICDSN2(name)────┘
   └─FILESEQ(──┴─value─┴──)──┘  └─FILESEQ2(──┴─value─┴──)──┘

   ┌─OFFLINE──────────────────────────────────────────────┐
►──┼──────────────────────────────────────────────────────┼──┬─────────────────┬──────────────────────►
   └─ONLINE──┬────────────────────────────────────────────┘  └─RECDCT(value)───┘
             │                                  ┌─NORECOV─┐  │
             └─STOPTIME(time_stamp)─────────────┼─────────┼──┘
                                                └─RECOV───┘

►──┬───────────────────────────┬──┬────────────────────────────┬──┬────────────────────────────────┬──►
   │        ┌─3400─────────┐    │  │         ┌─3400─────────┐    │  │                ┌─,────────┐    │
   └─UNIT(──┴─unittype─────┴─)──┘  └─UNIT2(──┴─unittype─────┴─)──┘  │                ▼          │    │
                                                                   └─VOLLIST(──────┴─volser───┴─)──┘

►──┬────────────────────────────────┬──┬──────────────┬─────────────────────────────────────────────►◄
   │               ┌─,────────┐     │  └─USID(value)──┘
   │               ▼          │     │
   └─VOLLIST2(─────┴─volser───┴─)───┘
```

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 130).

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 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.DBDS` 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 2 147 483 647.

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(DB1)  DDN(DD1) -
                   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(gsgname)┘  └─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 130).

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 16 indicates the keyword combinations that correspond to the type of secondary-online-log data-set entry.

Table 16. Parameters of NOTIFY.SECLOG (for OLDS) Command for Open, Switch, and Close

Type of Online Log Entry	Required Keywords
OLDS Open	STARTIME, DSN, FIRSTREC
OLDS Switch	FIRSTREC, STARTIME, DSN, NXTOLDS
OLDS Close	LASTREC, STARTIME, RUNTIME

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 130).

See Table 16 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')`.

- 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 OLDS.

GSG is required if LOGTOKEN is specified.

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.

LASTREC(*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 `CLOSE`command. It is optional for the `SWITCH` command; if it is omitted, the FIRSTREC value minus 1, is recorded for the OLDS being closed. It is invalid for an `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: `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.

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 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.

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.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)──────────────────────►
                   └─RUNTIME(time_stamp)──┘

►──┬───────────────────────┬──┬────────────────────┬──┬───────────────────────┬─────────►
   │         ┌─0─────┐     │  └─CHKPTID(chkptid)───┘  │         ┌─1─────┐     │
   └─CHKPTCT(─┴─value─┴─)──┘                          └─FILESEQ(─┴─value─┴─)──┘

                                                                 ┌─LOCAL────┐
►──┬──────────────────┬──┬──────────────┬──┬─────────┬──┬──────────────────┬──┼──────────┼──►
   └─FIRSTREC(number)─┘  └─GSG(gsgname)─┘  └─INTERIM─┘  └─LASTREC(number)──┘  └─NONLOCAL─┘

►──┬──────┬──┬────────────┬──┬──────────────────────────┬──┬────────────────┬──────────►◄
   └─RLDS─┘  └─SSID(name)─┘  │      ┌─3400─────┐        │  └─VOLSER(volser)─┘
                             └─UNIT(─┴─unittype─┴─)─────┘
```

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 130).

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 17 indicates the keyword combinations that correspond to the type of secondary recovery log data set entry.

Table 17. Parameters of NOTIFY.SECLOG (for RLDS) Command for Open, EOV, and Close

Type of Recovery Log Entry	Required Keywords
RLDS Open	STARTIME, DSN, VOLSER
RLDS EOV	STARTIME, VOLSER, RUNTIME
RLDS Close	STARTIME, RUNTIME

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 130).

If you issue a subsequent STARTIME parameter, that time is the start time of the volume. See Table 17 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 130).

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.

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 | *unittype*)
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 17 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.

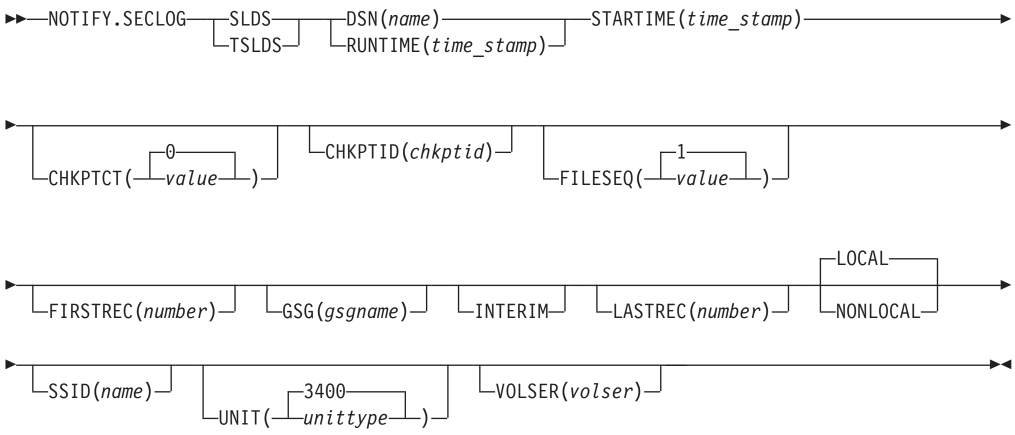
```
//NFYSECLG JOB

⋮
//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



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.

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 130).

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 18 indicates the keyword combinations that correspond to the type of secondary system log data set entry.

Table 18. Parameters of NOTIFY.SECLOG (for SLDS or TSLDS) Command for Open, EOV, and Close

Type of System Log Entry	Required Keywords
SLDS Open	STARTIME, DSN, VOLSER
SLDS EOV	STARTIME, VOLSER, RUNTIME
SLDS Close	STARTIME, RUNTIME

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 130).

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 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 130).

FILESEQ(1 | *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(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.

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 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(3400 | *unittype*)
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(*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 18 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.

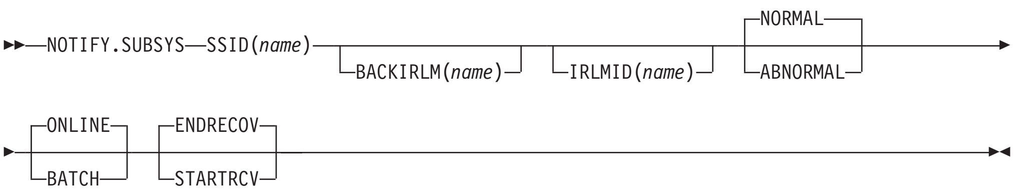
```
//NFYSECLG JOB

⋮
//SYSIN    DD   *
     NOTIFY.SECLOG   SLDS STARTIME(820670201010)  -
                     SSID(IMSC) DSN(DSN006) VOLSER(VOL009)
     NOTIFY.SECLOG   SLDS STARTIME(820670201010)  -
                     RUNTIME(820680204500)  VOLSER(VOL003)
     NOTIFY.SECLOG   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.

If you specify this command when you are in a recovery-only environment, the command fails.

NOTIFY.SUBSYS Syntax



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.

BACKIRLM(*name*)
Optional parameter you use to specify the name of the alternate subsystem IRLM. *name* is a five-character alphanumeric string. When you specify BACKIRLM, you must also specify IRLMID.

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 29 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

```
                                            ┌─CURRENT─────────────┐
►►──NOTIFY.UIC──DBD(name)──┬─DDN(name)──┬──┼─────────────────────┼──┬─────────────────┬──►
                           └─AREA(name)─┘  └─RUNTIME(time_stamp)─┘  └─UDATA('string')─┘

►──┬──────────────┬──────────────────────────────────────────────────────────────────►◄
   └─USID(number)─┘
```

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 130).

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.

```
 //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 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

```
►►──RESET.GSG──GSGNAME(gsgname)──┬─BOTH───┬──────────────────────────────────►◄
                                 ├─RECON1─┤
                                 └─RECON2─┘
```

Parameters for RESET.GSG

GSGNAME(*gsgname*)
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
/*
```

Part 3. DBRC Application Programming Interface (API) Reference

Chapter 16. Using the DBRC API

The DBRC 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 421)
- 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 417)
- Shutdown DBRC and the API (see Chapter 20, “DBRC Stop Request,” on page 425)

The API is provided with IMS in the DSPAPI macro.

Prerequisite Knowledge: In order 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” on page 350
- “Ending the DBRC API Environment” on page 350
- “Addressing and Residency Mode When Using the DBRC API” on page 350

- “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”
- “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 program 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:

```
►►──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,                                                                        C
       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” on page 352). 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 can not 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 to12 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 to12 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 354
- “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 57 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 Version 9: 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:

```
yyyydddFhhmmsstxxxxxfqqs
```

Where:

yyyy Is the year (0000 to 9999)

ddd Is the day (000 to 366)

F Is a hexidecimal 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 421). 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 Version 9: 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 393.
- Online Log Data Set (TYPE=OLDS) - See “OLDS Query” on page 402.
- RECON status (TYPE=RECON) - See “RECON Status Query” on page 407. 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 411.

Related Reading: See “Records in the RECON Data Set” on page 68 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 522 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 417.

Figure 15 illustrates the general format of the output from a Query request. The sample DSECT that follows Figure 15 describes in detail the fields of the storage blocks and their relationship to each other.

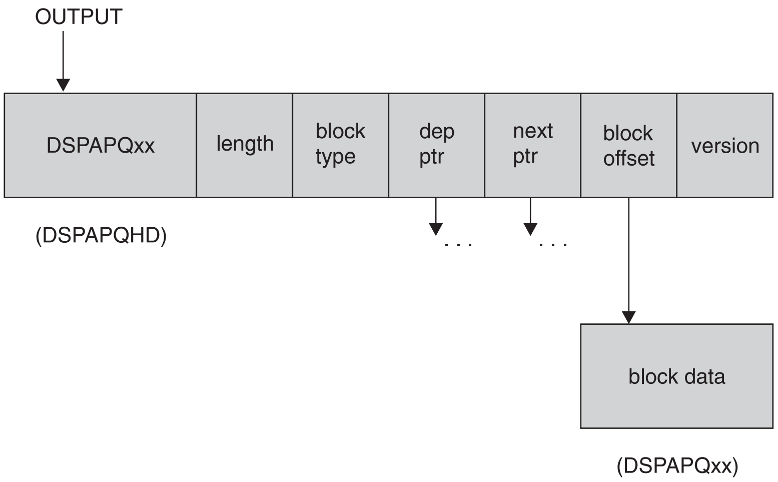


Figure 15. General Format of Query Output

Each storage block begins with a standard header mapped by DSPAPQHD, as illustrated in Figure 16.

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH   NAME (DIM)        DESCRIPTION
======== ======== ======== ========   ==============    ===============================
       0      (0) STRUCTURE       32   APQHD
       0      (0) CHARACTER        8   APQHD_EYECATCHER  Output area eyecatcher
       8      (8) SIGNED           4   APQHD_LENGTH      Block length, hdr + data
      12      (C) SIGNED           2   APQHD_BLKTYPE     Block type
      14      (E) SIGNED           2   *                 Reserved
      16     (10) ADDRESS          4   APQHD_DEPPTR      Ptr to block dependent
      20     (14) ADDRESS          4   APQHD_NEXTPTR     Ptr to next block of the same type
      24     (18) UNSIGNED         4   APQHD_BLKOFFSET   Offset to block data
      28     (1C) SIGNED           4   APQHD_VERSION     Version of output block
```

Figure 16. DSECT of DSPAPQHD - QUERY Output Block Header

Query Requests Reference

The following topics 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 393
- "OLDS Query" on page 402
- "RECON Status Query" on page 407
- "Subsystem Query" on page 411

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

```
                                                                    ┌─SSID=*────┐
►►──┬──────┬──DSPAPI──FUNC=QUERY──TYPE=BACKOUT──TOKEN=address──┴───────────┴──────────►
    └─name─┘                                                        └─SSID=name─┘

  ┌─MF=S──────────────────────────┐
►─┴───────────────────────────────┴──OUTPUT=output──┬──────────────────────┬──────────►
  ├─MF=(S,list)───────────────────┤                 └─RETCODE=returncode───┘
  ├─MF=(L,list)───────────────────┤
  │                ┌─,COMPLETE─┐  │
  ├─MF=(M,list─────┴───────────┴─)┤
  │                └─,NOCHECK──┘  │
  │                ┌─,COMPLETE─┐  │
  └─MF=(E,list─────┴───────────┴─)┘
                   └─,NOCHECK──┘

                          ┌─VERSION=1.0────┐
►──┬──────────────────┬───┴────────────────┴──────────────────────────────────────►◄
   └─RSNCODE=reasoncode┘   └─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=*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 subsystem name for the backout being queried. Specify '*' to request 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=*symbol* **| (2 - 12)**
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 (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=BACKOUT Query Request

Table 19 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 Version 9: Diagnosis Guide and Reference*.

Table 19. Return and Reason Codes for TYPE=BACKOUT Query Requests

Return Codes	Reason Codes	Meaning
X'00000000'	X'00000000'	Request completed successfully.
Warning		
X'00000008'	X'D87000001'	No backout records exist.
Severe Error		
X'0000000C'	X'C9000001'	Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.
Storage Error		
X'00000028'	X'D87000001'	Error obtaining storage for BACKOUT block.
X'00000028'	X'D91000001'	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.
Internal Error		
X'0000002C'	all	DBRC internal error. Refer to “Database Recovery Control Service Aids” in the *IMS Version 9: Diagnosis Guide and Reference* for further information.

Table 19. Return and Reason Codes for TYPE=BACKOUT Query Requests (continued)

Return Codes	Reason Codes	Meaning
Parameter Error		
X'00000030'	X'C9000001'	Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.
X'00000030'	X'C9000002'	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.
X'00000030'	X'C9000003'	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.
X'00000030'	X'C9000004'	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.
X'00000030'	X'C9000005'	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.
X'00000030'	X'C9000010'	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.
X'00000030'	X'D80000001'	Parameter error. Missing or invalid OUTPUT parameter.

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.

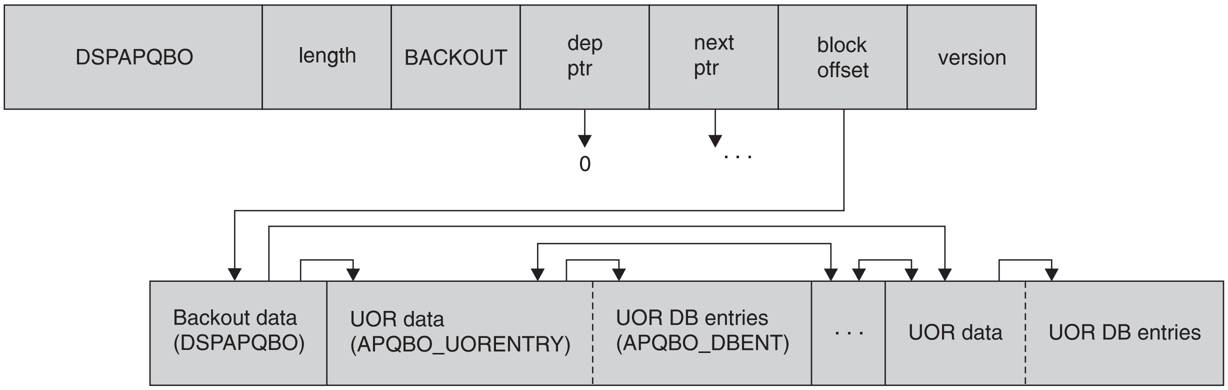


Figure 17. Format for QUERY TYPE=BACKOUT Output

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)       DESCRIPTION
======== ======== ======== ========  ==============   ===============================
      0      (0) STRUCTURE       48  DSPAPQBO
      0      (0) CHARACTER        8   APQBO_SSID      Subsystem identifier
      8      (8) UNSIGNED         4   APQBO_FIRSTUOR  Offset of first UOR entry
     12      (C) UNSIGNED         4   APQBO_LASTUOR   Offset of last UOR entry
     16     (10) CHARACTER       12   APQBO_TIMEFIRST Earliest UOR time
     28     (1C) CHARACTER       12   APQBO_TIMELAST  Latest UOR time
     40     (28) BIT(8)           1   APQBO_FLAGS     Backout flags
                  1... ....            APQBO_SAVER    SAVUOR call during restart
     41     (29) CHARACTER        3   *               Reserved
     44     (2C) SIGNED           4   APQBO_UORCOUNT  Number of UORs

      ==============================================================================
      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 ith database for this UOR is given by:
          rfy apqbo_DBent
                      based(addr(apqbo_UORentry) + apqbo_DBoffset
                            + (i-1) apqbo_DBLength)
      ==============================================================================
```

Figure 18. DSECT of DSPAPQBO (Part 1 of 2)

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE       64  APQBO_UORENTRY Unit of Recovery entry
       0      (0) CHARACTER       64  APQBO_PREFIX   Prefix section
       0      (0) UNSIGNED         4  APQBO_NEXTUOR  Offset of next UOR entry
       4      (4) UNSIGNED         4  APQBO_PREVUOR  Offset of previous UOR entry
       8      (8) UNSIGNED         4  APQBO_DBOFFSET Offset to DB entries
      12      (C) CHARACTER       12  APQBO_UORTIME  Time stamp for this UOR
      24     (18) CHARACTER        8  APQBO_UORPSB   PSB name
      32     (20) BIT(16)          2  APQBO_UORFLAGS
                  1... ....           APQBO_DEFBO    Deferred backout - dynamic backout failure
                  .1.. ....           APQBO_INFLT    Inflight UOR
                  ..1. ....           APQBO_INDOU    Indoubt UOR
                  ...1 ....           APQBO_BMP      BMP UOR
                  .... 1...           APQBO_BOCAN    BBO identified candidate
                  .... .1..           APQBO_COLDN    Cold start ended for UOR
                  .... ..1.           APQBO_BBOK     Backed out OK by BBO
                  .... ...1           APQBO_CMD      UOR modified by command
      33     (21) 1... ....           APQBO_BATCH    Batch IMS UOR
      34     (22) CHARACTER        6  *              Reserved
      40     (28) CHARACTER       16  APQBO_RTOKN    Recovery token
      40     (28) CHARACTER        8  APQBO_RTSSID   SSID for this token
      48     (30) CHARACTER        8  APQBO_UORID    Unique UOR ID
      56     (38) SIGNED           4  APQBO_DBCOUNT  Number of DBs for this UOR
      60     (3C) UNSIGNED         2  APQBO_DBLENGTH Length of each DB entry
      62     (3E) CHARACTER        2  *              Reserved

 OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE       16  APQBO_DBENT    Database entry
       0      (0) CHARACTER        8  APQBO_DBNAME   Database name
       8      (8) BIT(8)           1  APQBO_DBFLAGS  Flags
                  1... ....           APQBO_DBOUT    UOR backed out for this DB
                  .1.. ....           APQBO_DBDEF    Dyn backout failure this DB
       9      (9) CHARACTER        7  *              Reserved

CONSTANTS

  LEN  TYPE           VALUE          NAME           DESCRIPTION
=====  =========  =================  ============== ===============================
    8  CHARACTER   DSPAPQBO          APQBO_EYECATCHER
```

Figure 18. DSECT of DSPAPQBO (Part 2 of 2)

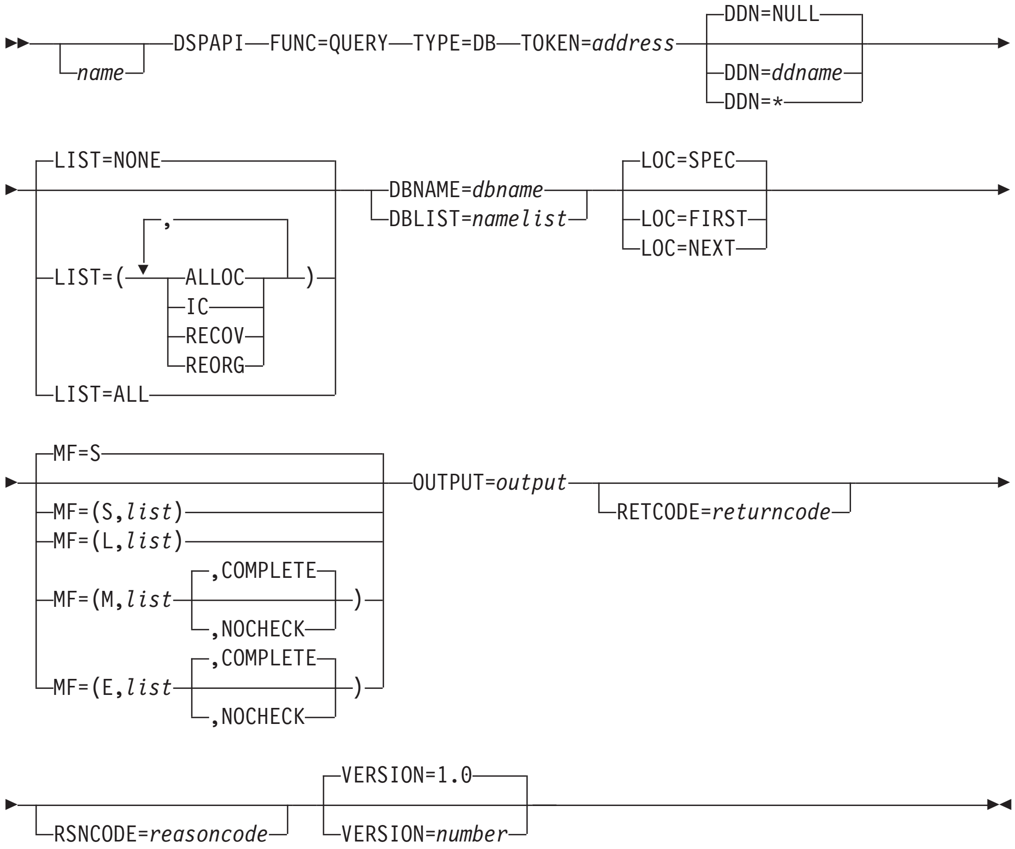
Database Query

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



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 also be specified indicating that information about all DBDSs or areas is requested. 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=*dbname* 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 20 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 *IMS Version 9: Diagnosis Guide and Reference*.

Table 20. Return and Reason Codes for TYPE=DB Queries

Return Codes	Reason Codes	Meaning
X'00000000'	X'00000000'	Request completed successfully.
	Partial Success	
X'00000004'	X'D8200001'	One or more databases in DBLIST is not registered in the RECON. A “DB not found” data block with an eyecatcher of DSPAPQNF is built for each database that was not found. The block is in the chain of blocks returned.
	Warning	
X'00000008'	X'D8200001'	None of the databases in the DBLIST are registered in the RECON. No information blocks are returned.
X'00000008'	X'D8200002'	The specified database is not registered in the RECON. No information blocks are returned.
X'00000008'	X'D8210002'	The specified DBDS or Area is not registered in the RECON. No information blocks are returned.
	Severe Error	
X'0000000C'	X'C9000001'	Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.
	Storage Error	
X'00000028'	X'D8200001'	Error obtaining storage for IMSDB block.
X'00000028'	X'D8200002'	Error obtaining storage for HALDB block.
X'00000028'	X'D8200003'	Error obtaining storage for FPDEDB block.
X'00000028'	X'D8200004'	Error obtaining storage for DB not found block (DBNOTFND).
X'00000028'	X'D8210001'	Error obtaining storage for DBDS block.

Table 20. Return and Reason Codes for TYPE=DB Queries (continued)

Return Codes	Reason Codes	Meaning
X'00000028'	X'D8210002'	Error obtaining storage for AREA block.
X'00000028'	X'D8210003'	Error obtaining storage for RCVINFO block.
X'00000028'	X'D8210004'	Error obtaining storage for ALLOC block.
X'00000028'	X'D8210005'	Error obtaining storage for IC block.
X'00000028'	X'D8210006'	Error obtaining storage for REORG block.
X'00000028'	X'D8210007'	Error obtaining storage for RECOV block.
X'00000028'	X'D8220001'	Error obtaining storage for PART block.
X'00000028'	X'D9100001'	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.
Internal Error		
X'0000002C'	all	DBRC internal error. Refer to “Database Recovery Control Service Aids” in the *IMS Version 9: Diagnosis Guide and Reference* for further information.
Parameter Error		
X'00000030'	X'C9000001'	Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.
X'00000030'	X'C9000002'	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.
X'00000030'	X'C9000003'	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.
X'00000030'	X'C9000004'	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.
X'00000030'	X'C9000005'	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.
X'00000030'	X'C9000008'	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.
X'00000030'	X'C9000009'	Invalid DDN address. The address of the field containing the DDN failed validity checking. The address specifies storage not owned by the calling program.
X'00000030'	X'D8000001'	Parameter error. Missing or invalid OUTPUT parameter.
X'00000030'	X'D8200001'	Parameter error. LOC parameter is not allowed with DBLIST.

Table 20. Return and Reason Codes for TYPE=DB Queries (continued)

Return Codes	Reason Codes	Meaning
X'00000030'	X'D8200002'	Parameter error. DBNAME parameter is not allowed with LOC=FIRST.
X'00000030'	X'D8200003'	Parameter error. DBNAME parameter is required with LOC=NEXT.
X'00000030'	X'D8200004'	Parameter error. DBNAME or DBLIST is required.
X'00000030'	X'D8200005'	Parameter error. Count of databases in DBLIST is zero.
X'00000030'	X'D8200006'	Parameter error. Database information is being requested for a HALDB partition. DBNAME or DBLIST contains a partition name.

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:

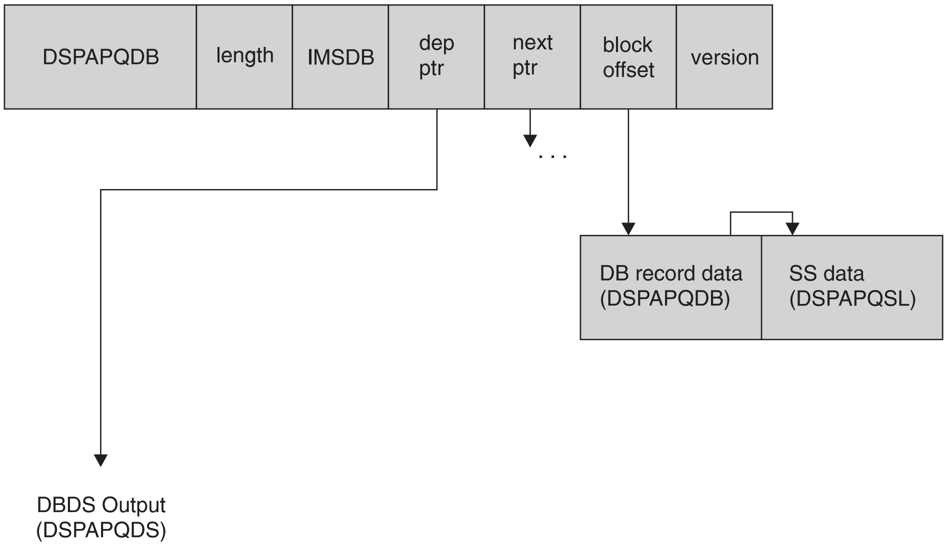


Figure 19. Format of QUERY TYPE=DB (Full Function) Output

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.

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE       LENGTH  NAME (DIM)      DESCRIPTION
======== ======== ========= ========  ==============  ===============================
       0      (0) STRUCTURE       96  DSPAPQDB
       0      (0) CHARACTER        8   APQDB_DBNAME   Database name
       8      (8) UNSIGNED         4   APQDB_SSLIST   Offset to SS list (DSPAPQSL), zero is no SS auth'd
      12      (C) SIGNED           4   *(3)           Reserved
      24     (18) UNSIGNED         2   APQDB_IRCNT    IC receive needed counter
      26     (1A) BIT(8)           1   APQDB_AUFLAG   Authorization flags
                  1... ....             APQDB_BKFLG   Backout needed flag
                  .1.. ....             APQDB_PAFLG   Prohibit authorization
                  ..1. ....             APQDB_RDFLG   Read only SS auth
                  ...1 ....             APQDB_NONRV   Non-recoverable
      27     (1B) CHARACTER        5   APQDB_IRLMAU   IRLMID of auth SS
      32     (20) SIGNED           2   APQDB_RCVCTR   Recovery needed count
      34     (22) SIGNED           2   APQDB_ICCTR    IC needed count
      36     (24) SIGNED           2   APQDB_ICRECCTR IC recommended counter
      38     (26) UNSIGNED         1   APQDB_SHRLVL   Share level of DB
      39     (27) UNSIGNED         1   APQDB_HELDAU   Held auth state
                  1... ....             APQDB_HAUBIT  High order bit flag
      40     (28) UNSIGNED         2   APQDB_DMBNUM   Global DMB number
      42     (2A) SIGNED           2   APQDB_SSNUM    # of SS auth DB
      44     (2C) UNSIGNED         2   APQDB_SSENTLEN Length of each SS entry
      46     (2E) UNSIGNED         1   APQDB_CACCSS   Access state for chg auth
      47     (2F) UNSIGNED         1   APQDB_CANCDD   Encode state for chg auth
      48     (30) UNSIGNED         1   APQDB_CAHELD   Held state for chg auth
      49     (31) CHARACTER        3   *              Reserved
      52     (34) UNSIGNED         2   APQDB_EQECNT   Total EQE count
      54     (36) BIT(16)          2   APQDB_RSRFLG   Flags
                  1... ....             APQDB_RCVTRK  Recovery level tracking
                  .1.. ....             APQDB_TRKSPN  Tracking is suspended
                  ..1. ....             APQDB_PURBIT  Suspended by time
                  ...1 ....             APQDB_ICNDIS  IC needed disabled option
      56     (38) CHARACTER        8   APQDB_GSGNAME  Global Service Group name
      64     (40) UNSIGNED         4   APQDB_USID     Last alloc USID
      68     (44) UNSIGNED         4   APQDB_AUSID    Last authorized USID
      72     (48) UNSIGNED         4   APQDB_RUSID    Last received USID
      76     (4C) UNSIGNED         4   APQDB_HUSID    Hardened by tracker USID
      80     (50) UNSIGNED         4   APQDB_RNUSID   Receive needed USID
      84     (54) CHARACTER        8   APQDB_RECOVGRP Recovery Group name
      92     (5C) CHARACTER        4   *              Reserved

CONSTANTS

  LEN  TYPE           VALUE          NAME            DESCRIPTION
=====  =========  =================  ==============  ===============================
    8  CHARACTER   DSPAPQDB          APQDB_EYECATCHER
```

Figure 20. DSECT of DSPAPQDB

```
DSPAPQSL

  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)      DESCRIPTION
======== ======== ========= ========  ==============  ===============================
       0      (0) STRUCTURE       16  DSPAPQSL
       0      (0) CHARACTER       16   APQSL_SSINFO   Subsystem list entry
       0      (0) CHARACTER        8    APQSL_SSNAME  Subsystem ID
       8      (8) UNSIGNED         1    APQSL_ACCESS  Access intent
       9      (9) UNSIGNED         1    APQSL_NCDDST  Encoded state
      10      (A) BIT(8)           1    APQSL_SSFLGS  Flags
                  1... ....              APQSL_SSROLE 0 - Active SS, 1 - Tracking SS
                  .1.. ....              APQSL_SSXRFC 1 - XRF Capable
      11      (B) BIT(8)           1    *             Reserved
      12      (C) CHARACTER        4    APQSL_BKINFO  Backout information
      12      (C) SIGNED           2     APQSL_BKCTR  Backout needed count
      14      (E) SIGNED           2     APQSL_BKNUM  Backout done count

CONSTANTS

  LEN  TYPE           VALUE          NAME             DESCRIPTION
=====  =========  =================  ===============  ===============================

     Possible access intent values (apqsl_ACCESS)
=====================================================================================
    1  HEX        01                 APQSL_ACCRO      READ-GO
    1  HEX        02                 APQSL_ACCRD      READ
    1  HEX        03                 APQSL_ACCUP      UPDATE
    1  HEX        04                 APQSL_ACCEX      EXCLUSIVE
```

Figure 21. DSECT of DSPAPQSL

Fast Path DEDB Output:

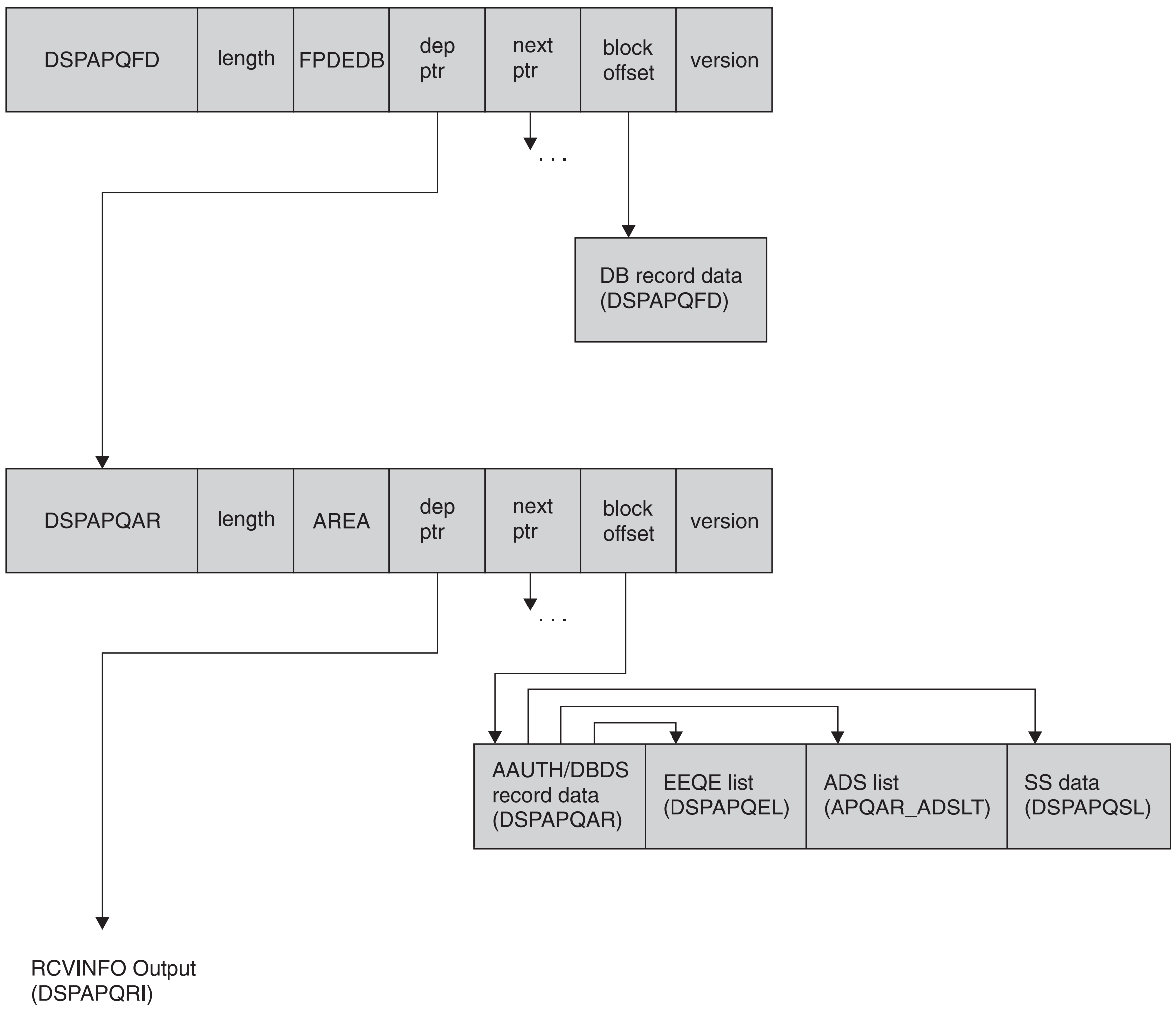


Figure 22. Format of QUERY TYPE=DB (Fast Path DEDB) Output

The area information is returned only if DDN is specified. Recovery information (RCVINFO) is only returned if the LIST parameter is specified.

Figure 23 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.

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)      DESCRIPTION
======== ======== ========= ========  ==============  ===============================
       0      (0) STRUCTURE       38  DSPAPQFD
       0      (0) CHARACTER        8   APQFD_DBNAME   Database name
       8      (8) SIGNED           4   *(4)           Reserved
      24     (18) SIGNED           2   APQFD_RCVCTR   Recovery Needed Counter
      26     (1A) SIGNED           2   APQFD_ICCTR    IC Needed Counter
      28     (1C) SIGNED           2   APQFD_ICRECCTR IC Recommended Counter
      30     (1E) UNSIGNED         2   APQFD_DMBNUM   Global DMB number
      32     (20) UNSIGNED         2   APQFD_EQECNT   Total EEQE count
      34     (22) SIGNED           2   APQFD_AUTHDAREAS  Number of Areas authorized
      36     (24) UNSIGNED         1   APQFD_SHRLVL   Share Level
      37     (25) BIT(8)           1   APQFD_FLAGS    Flags
                  1... ....             APQFD_PAFLG   Prohibit authorization
                  .1.. ....             APQFD_NONRV   Non-recoverable
                  ..1. ....             APQFD_ICNDIS  IC needed disabled option

CONSTANTS

  LEN  TYPE           VALUE          NAME             DESCRIPTION
=====  =========  =================  ===============  ===============================
    8  CHARACTER   DSPAPQFD          APQFD_EYECATCHER
```

Figure 23. DSECT of DSPAPQFD

```
 OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)      DESCRIPTION
======== ======== ========= ========  ==============  ===============================
       0      (0) STRUCTURE      196  DSPAPQAR
       0      (0) CHARACTER        8   APQAR_DBNAME   Database name
       8      (8) CHARACTER        8   APQAR_AREANAME Area name
      16     (10) UNSIGNED         4   APQAR_EEQELIST Offset to EEQE list
                                                      (DSPAPQEL), zero if no EEQEs
      20     (14) UNSIGNED         4   APQAR_SSLIST   Offset to SS list (DSPAPQSL),
                                                      zero if no SS auth'd
      24     (18) UNSIGNED         4   APQAR_ADSLIST  Offset to ADS list, zero if
                                                      none registered
      28     (1C) SIGNED           4   *              Reserved
      32     (20) UNSIGNED         1   APQAR_SHRLVL   Share level of DB
      33     (21) UNSIGNED         1   APQAR_HELDAU   Held auth state
                  1... ....             APQAR_HAUBIT  High order bit flag
      34     (22) UNSIGNED         2   APQAR_DMBNUM   Global DMB number
      36     (24) SIGNED           2   APQAR_SSNUM    # subsytems authd to Area
      38     (26) UNSIGNED         2   APQAR_SSENTLEN Length of each SS entry
      40     (28) UNSIGNED         1   APQAR_CACCSS   Access state for CHG AUTH
      41     (29) UNSIGNED         1   APQAR_CANCDD   Encoded state for CHG AUTH
      42     (2A) UNSIGNED         1   APQAR_CAHELD   Held state for CHG AUTH
      43     (2B) CHARACTER        5   APQAR_IRLMAU   IRLMID of auth SS
      48     (30) BIT(16)          2   APQAR_FLAGS
                  1... ....             APQAR_RECYC   REUSE image copies
                  .1.. ....             APQAR_ICREC   Image Copy Recommended
                  ..1. ....             APQAR_IC      Image Copy Needed
                  ...1 ....             APQAR_ICNDIS  IC needed disabled option 1 =
                                                      IC Needed Disabled
                  .... 1...             APQAR_RECOV   Recovery needed
                  .... .1..             APQAR_INPRO   HSSP CIC in progress
                  .... ..1.             APQAR_GT240   M/C FP GT240 area DEDB
                  .... ...1             APQAR_VSO     VSO flag
      49     (31) 1... ....             APQAR_PREOP   PREOPEN flag
                  .1.. ....             APQAR_PRELD   PRELOAD flag
                  ..1. ....             APQAR_LKASD   VSO CF buffer lookaside
                  ...1 ....             APQAR_MAS     VSO area resides in multi-area
                                                      CF structure
                  .... 1...             APQAR_RRGAL   REORG since last ALLOC, only
                                                      set if RSR-covered
                  .... .1..             APQAR_TSRAL   TS recov since last ALLOC, only
                                                      set if RSR-covered
      50     (32) BIT(8)           1   APQAR_RSRFLAGS Remote Site Recovery flags
                  1... ....             APQAR_RCVTRK  Recovery Level Tracking
                  .1.. ....             APQAR_TRKSPN  Tracking was suspended
                  ..1. ....             APQAR_PURBIT  Suspended by time
                  ...1 ....             APQAR_RCVRQ   Receive Required
      51     (33) BIT(8)           1   APQAR_AUFLAG   Authorization flags
                  1... ....             APQAR_PAFLG   Prohibit authorization
                  .1.. ....             APQAR_NONRV   Non-recoverable
      52     (34) BIT(8)           1   APQAR_DSORG    Data set organization
                  1... ....             APQAR_VSAM    1 = VSAM, 0 = NON-VSAM
                  .1.. ....             APQAR_INDEX   0 = Non-indexed (OSAM or ESDS),
                                                      1 = Indexed(ISAM or KSDS)
                  ..11 1111             *             Reserved - zeroes
```

Figure 24. DSECT of DSPAPQAR (Part 1 of 2)

```
       53     (35) CHARACTER        1   APQAR_DBORG   IMS DB organization
      54     (36) CHARACTER        8   APQAR_GSGNAME  GSG Name
      62     (3E) CHARACTER        2   *              Reserved
      64     (40) UNSIGNED         4   APQAR_USID     Last ALLOC USID
      68     (44) UNSIGNED         4   APQAR_AUSID    Last authorized USID
      72     (48) UNSIGNED         4   APQAR_RUSID    Last received USID
      76     (4C) UNSIGNED         4   APQAR_HUSID    Hardened USID
      80     (50) UNSIGNED         4   APQAR_RNUSID   Receive needed USID
      84     (54) CHARACTER        8   APQAR_RECOVGRP Recovery Group name
      92     (5C) CHARACTER        8   APQAR_CAGRPNAME Change Accum group name
     100     (64) UNSIGNED         2   APQAR_GENMX    Max number of ICs that may be
                                                      predefined for this area
     102     (66) UNSIGNED         2   APQAR_GENNO    Number of available ICs for this area
     104     (68) UNSIGNED         2   APQAR_USDIC    Number of ICs used
     106     (6A) SIGNED           2   APQAR_EEQECOUNT EEQE count
     108     (6C) UNSIGNED         2   APQAR_EEQELENGTH EEQE entry length
     110     (6E) UNSIGNED         1   APQAR_NOADS    # of ADS in the area
     111     (6F) UNSIGNED         1   APQAR_AVADS    # of available ADS
     112     (70) UNSIGNED         2   APQAR_ADSLENGTH ADS entry length
     114     (72) CHARACTER        2   *              Reserved
     116     (74) CHARACTER       40   APQAR_JCL      GENJCL members
     116     (74) CHARACTER        8    APQAR_ICJCL   Image copy member
     124     (7C) CHARACTER        8    APQAR_OIJCL   Online IC member
     132     (84) CHARACTER        8    APQAR_RCJCL   Recovery member
     140     (8C) CHARACTER        8    APQAR_DFJCL   DEFLTJCL member
     148     (94) CHARACTER        8    APQAR_RVJCL   Receive JCL member
     156     (9C) UNSIGNED         2   APQAR_RTPRD    IC retention period
     158     (9E) UNSIGNED         2   APQAR_DSID     IMS data set ID
     160     (A0) UNSIGNED         4   APQAR_DSSN     Data set sequence number
     164     (A4) CHARACTER       16   APQAR_CFST1    VSO CF Structure 1
     180     (B4) CHARACTER       16   APQAR_CFST2    VSO CF Structure 2

  OFFSET   OFFSET
 DECIMAL     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
                  1... ....             APQAR_ADSAV   Avail status of ADS
                  .1.. ....             APQAR_ADSFM   Format status of create util
                  ..1. ....             APQAR_ADSCP   Copy status of create util
      53     (35) CHARACTER        3   *              Reserved

CONSTANTS

  LEN  TYPE           VALUE          NAME             DESCRIPTION
=====  =========  =================  ===============  ===============================
    8  CHARACTER   DSPAPQAR          APQAR_EYECATCHER
```

Figure 24. DSECT of DSPAPQAR (Part 2 of 2)

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)      DESCRIPTION
======== ======== ========= ========  ==============  ===============================
       0      (0) STRUCTURE       13  DSPAPQEL
       0      (0) CHARACTER       13   APQEL_EEQEENTRY EEQE descriptor entry
       0      (0) BIT(8)           1    APQEL_EQEFG   EEQE flags
                  1... ....              APQEL_ERTL   Toleration error
                  .1.. ....              APQEL_ERRD   Read error
                  ..1. ....              APQEL_ERWT   Write error
                  ...1 ....              APQEL_ERUS   DBRC user modified
                  .... 1...              APQEL_ERPM   DBRC permanent error
                  .... .1..              APQEL_INDT   Indoubt EEQE
                  .... ..1.              APQEL_CIIND  Index CI indicator
       1      (1) CHARACTER        4    APQEL_EQE     EEQE
       5      (5) CHARACTER        8    APQEL_SSID    SSID which owns the EEQE
```

Figure 25. DSECT of DSPAPQEL

HALDB (Master and All Partitions) Output:

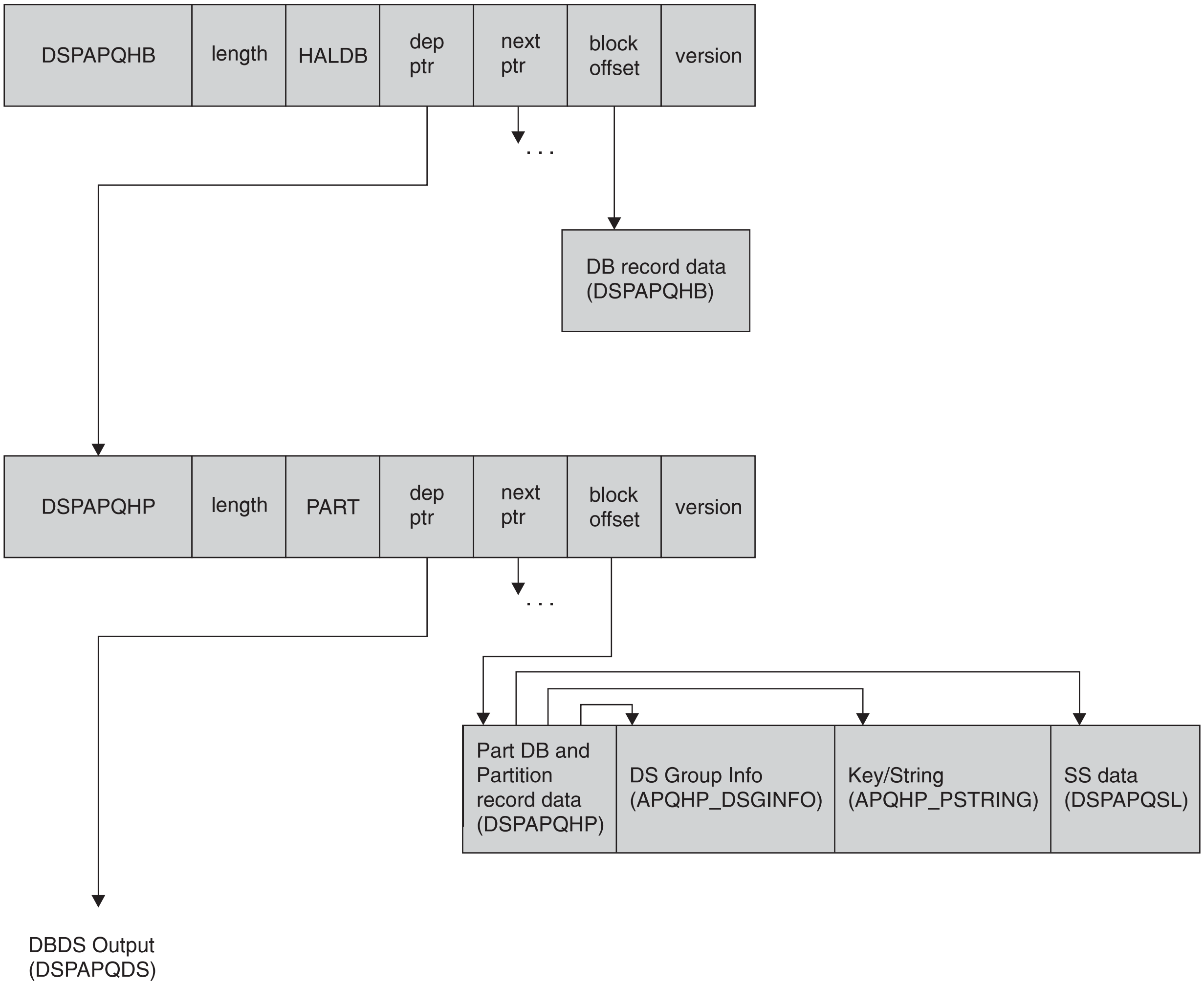


Figure 26. Format of QUERY TYPE=DB (HALDB and Partitions) Output

The DBDS information is returned only if DDN is specified.

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.

```
  OFFSET   OFFSET
 DECIMAL      HEX  TYPE        LENGTH  NAME (DIM)      DESCRIPTION
======== ======== ========= ========  ==============  ===============================
       0      (0) STRUCTURE       60  DSPAPQHB
       0      (0) CHARACTER        8   APQHB_DBNAME   HALDB name
       8      (8) SIGNED           4   *(4)           Reserved
      24     (18) BIT(8)           1   APQHB_FLAGS    Flags
                  1... ....             APQHB_NONRV   Non-recoverable
                  .1.. ....             APQHB_ICNDIS  IC needed disabled
                  ..1. ....             APQHB_OLRCAP  HALDB is OLR capable
      25     (19) BIT(8)           1   APQHB_ORG      DB organization
                  1... ....             APQHB_PSINDEX PSINDEX DB
                  .1.. ....             APQHB_PHIDAM  PHIDAM DB
                  ..1. ....             APQHB_PHDAM   PHDAM DB
                  ...1 ....             APQHB_OSAM    OSAM DB
                  .... 1111             *             Reserved
      26     (1A) UNSIGNED         1   APQHB_SHRLVL   Share level
      27     (1B) UNSIGNED         1   APQHB_DSGCNT   # DS Group members
      28     (1C) UNSIGNED         2   APQHB_DMBNUM   Global DMB number
      30     (1E) UNSIGNED         2   APQHB_PARTID   Current Partition ID
      32     (20) SIGNED           2   APQHB_PART#    Number of parts in HALDB
      34     (22) UNSIGNED         2   APQHB_VERSION# Version number
      36     (24) CHARACTER        8   APQHB_PSNAME   Name of Part Sel Routine
      44     (2C) CHARACTER        8   APQHB_GSGNAME  GSG name
      52     (34) CHARACTER        8   APQHB_RECOVGRP Recovery Group name

CONSTANTS

  LEN  TYPE           VALUE          NAME             DESCRIPTION
=====  =========  =================  ===============  ===============================
    8  CHARACTER   DSPAPQHB          APQHB_EYECATCHER
```

Figure 27. DSECT of DSPAPQHB

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE      192  DSPAPQHP
       0      (0) CHARACTER        8  APQHP_HALDBNAME  HALDB name
       8      (8) CHARACTER        8  APQHP_PARTNAME   Partition name
      16     (10) UNSIGNED         4  APQHP_SSLIST     Offset to SS list (DSPAPQSL),
                                                       zero if no SS auth'd
      20     (14) UNSIGNED         4  APQHP_KEYSTRING  Offset to KEYSTRING
                                                       (apqhp_PString), zero is no
                                                       key/string
      24     (18) UNSIGNED         4  APQHP_DSGINFOOFFSET
                                                       Offset to data set group
                                                       information
      28     (1C) SIGNED           4  *                Reserved
      32     (20) CHARACTER       44  *
      32     (20) CHARACTER       37  APQHP_DSNBASE  Base Partition DSN
      76     (4C) CHARACTER       18  APQHP_HDAM     PHDAM fields
      76     (4C) CHARACTER        8  APQHP_RMNAME   Randomizing module name
      84     (54) SIGNED           4  APQHP_RBN      Max relative block number
      88     (58) SIGNED           4  APQHP_BYTES    Max # of bytes
      92     (5C) UNSIGNED         2  APQHP_ANCHR    # of root anchor points
      94     (5E) UNSIGNED         1   APQHP_FBFF    Free block frequency factor
      95     (5F) UNSIGNED         1   APQHP_FSPF    Free space percentage factor
      96     (60) UNSIGNED         2   APQHP_PARTID  Partition ID
      98     (62) SIGNED           2   APQHP_PSTLN   Length of Part Key/String,
                                                     apqhp_PString
     100     (64) UNSIGNED         2   APQHP_DSGINFOLEN
                                                     Length of each
                                                     apqhp_DSGinfo entry
     102     (66) UNSIGNED         1   APQHP_DSGCNT  DSG count
     103     (67) BIT(8)           1   APQHP_FLAGS   Flags
                  1... ....             APQHP_PINIT  Partition must be initialized
                  .1.. ....             APQHP_ORDBDS 0=A-J/1=M-V DBDS active
                  ..1. ....             APQHP_OLRON  OLR active
                  ...1 ....             APQHP_DISAB  Partition Disabled
                  .... 1...             APQHP_MVDBDS 1 = M-V DBDS exist
                  .... .1..             APQHP_OLRCAP Partition is OLR capable
        ============================================================================
         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.
        ============================================================================
     104     (68) CHARACTER        8   APQHP_PREV    DDN of previous partition
     112     (70) CHARACTER        8   APQHP_NEXT    DDN of next partition
     120     (78) CHARACTER        8   APQHP_OLRIMS  Owning IMS for OLR
     128     (80) UNSIGNED         2   APQHP_IRCNT   IC receive needed counter
     130     (82) BIT(8)           1   APQHP_AUFLAG  Authorization flags
                  1... ....             APQHP_BKFLG  Backout needed
                  .1.. ....             APQHP_PAFLG  Prohibit authorization
                  ..1. ....             APQHP_RDFLG  Read only SS auth
                  ...1 ....             APQHP_NONRV  Non-recoverable
```

Figure 28. DSECT of DSPAPQHP (Part 1 of 2)

```
     131     (83) CHARACTER        5   APQHP_IRLMAU  IRLM ID of auth SS
     136     (88) SIGNED           2   APQHP_RCVCTR  Recovery needed count
     138     (8A) SIGNED           2   APQHP_ICCTR   IC needed count
     140     (8C) UNSIGNED         1   APQHP_SHRLVL  Share level of DB
     141     (8D) UNSIGNED         1   APQHP_HELDAU  Held auth state
                  1... ....            APQHP_HAUBIT  High order bit flag
     142     (8E) UNSIGNED         2   APQHP_DMBNUM  Global DMB number
     144     (90) SIGNED           2   APQHP_SSNUM   # of SS auth DB
     146     (92) UNSIGNED         2   APQHP_SSENTLEN  Length of each SS entry
     148     (94) UNSIGNED         1   APQHP_CACCSS  Access state for chg auth
     149     (95) UNSIGNED         1   APQHP_CANCDD  Encode state for chg auth
     150     (96) UNSIGNED         1   APQHP_CAHELD  Held state for chg auth
     151     (97) CHARACTER        1   *             Reserved
     152     (98) UNSIGNED         2   APQHP_EQECNT  Total EQE count
     154     (9A) BIT(16)          2   APQHP_RSRFLG  Flags
                  1... ....            APQHP_RCVTRK  Only recov level trackng
                  .1.. ....            APQHP_TRKSPN  Tracking is suspended
                  ..1. ....            APQHP_PURBIT  Suspended by time
                  ...1 ....            APQHP_ICNDIS  IC needed disabled option
                  .... 1...            APQHP_NOHKEY  High key required
     156     (9C) CHARACTER        8   APQHP_GSGNAME GSG name
     164     (A4) UNSIGNED         4   APQHP_USID    Last alloc USID
     168     (A8) UNSIGNED         4   APQHP_AUSID   Last authorized USID
     172     (AC) UNSIGNED         4   APQHP_RUSID   Last received USID
     176     (B0) UNSIGNED         4   APQHP_HUSID   Hardened by tracker USID
     180     (B4) UNSIGNED         4   APQHP_RNUSID  Receive needed USID
     184     (B8) SIGNED           2   APQHP_ICRECCTR  IC Recommended Counter
     186     (BA) UNSIGNED         2   APQHP_VERSION#  Version number
     188     (BC) UNSIGNED         1   APQHP_OLRACTHARDCTR
                                                     OLR curs active count
     189     (BD) UNSIGNED         1   APQHP_OLRINACTHARDCTR
                                                     OLR curs inact count
     190     (BE) UNSIGNED         2   APQHP_REORG#  Partition reorg #

  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE        2  APQHP_DSGINFO  Data set group information
       0      (0) UNSIGNED         2   APQHP_BLKSZ   DS block size, OSAM only

  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE        *  APQHP_PSTRING  Partition Key/String

CONSTANTS

  LEN  TYPE           VALUE          NAME            DESCRIPTION
=====  =========  =================  =============== ===============================
    8  CHARACTER   DSPAPQHP          APQHP_EYECATCHER
```

Figure 28. DSECT of DSPAPQHP (Part 2 of 2)

DBDS Output:

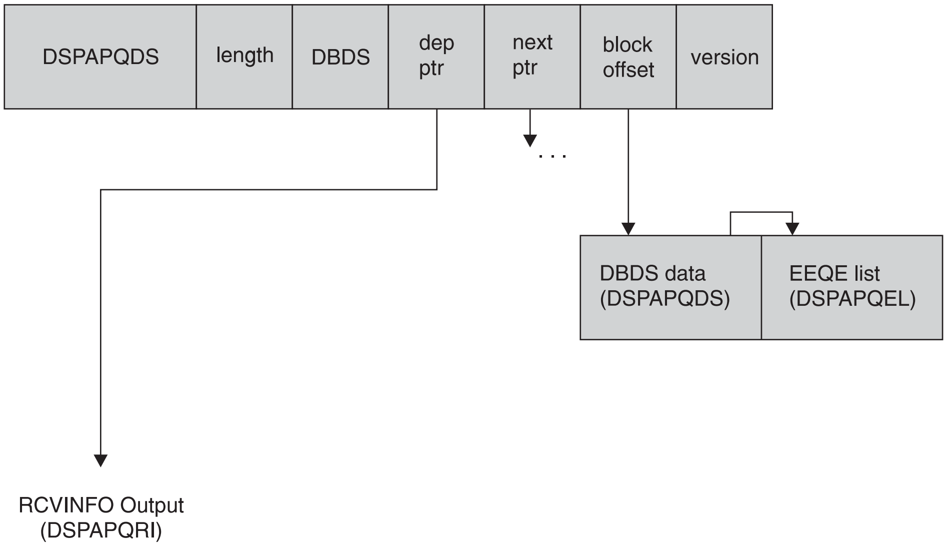


Figure 29. Format of QUERY TYPE=DB (DBDS) Output

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.

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ======== ========  ============== ==============================
       0      (0) STRUCTURE      160  DSPAPQDS
       0      (0) CHARACTER        8   APQDS_DBNAME  Database name
       8      (8) CHARACTER        8   APQDS_DDNAME  DD name
      16     (10) UNSIGNED         4   APQDS_EEQELIST
                                                     Offset to EEQE list
                                                     (DSPAPQEL), zero if no EEQEs
      20     (14) SIGNED           4   *(3)          Reserved
      32     (20) CHARACTER       44   APQDS_DSN     Data set name
      76     (4C) UNSIGNED         2   APQDS_RTPRD   IC retention period
      78     (4E) UNSIGNED         2   APQDS_DSID    Data set ID number
      80     (50) UNSIGNED         4   APQDS_DSSN    Data set sequence number
      84     (54) UNSIGNED         4   APQDS_RUSID   Recovered-to USID(trkr)
      88     (58) BIT(8)           1   APQDS_FLAGS   BINARY ZEROS
                  1... ....             APQDS_RECYC  REUSE image copies
                  .1.. ....             APQDS_ICREC  Image Copy Recommended
                  ..1. ....             APQDS_RCVRQ  Receive required
                  ...1 ....             APQDS_IC     Image Copy Needed
                  .... 1...             APQDS_RECOV  Recovery Needed
                  .... .1..             APQDS_NONRV  Non-recoverable
      89     (59) BIT(8)           1   APQDS_DSORG   Data set organization
                  1... ....             APQDS_VSAM   1 = VSAM, 0 = NON-VSAM
                  .1.. ....             APQDS_INDEX  0 = Non-indexed (OSAM or ESDS),
                                                     1 = Indexed(ISAM or KSDS)
                  ..11 1111             *            Reserved - zeroes
      90     (5A) CHARACTER        1   APQDS_DBORG   IMS DB organization
      91     (5B) UNSIGNED         1   *             Reserved
      92     (5C) UNSIGNED         2   APQDS_GENMX   Max number of ICs that may be
                                                     predefined for this area
      94     (5E) UNSIGNED         2   APQDS_AVAILIC#
                                                     Number of available ICs for
                                                     this area
      96     (60) UNSIGNED         2   APQDS_USEDIC# Number of ICs used
      98     (62) SIGNED           2   APQDS_EEQECOUNT
                                                     EEQE count
     100     (64) UNSIGNED         2   APQDS_EEQELENGTH
                                                     EEQE entry length
     102     (66) BIT(8)           1   APQDS_FLG1    Flags
                  1... ....             APQDS_RRGAL  REORG since last ALLOC, only
                                                     set if RSR-covered
                  .1.. ....             APQDS_TSRAL  TS recov since last ALLOC, only
                                                     set if RSR-covered
     103     (67) BIT(8)           1   APQDS_FLG2    DBDS type flags
                  1... ....             APQDS_PART   TYPEPART record
                  .1.. ....             APQDS_PDATA  TYPEPART subtype DATA
                  ..1. ....             APQDS_PILE   TYPEPART subtype ILE
                  ...1 ....             APQDS_PINDX  TYPEPART subtype Index
     104     (68) CHARACTER        8   APQDS_CAGRPNAME
                                                     Change Accum group name
     112     (70) CHARACTER       40   APQDS_JCL     GENJCL members
     112     (70) CHARACTER        8    APQDS_ICJCL  Image copy member
     120     (78) CHARACTER        8    APQDS_OIJCL  Online IC member
     128     (80) CHARACTER        8    APQDS_RCJCL  Recovery member
     136     (88) CHARACTER        8    APQDS_DFJCL  DEFLTJCL member
     144     (90) CHARACTER        8    APQDS_RVJCL  Receive JCL member
     152     (98) CHARACTER        8   APQDS_ODDN    OLR partner DBDS

CONSTANTS

  LEN  TYPE           VALUE          NAME            DESCRIPTION
=====  =========  =================  ============== ==============================
    8  CHARACTER   DSPAPQDS          APQDS_EYECATCHER
```

Figure 30. DSECT of DSPAPQDS

Recovery Information (RCVINFO) Output:

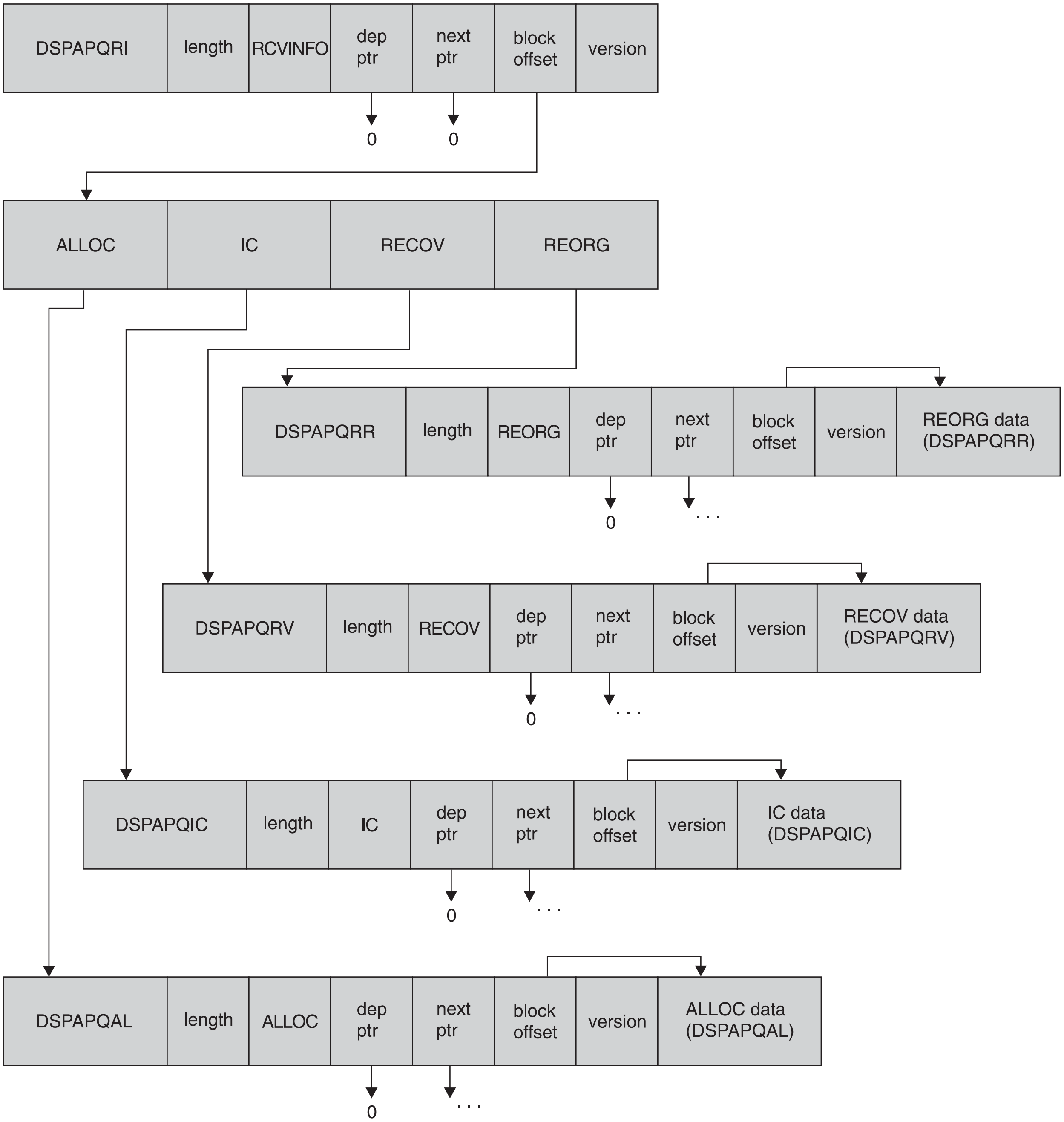


Figure 31. Format of QUERY TYPE=DB (RCVINFO) Output

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 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.

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ======== ========  ============== ===============================
       0      (0) STRUCTURE       32  DSPAPQRI
       0      (0) CHARACTER        8   APQRI_DBNAME  Database name
       8      (8) CHARACTER        8   *
       8      (8) CHARACTER        8    APQRI_DDNAME DD name or
       8      (8) CHARACTER        8    APQRI_AREANAME
                                                     Area name
      16     (10) ADDRESS          4   APQRI_ALLOCPTR
                                                     ptr to ALLOC chain
      20     (14) ADDRESS          4   APQRI_ICPTR   ptr to IC chain
      24     (18) ADDRESS          4   APQRI_RECOVPTR
                                                     ptr to RECOV chain
      28     (1C) ADDRESS          4   APQRI_REORGPTR
                                                     ptr to REORG chain

CONSTANTS

  LEN  TYPE           VALUE          NAME            DESCRIPTION
=====  =========  =================  ============== ===============================
    8  CHARACTER   DSPAPQRI          APQRI_EYECATCHER
```

Figure 32. DSECT of DSPAPQRI

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ======== ========  ============== ===============================
       0      (0) STRUCTURE       88  DSPAPQAL
       0      (0) CHARACTER        8  APQAL_DBNAME  Database name
       8      (8) CHARACTER        8  *
       8      (8) CHARACTER        8  APQAL_DDNAME  DD name or
       8      (8) CHARACTER        8  APQAL_AREANAME Area name
      16     (10) CHARACTER       12  APQAL_ALLOCTM Allocation time
      28     (1C) CHARACTER       12  APQAL_DALTM   Deallocation time
      40     (28) CHARACTER       12  APQAL_STRTM   Log start time
      52     (34) UNSIGNED         4  APQAL_DSSN    Field for DSSN value
      56     (38) UNSIGNED         4  APQAL_USID    Update set identifier
      60     (3C) CHARACTER        8  APQAL_ALRID   LRID of begin-upd rec
      68     (44) CHARACTER        8  APQAL_DLRID   LRID of end-upd rec
      76     (4C) CHARACTER        8  APQAL_SLRID   Last LRID applied if suspended
      84     (54) BIT(8)           1  APQAL_FLAGS   Flags
                  1... ....           APQAL_TSUSP  Tracking is suspended
                  .1.. ....           APQAL_NAPPL  No records applied
                  ..1. ....           APQAL_CICPT  Fuzzy ic purge time
      85     (55) CHARACTER        3  *             Reserved
CONSTANTS

  LEN  TYPE           VALUE          NAME            DESCRIPTION
=====  =========  =================  ============== ===============================
    8  CHARACTER   DSPAPQAL          APQAL_EYECATCHER
```

Figure 33. DSECT of DSPAPQAL

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE       64  DSPAPQIC
       0      (0) CHARACTER        8   APQIC_DBNAME  Database name
       8      (8) CHARACTER        8   *
       8      (8) CHARACTER        8    APQIC_DDNAME DD name or
       8      (8) CHARACTER        8    APQIC_AREANAME
                                                     Area name
      16     (10) CHARACTER       12   APQIC_STARTIME
                                                     IC start time, packed decimal
      28     (1C) CHARACTER       12   APQIC_STOPTIME
                                                     IC stop time, packed decimal
      40     (28) BIT(8)           1   APQIC_TYPE    IMAGE COPY TYPE
                  1... ....             APQIC_BATCH  BATCH
                  .1.. ....             APQIC_CIC    CONCURRENT
                  ..1. ....             APQIC_USERIC USER IMAGE COPY
                  ...1 ....             APQIC_ONLINE ONLINE
                  .... 1...             APQIC_SMSIC  SMS IC w/ DB exclusive
                  .... .1..             APQIC_SMSCC  SMS IC w/ DB shared
      41     (29) BIT(8)           1   APQIC_STATUS  IC status flags
                  1... ....             APQIC_AVAIL  Available IC
                  .1.. ....             APQIC_IC1    Image Copy 1 exists
                  ..1. ....             APQIC_IC2    Image Copy 2 exists
                  ...1 ....             APQIC_ERR1   Error on image 1
                  .... 1...             APQIC_ERR2   Error on image 2
                  .... .1..             APQIC_EMP2   Image 2 defined and unused
      42     (2A) BIT(8)           1   APQIC_FLAGS
                  1... ....             APQIC_HSINP  HSSP CIC in progress
                  .1.. ....             APQIC_CAT    Catalogued IC (HSSP)
      43     (2B) CHARACTER        1   *             Reserved
      44     (2C) CHARACTER        2   *
      44     (2C) UNSIGNED         2    APQIC_OFF1   Offset to image 1 data
      44     (2C) UNSIGNED         2    APQIC_OFFU   Offset to user IC data
      46     (2E) UNSIGNED         2   APQIC_OFF2    Offset to image 2 data
      48     (30) UNSIGNED         4   APQIC_CNT12   Record count
      52     (34) UNSIGNED         4   APQIC_USID    Update set ID
      56     (38) CHARACTER        2   *
      56     (38) UNSIGNED         2    APQIC_LEN12  Length of image 1/2 data
      56     (38) UNSIGNED         2    APQIC_LENU   Length of user IC data
      58     (3A) CHARACTER        6   *             Reserved
```

Figure 34. DSECT of DSPAPQIC (Part 1 of 2)

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ======== ========  ============== ===============================
       0      (0) STRUCTURE       64  APQIC_IC12     Data for image 1 or 2
       0      (0) CHARACTER       44   APQIC_DSN12   Data set name
      44     (2C) UNSIGNED         2   APQIC_FILE    File sequence number
      46     (2E) CHARACTER        8   APQIC_RUT12   Unit device type
      54     (36) UNSIGNED         2   APQIC_VOLCT   # of volumes predefined
      56     (38) UNSIGNED         2   APQIC_VOLUS   # of volumes used
      58     (3A) UNSIGNED         2   APQIC_VOLLISTLEN
                                                     Length of each volume list
                                                     entry in apqic_VOLS
      60     (3C) UNSIGNED         4   APQIC_VOLLISTOFFSET
                                                     Offset to volume list

  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ======== ========  ============== ===============================
       0      (0) CHARACTER        6   APQIC_VOLS    List of VOLSERs

  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ======== ========  ============== ===============================
       0      (0) STRUCTURE       80  APQIC_USER     Data for user IC
       0      (0) CHARACTER       80   APQIC_UDATA   User supplied data

CONSTANTS

  LEN  TYPE           VALUE          NAME           DESCRIPTION
=====  =========  ================  ============== ===============================
    1  DECIMAL           255         APQIC_MAXV     max # volumes
    8  CHARACTER   DSPAPQIC          APQIC_EYECATCHER
```

Figure 34. DSECT of DSPAPQIC (Part 2 of 2)

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ======== ========  ============== ===============================
       0      (0) STRUCTURE       49  DSPAPQRV
       0      (0) CHARACTER        8   APQRV_DBNAME  Database name
       8      (8) CHARACTER        8   *
       8      (8) CHARACTER        8    APQRV_DDNAME DD name or
       8      (8) CHARACTER        8    APQRV_AREANAME
                                                     Area name
      16     (10) CHARACTER       12   APQRV_RUNTIME The time at which the DBDS was
                                                     recovered.
      28     (1C) CHARACTER       12   APQRV_ENDTIME Partial recovery only, the time
                                                     to which the DBDS was restored
      40     (28) UNSIGNED         4   APQRV_FUSID   First undone USID
      44     (2C) UNSIGNED         4   APQRV_LUSID   Last undone USID
      48     (30) BIT(8)           1   APQRV_FLAGS   Flags
                  1... ....             APQRV_PITR   Point In Time Recovery

CONSTANTS

  LEN  TYPE           VALUE          NAME           DESCRIPTION
=====  ========   ================  ============== ===============================
    8  CHARACTER   DSPAPQRV          APQRV_EYECATCHER
```

Figure 35. DSECT of DSPAPQRV

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE       60  DSPAPQRR
       0      (0) CHARACTER        8   APQRR_DBNAME  Database name
       8      (8) CHARACTER        8   *
       8      (8) CHARACTER        8    APQRR_DDNAME DD name or
       8      (8) CHARACTER        8    APQRR_AREANAME
                                                     Area name
      16     (10) CHARACTER       12   APQRR_RUNTIME The time at which the DBDS was
                                                     reorganized.
      28     (1C) CHARACTER       12   APQRR_STOPTIME
                                                     Stoptime of online reorg
      40     (28) BIT(8)           1   APQRR_FLAGS
                  1... ....             APQRR_ONL    1=ONLINE/0=OFFLINE reorg
                  .1.. ....             APQRR_RECOV  1=May be used for recovery
      41     (29) CHARACTER        3   *             Reserved
      44     (2C) UNSIGNED         4   APQRR_USID    Associated USID
      48     (30) CHARACTER       12   APQRR_PITR    Stoptime moved - PITR

CONSTANTS

  LEN  TYPE           VALUE          NAME           DESCRIPTION
=====  =========  =================  ============== ===============================
    8  CHARACTER   DSPAPQRR          APQRR_EYECATCHER
```

Figure 36. DSECT of DSPAPQRR

Database Not Found Output:

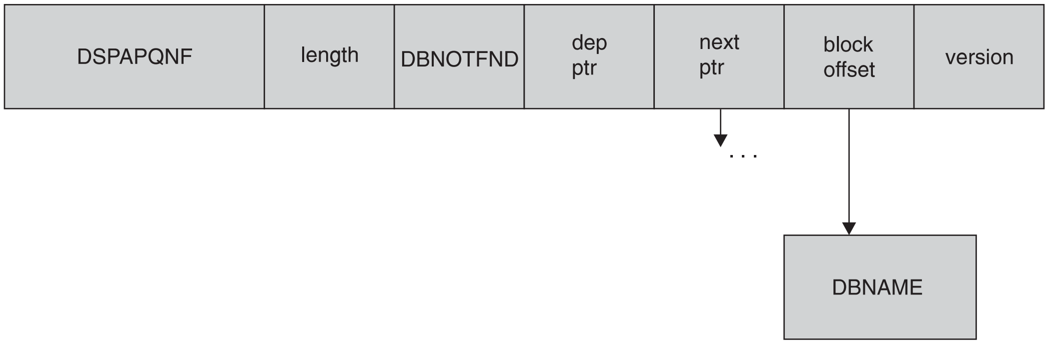


Figure 37. Format of QUERY TYPE=DB Database Not Found Output

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.

```
OFFSET   OFFSET
DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ======== ========  ============= ==============================
       0      (0) STRUCTURE       8   DSPAPQNF
       0      (0) CHARACTER       8   APQNF_DBNAME  DB name
```

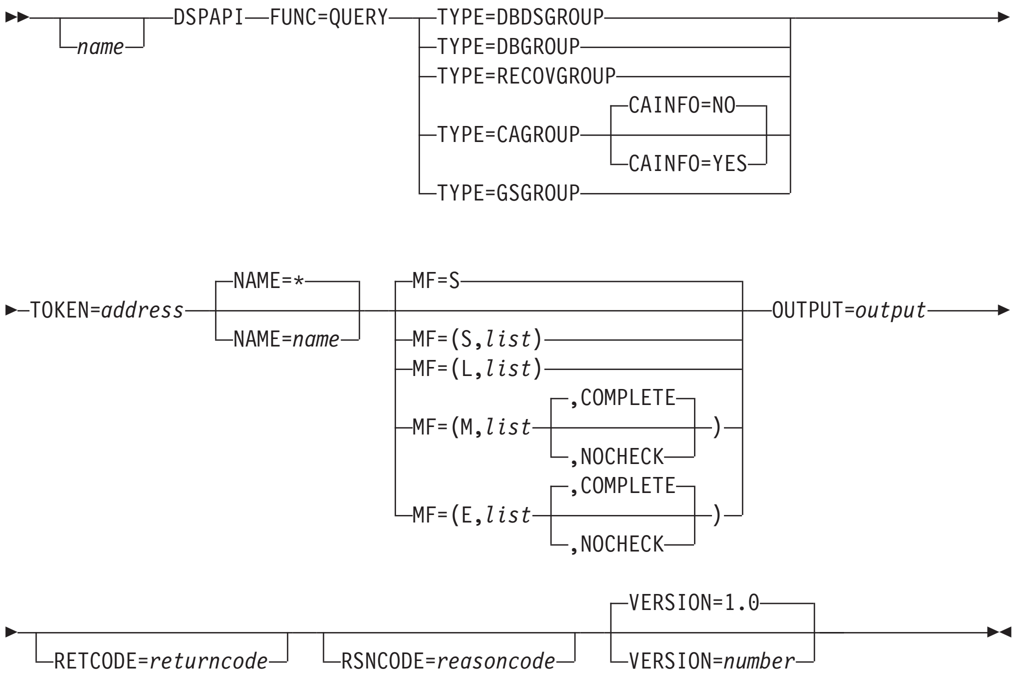
Figure 38. DSECT of DSPAPQNF

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

TYPE=CAGROUP. CAINFO defaults to NO when TYPE=CAGROUP is specified, indicating that only the CA group member information is requested.

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) indicates a request for 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.

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=xxxxGROUP Query Request

Table 21 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 Version 9: Diagnosis Guide and Reference*.

Table 21. Return and Reason Codes for TYPE=xxxxGROUP Query Requests

Return Codes	Reason Codes	Meaning
X'00000000'	X'00000000'	Request completed successfully.
Warning		
X'00000008'	X'D8300001'	No group records of the requested type exist in the RECON.
Severe Error		
X'0000000C'	X'C9000001'	Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.
Storage Error		
X'00000028'	X'D8300001'	Error obtaining storage for CAGROUP block.
X'00000028'	X'D8300002'	Error obtaining storage for GSG block.
X'00000028'	X'D8300003'	Error obtaining storage for DBDSGRP, DBGRP, or RECOVGRP block.
X'00000028'	X'D8310001'	Error obtaining storage for CA block.
X'00000028'	X'D9100001'	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.
Internal Error		
X'0000002C'	all	DBRC internal error. Refer to “Database Recovery Control Service Aids” in the *IMS Version 9: Diagnosis Guide and Reference* for further information.
Parameter Error		
X'00000030'	X'C9000001'	Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.
X'00000030'	X'C9000002'	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.
X'00000030'	X'C9000003'	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.
X'00000030'	X'C9000004'	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.
X'00000030'	X'C9000005'	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.

Table 21. Return and Reason Codes for TYPE=xxxxGROUP Query Requests (continued)

Return Codes	Reason Codes	Meaning
X'00000030'	X'C9000008'	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.
X'00000030'	X'D8000001'	Parameter error. Missing or invalid OUTPUT parameter.
X'00000030'	X'D8300001'	Parameter error. CAINFO=YES is only allowed with TYPE=CAGROUP.

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:

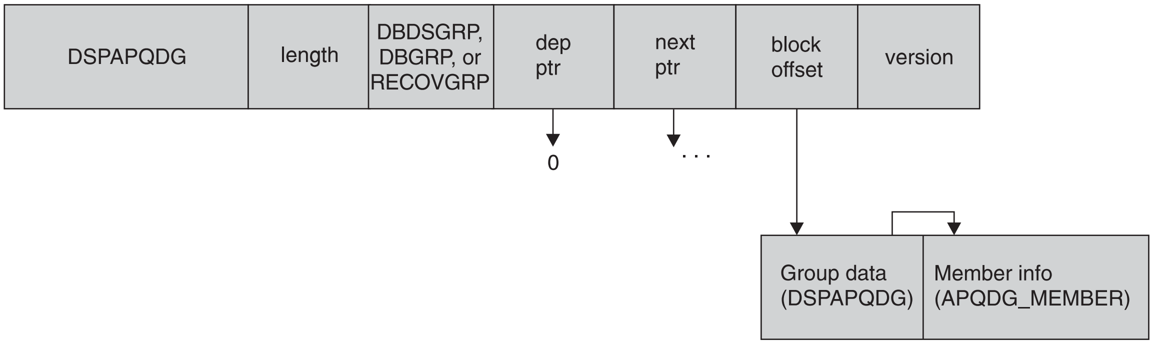


Figure 39. Format for QUERY TYPE=DBDSGROUP, DBGROUP, RECOVGROUP Output

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE       32  DSPAPQDG
       0      (0) CHARACTER        8   APQDG_GROUPNAME
                                                     Group name
       8      (8) UNSIGNED         4   APQDG_MEMBERINFO
                                                     Offset to group member list
      12      (C) UNSIGNED         2   APQDG_MEMBERLEN
                                                     Length of group member entry
      14      (E) SIGNED           2   APQDG_MEMBERCOUNT
                                                     Number of group members
      16     (10) SIGNED           4   *(3)          Reserved
      28     (1C) CHARACTER        4   *             Reserved

  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE       16  APQDG_MEMBER   List of group members
       0      (0) CHARACTER       16   *
       0      (0) CHARACTER       16    APQDG_DBDSG  DBDS group
       0      (0) CHARACTER        8     APQDG_DBDSG_DBNAME
                                                     Database name
       8      (8) CHARACTER        8     *
       8      (8) CHARACTER        8      APQDG_DBDSG_DDNAME
                                                     DD name or
       8      (8) CHARACTER        8      APQDG_DBDSG_AREANAME
                                                     AREA name
       0      (0) CHARACTER       16    APQDG_DBG    DB group
       0      (0) CHARACTER        8     *
       0      (0) CHARACTER        8      APQDG_DBG_DBNAME
                                                     Database name or
       0      (0) CHARACTER        8      APQDG_DBG_AREANAME
                                                     AREA name
       8      (8) CHARACTER        8     *           Not used
       0      (0) CHARACTER       16    APQDG_RECOVG Recovery group
       0      (0) CHARACTER        8     APQDG_RECOVG_DBNAME
                                                     Database name
       8      (8) CHARACTER        8     APQDG_RECOVG_AREANAME
                                                     AREA name, null if
                                                     not Fast Path

CONSTANTS

  LEN  TYPE           VALUE          NAME            DESCRIPTION
=====  =========  =================  =============== ===============================
    8  CHARACTER   DSPAPQDG          APQDG_EYECATCHER
```

Figure 40. DSECT of DSPAPQDG

Output for QUERY TYPE=CAGROUP:

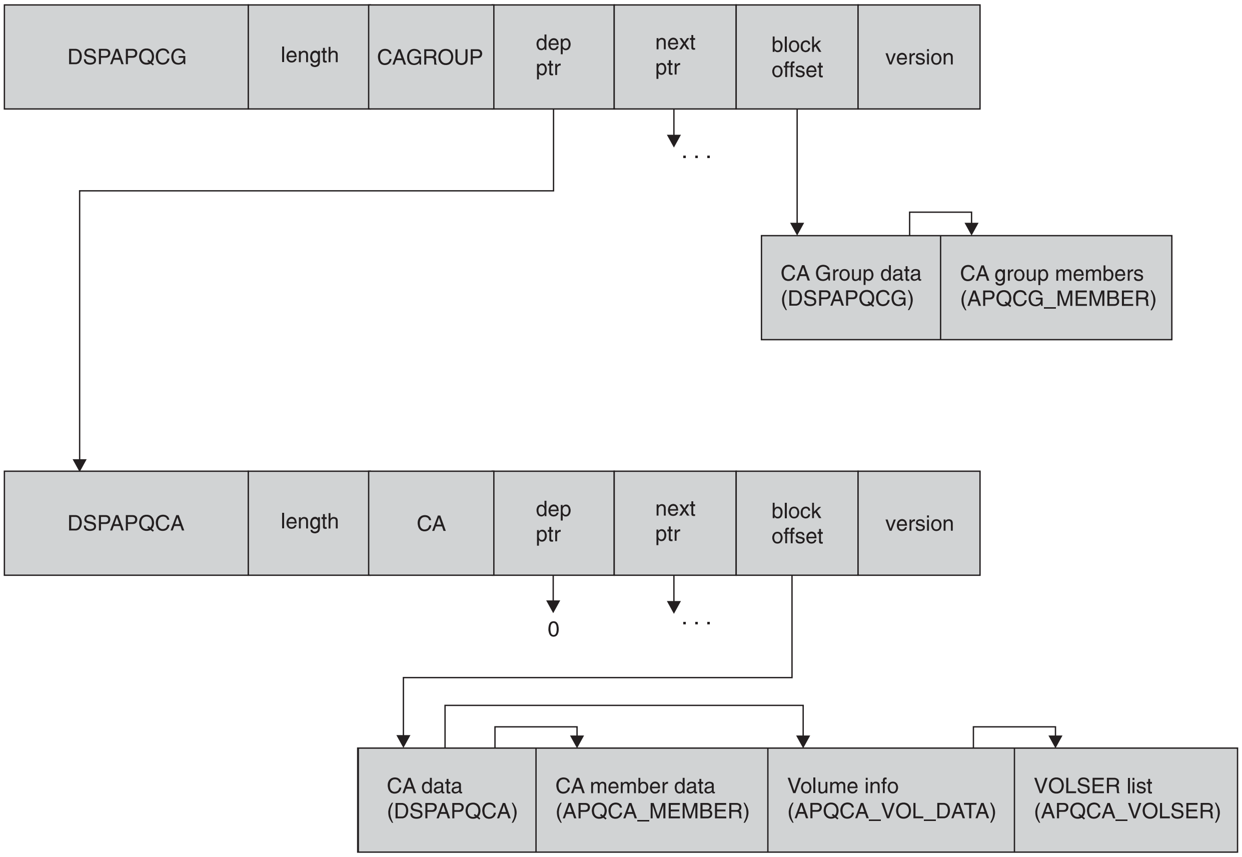


Figure 41. Format for QUERY TYPE=CAGROUP Output

The CA block is only returned when CAINFO=YES is specified and records of a change accumulation exist in the RECON.

```
  OFFSET   OFFSET
 DECIMAL      HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE       48  DSPAPQCG
       0      (0) CHARACTER        8   APQCG_GROUPNAME
                                                     Group name
       8      (8) UNSIGNED         4   APQCG_MEMBERINFO
                                                     Offset to group member list
      12      (C) UNSIGNED         2   APQCG_MEMBERLEN
                                                     Length of group member entry
      14      (E) SIGNED           2   APQCG_MEMBERCOUNT
                                                     Number of group members
      16     (10) SIGNED           4   *(2)          Reserved
      24     (18) SIGNED           2   APQCG_GRPMAX  Maximum number of CAs that may
                                                     be predefined for this CA group
      26     (1A) SIGNED           2   APQCG_AVAILCA#
                                                     Number of available CA data
                                                     sets for this group
      28     (1C) SIGNED           2   APQCG_USEDCA# Number of used CA data sets
      30     (1E) BIT(8)           1   APQCG_FLAGS   Flags
                  1... ....             APQCG_REUSE  Reuse CA data sets
      31     (1F) CHARACTER        1   *             Reserved
      32     (20) CHARACTER        8   APQCG_CAJCL   GENJCL CAJCL member name
      40     (28) CHARACTER        8   APQCG_DFJCL   DEFLTJCL member name

  OFFSET   OFFSET
 DECIMAL      HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE       16  APQCG_MEMBER   List of group members
       0      (0) CHARACTER        8   APQCG_DBNAME  Database name
       8      (8) CHARACTER        8   *
       8      (8) CHARACTER        8    APQCG_DDNAME DD name or
       8      (8) CHARACTER        8    APQCG_AREANAME

                                                    AREA name

CONSTANTS

  LEN  TYPE           VALUE          NAME            DESCRIPTION
=====  =========  =================  =============== ===============================
    8  CHARACTER   DSPAPQCG          APQCG_EYECATCHER
```

Figure 42. DSECT of DSPAPQCG

Output for QUERY TYPE=GSGROUP:

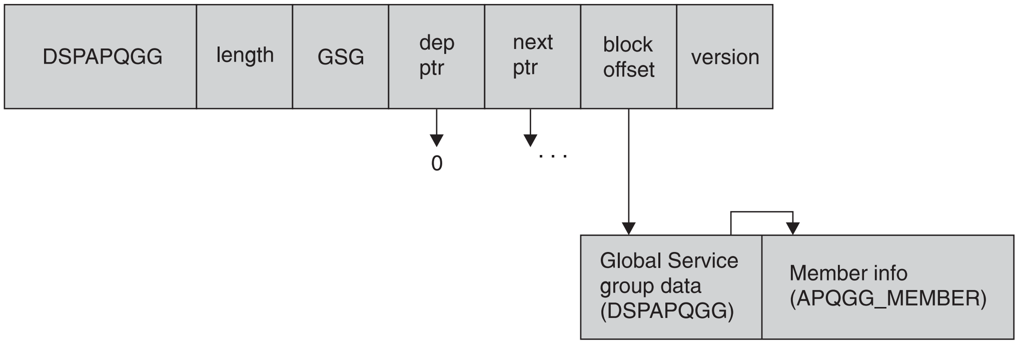

DSPAPQGG
length
GSG
dep ptr
next ptr
block offset
version
0
. . .
Global Service group data (DSPAPQGG)
Member info (APQGG_MEMBER)

Figure 43. Format for QUERY TYPE=GSGROUP Output

```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE       80  DSPAPQGG
       0      (0) CHARACTER        8   APQGG_GROUPNAME
                                                     Group name
       8      (8) UNSIGNED         4   APQGG_MEMBERINFO
                                                     Offset to group member list
      12      (C) UNSIGNED         2   APQGG_MEMBERLEN
                                                     Length of group member entry
      14      (E) SIGNED           2   APQGG_MEMBERCOUNT
                                                     Number of group members
      16     (10) SIGNED           4   *(2)          Reserved
      24     (18) UNSIGNED         4   APQGG_SDSN#   SLDS DSN sequence number
      28     (1C) UNSIGNED         4   APQGG_PTOKEN  Current PRILOG token
      32     (20) UNSIGNED         4   APQGG_MPTOK   Min required PRILOG token
      36     (24) UNSIGNED         4   APQGG_TTOKN   Planned takeover token
      40     (28) CHARACTER       12   APQGG_LOGTIME Current log start time
      52     (34) CHARACTER       12   APQGG_HITIME  Highest time ever received from
                                                     the active site
      64     (40) CHARACTER        8   APQGG_TSNAME  Tracking subsystem ID
      72     (48) BIT(8)           1   APQGG_FLAGS
                  1... ....             APQGG_ACTTKO Active takeover in progress
                  .1.. ....             APQGG_1STSIGNON
                                                     Indicates the first signon of
                                                     an active subsystem after an
                                                     RSR takeover
                  ..1. ....             APQGG_TRKTKO Tracker takeover in progress
                  ...1 ....             APQGG_RESET  RESET.GSG issued
      73     (49) CHARACTER        7   *             Reserved

  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE       16  APQGG_MEMBER   List of group members
       0      (0) CHARACTER        8   APQGG_SGNAME  Service group name
       8      (8) CHARACTER        1   APQGG_SGROLE  Role 'A' OR 'T'
       9      (9) BIT(8)           1   APQGG_SGFLAGS Flags
                  1... ....             APQGG_SGLOCAL
                                                     1 = local SG
      10      (A) CHARACTER        6   *             Reserved

CONSTANTS

  LEN  TYPE           VALUE          NAME           DESCRIPTION
=====  =========  =================  ============== ===============================
    8  CHARACTER   DSPAPQGG          APQGG_EYECATCHER
```

Figure 44. DSECT of DSPAPQGG

Log Query

The Log query (DSPAPI FUNC=QUERY TYPE=LOG) request is used to retrieve log information from RECON for a specific instance of a subsystem.

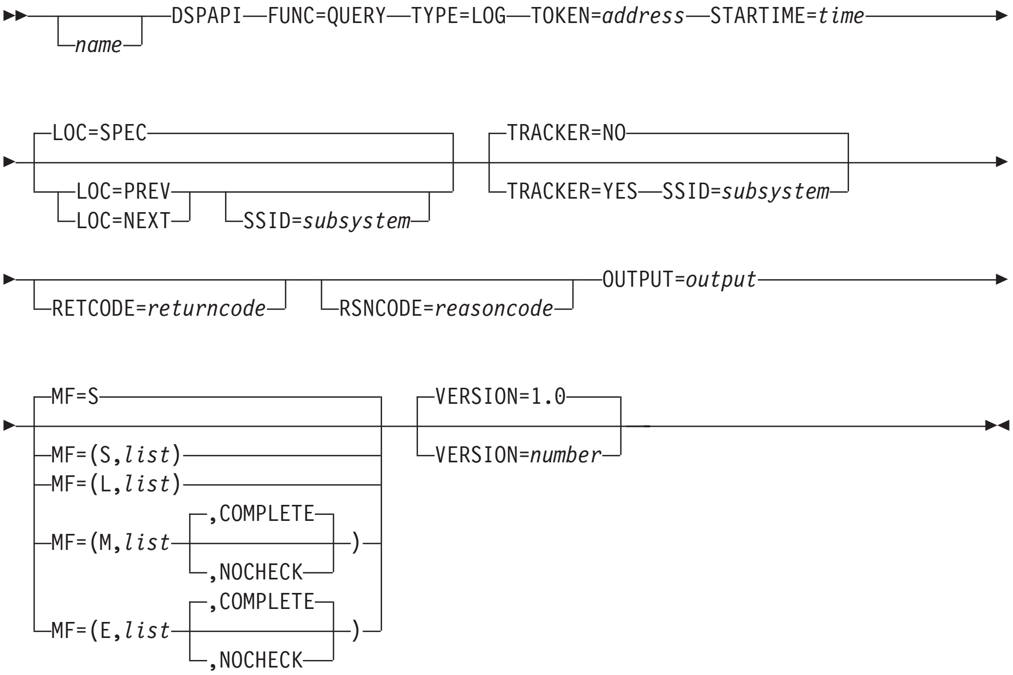
If the request is for non-RSR logs or RSR active subsystem logs, information from the following RECON records is returned:

- 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



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.

LOC=SPEC is the default.

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 397 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.

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 parameter when you request the previous or next log of a specific subsystem, that is, you specify LOC=PREV or LOC=NEXT.

Return and Reason Codes for the TYPE=LOG Query Request

Table 22 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 Version 9: Diagnosis Guide and Reference*.

Table 22. Return and Reason Codes for TYPE=LOG Query Requests

Return Codes	Reason Codes	Meaning
X'00000000'	X'00000000'	Request completed successfully.
	Warning	
X'00000008'	X'D8400001'	No previous or next log record of the requested log type - PRILOG or PRITSLDS.
X'00000008'	X'D8400002'	The specified log record of the requested log type - PRILOG or PRITSLDS - does not exist.
	Severe Error	
X'0000000C'	X'C9000001'	Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.
	Storage Error	
X'00000028'	X'D8400001'	Error obtaining storage for LOGINFO block.
X'00000028'	X'D8400002'	Error obtaining storage for PRILOG, SECLOG, PRISLDS, SECSLDS, PRITSLDS, or SECTSLDS block.
X'00000028'	X'D8400003'	Error obtaining storage for LOGALL block.
X'00000028'	X'D9100001'	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.
	Internal Error	
X'0000002C'	all	DBRC internal error. Refer to “Database Recovery Control Service Aids” in the *IMS Version 9: Diagnosis Guide and Reference* for further information.
	Parameter Error	
X'00000030'	X'C9000001'	Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.
X'00000030'	X'C9000001'	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.
X'00000030'	X'C9000001'	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.
X'00000030'	X'C9000001'	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.

Table 22. Return and Reason Codes for TYPE=LOG Query Requests (continued)

Return Codes	Reason Codes	Meaning
X'00000030'	X'C9000001'	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.
X'00000030'	X'C9000001'	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.
X'00000030'	X'C9000001'	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.
X'00000030'	X'D8000001'	Parameter error. Missing or invalid OUTPUT parameter.
X'00000030'	X'D8400001'	Parameter error. STARTIME parameter is required.
X'00000030'	X'D8400002'	Parameter error. SSID is required with TRACKER=YES.
X'00000030'	X'D8400003'	Parameter error. SSID parameter is not allowed for active logs (TRACKER=NO) unless specified with LOC=PREV or NEXT.

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:

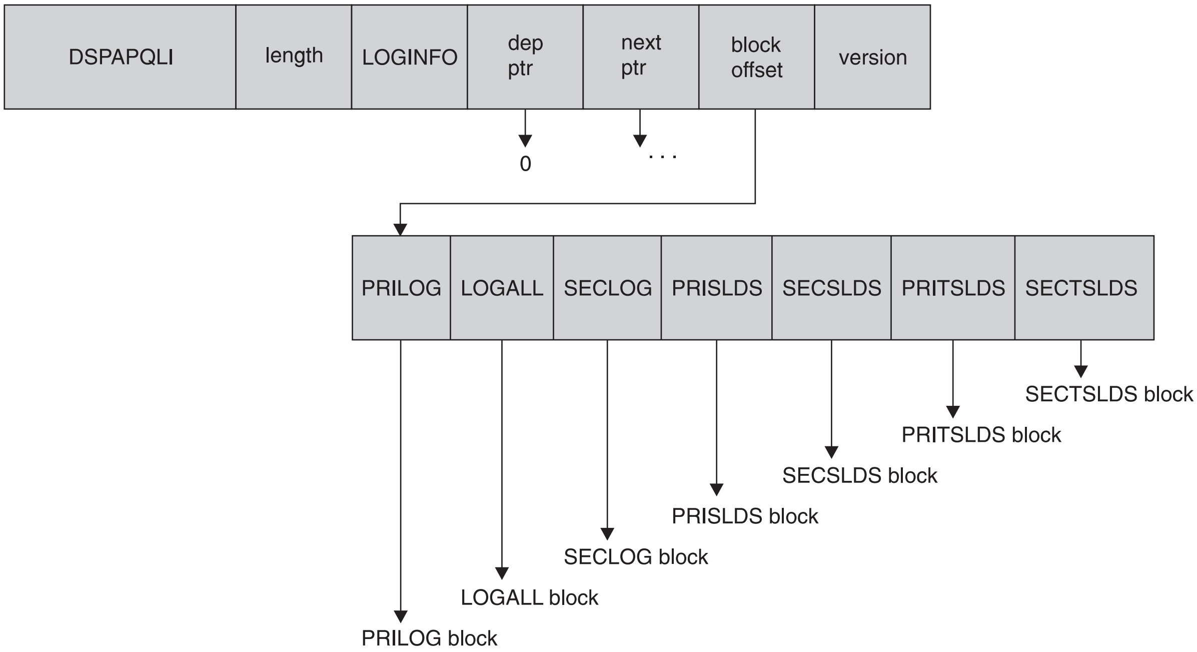


Figure 45. Format for QUERY TYPE=LOG Log Information Output

```
  OFFSET   OFFSET
 DECIMAL      HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE       48  DSPAPQLI
       0      (0) CHARACTER        8   APQLI_SSID    Log SSID
       8      (8) CHARACTER       12   APQLI_STARTTIME
                                                     Log start time
      20     (14) ADDRESS          4   APQLI_PRILOGPTR
                                                     ptr to PRILOG block
      24     (18) ADDRESS          4   APQLI_LOGALLPTR
                                                     ptr to LOGALL block
      28     (1C) ADDRESS          4   APQLI_SECLOGPTR
                                                     ptr to SECLOG block
      32     (20) ADDRESS          4   APQLI_PRISLDSPTR
                                                     ptr to PRISLDS block
      36     (24) ADDRESS          4   APQLI_SECSLDSPTR
                                                     ptr to SECSLDS block
      40     (28) ADDRESS          4   APQLI_PRITSLDSPTR
                                                     ptr to PRITSLDS block
      44     (2C) ADDRESS          4   APQLI_SECTSLDSPTR
                                                     ptr to SECTSLDS block

CONSTANTS

  LEN  TYPE           VALUE            NAME           DESCRIPTION
=====  =========  =================  =============== ===============================
    8  CHARACTER   DSPAPQLI          APQLI_EYECATCHER
```

Figure 46. DSECT of DSPAPQLI

TYPE=LOG Output for PRILOG, SECLOG, PRISLDS, SECSLDS, PRITSLDS, and SECTSLDS:

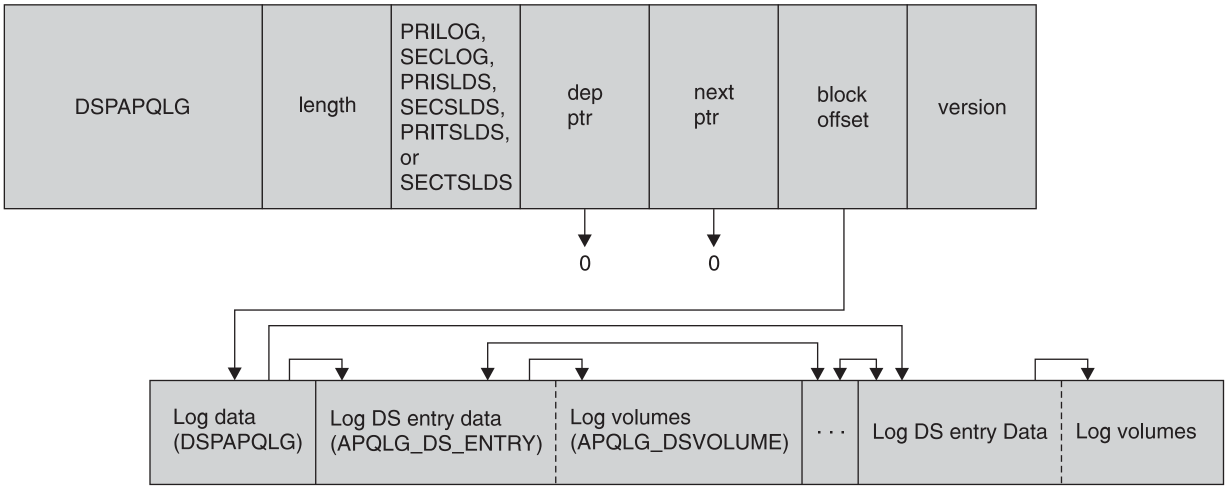

DSPAPQLG
length
PRILOG, SECLOG, PRISLDS, SECSLDS, PRITSLDS, or SECTSLDS
dep ptr
next ptr
block offset
version
0
0
Log data (DSPAPQLG)
Log DS entry data (APQLG_DS_ENTRY)
Log volumes (APQLG_DSVOLUME)
. . .
Log DS entry Data
Log volumes

Figure 47. Format for QUERY TYPE=LOG Output for PRILOG, SECLOG, PRISLDS, SECSLDS, PRITSLDS, and SECTSLDS

```
  OFFSET   OFFSET
 DECIMAL      HEX  TYPE         LENGTH  NAME (DIM)          DESCRIPTION
======== ======== ========= ========   ==============      ===============================
       0      (0) STRUCTURE        96  DSPAPQLG
       0      (0) UNSIGNED          4   APQLG_FIRSTLOGDS   Offset to first log DS entry
       4      (4) UNSIGNED          4   APQLG_LASTLOGDS    Offset to last log DS entry
       8      (8) SIGNED            4   *(2)               Reserved
      16     (10) CHARACTER         8   APQLG_SSID         Subsystem ID
      24     (18) CHARACTER        12   APQLG_STARTTIME    Log start time
      36     (24) CHARACTER        12   APQLG_ENDTIME      Log end time
      48     (30) SIGNED            4   APQLG_DSNCOUNT     Number of data sets
      52     (34) UNSIGNED          1   APQLG_RELVL        Log Release Level
      53     (35) BIT(8)            1   APQLG_FLAGS1       Flags
                  1... ....              APQLG_ONLINE      Online log
                  .1.. ....              APQLG_LSTAR       Last OLDS has been archived
                  ..1. ....              APQLG_LSTNA       Last OLDS has not been archived
                  ...1 ....              APQLG_GAP         There is a gap in this log
                  .... 1...              APQLG_PRGAP       There is a gap in a prev log
      54     (36) BIT(8)            1   APQLG_FLAGS2       Flags
                  1... ....              APQLG_TRKNG       Tracking log data set
                  .1.. ....              APQLG_NTERM       IMS subsystem has terminated normally
                  ..1. ....              APQLG_BKLOG       Batch backout log
      55     (37) CHARACTER         1   *                  Reserved
      56     (38) CHARACTER         8   APQLG_FIRSTLRID    Id of first log record
      64     (40) UNSIGNED          4   APQLG_PTOKEN       PRILOG token
      68     (44) CHARACTER         8   APQLG_GSGNAME      GSG name
      76     (4C) CHARACTER        12   APQLG_CHKPT0       Checkpoint 0 time
      88     (58) CHARACTER         8   *                  Reserved
```

Figure 48. DSECT of DSPAPQLG (Part 1 of 2)

```
  OFFSET   OFFSET
 DECIMAL      HEX  TYPE        LENGTH  NAME (DIM)         DESCRIPTION
======== ======== ======== ========  ==============     ===============================
       0      (0) STRUCTURE      120  APQLG_DS_ENTRY     Data set entry
       0      (0) UNSIGNED         4   APQLG_DS_NEXT     Offset to next log DS entry
       4      (4) UNSIGNED         4   APQLG_DS_PREV     Offset to prev log DS entry
       8      (8) UNSIGNED         4   APQLG_DS_VOLINFO  Offset to DS volume info
      12      (C) CHARACTER       44   APQLG_DS_DSNAME   Data set name
      56     (38) CHARACTER       12   APQLG_DS_STARTTIME DS start time
      68     (44) CHARACTER       12   APQLG_DS_ENDTIME  DS end time
      80     (50) BIT(8)           1   APQLG_DS_FLAGS1   Flags
                  1... ....             APQLG_DS_ERR I/O Error
                  .1.. ....             APQLG_DS_DUMMY   Log compressed, 1st DS dummy
                  ..1. ....             APQLG_DS_RSTBG   Restart begin
                  ...1 ....             APQLG_DS_RSTEN   Restart end
                  .... 1...             APQLG_DS_COLD    Cold start
                  .... .1..             APQLG_DS_NOBMP   ERE NOBMP
                  .... ..1.             APQLG_DS_SAVER   Backout UORs saved
                  .... ...1             APQLG_DS_NOID    Backouts not identified
      81     (51) BIT(8)           1   APQLG_DS_FLAGS2   Flags
                  1... ....             APQLG_DS_TRKARCH Tracking log DS archived
                  .1.. ....             APQLG_DS_TRKFEOV Tracking log closed FEOV
      82     (52) CHARACTER        2   APQLG_DS_DFLG3    Reserved
      84     (54) CHARACTER        8   APQLG_DS_FLRID    First log record ID
      92     (5C) CHARACTER        8   APQLG_DS_LLRID    Last log record ID
     100     (64) UNSIGNED         4   APQLG_DS_LASTBLKSEQNO   Last block seq number
     104     (68) CHARACTER        8   APQLG_DS_UNITTYPE Unit type name
     112     (70) UNSIGNED         2   APQLG_DS_FILESEQ  File sequence no
     114     (72) UNSIGNED         2   APQLG_DS_VOLCOUNT Number of volumes
     116     (74) UNSIGNED         1   APQLG_DS_CKPTCOUNT Number of chkpts on DSN
     117     (75) BIT(8)           1   APQLG_DS_CHKPTTYPES CHKPT types.
                  1... ....             APQLG_DS_SIMPL  SIMPLE checkpoint
                  .1.. ....             APQLG_DS_SNAPQ  SNAPQ checkpoint
                  ..1. ....             APQLG_DS_DUMPQ  DUMPQ checkpoint
                  ...1 ....             APQLG_DS_PURGE  PURGE checkpoint
                  .... 1...             APQLG_DS_FREEZ  FREEZE checkpoint
     118     (76) UNSIGNED         2   *                Reserved
  OFFSET   OFFSET
 DECIMAL      HEX  TYPE        LENGTH  NAME (DIM)         DESCRIPTION
======== ======== ======== ========  ==============     ===============================
       0      (0) STRUCTURE       48  APQLG_DSVOLUME     Volume information
       0      (0) UNSIGNED         4   APQLG_DSVOL_NEXT  Offset to next volume
       4      (4) CHARACTER        6   APQLG_DSVOL_SER   VOLSER
      10      (A) UNSIGNED         1   APQLG_DSVOL_CKPTCT Volume chkpt count
      11      (B) CHARACTER        1   *                 Reserved
      12      (C) CHARACTER       12   APQLG_DSVOL_ENDTIME Volume end time
      24     (18) CHARACTER       12   APQLG_DSVOL_CPTID Checkpoint ID
      36     (24) CHARACTER        6   APQLG_DSVOL_LOCKSN Lock Sequence Number
      42     (2A) CHARACTER        6   *             Reserved

CONSTANTS

  LEN  TYPE          VALUE          NAME               DESCRIPTION
=====  =========  ================  ==============     ===============================
    8  CHARACTER   DSPAPQLG         APQLG_EYECATCHER
```

Figure 48. DSECT of DSPAPQLG (Part 2 of 2)

TYPE=LOG Output for LOGALL:

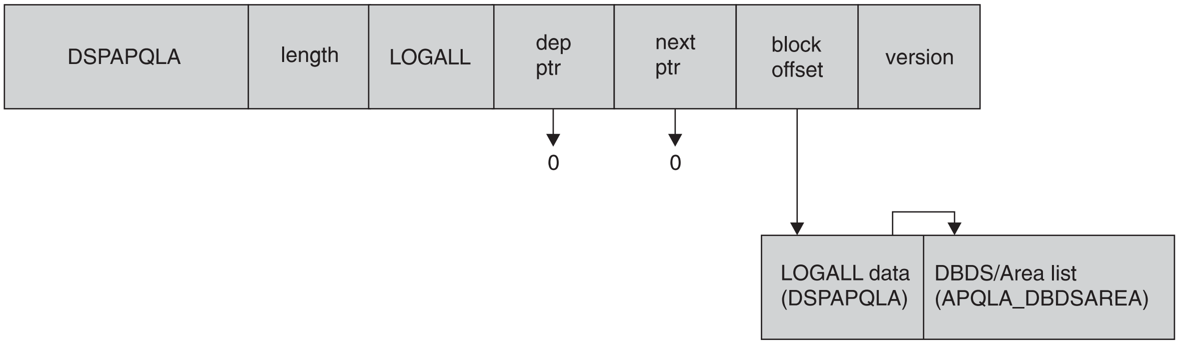


Figure 49. Format for QUERY TYPE=LOG Output for LOGALL

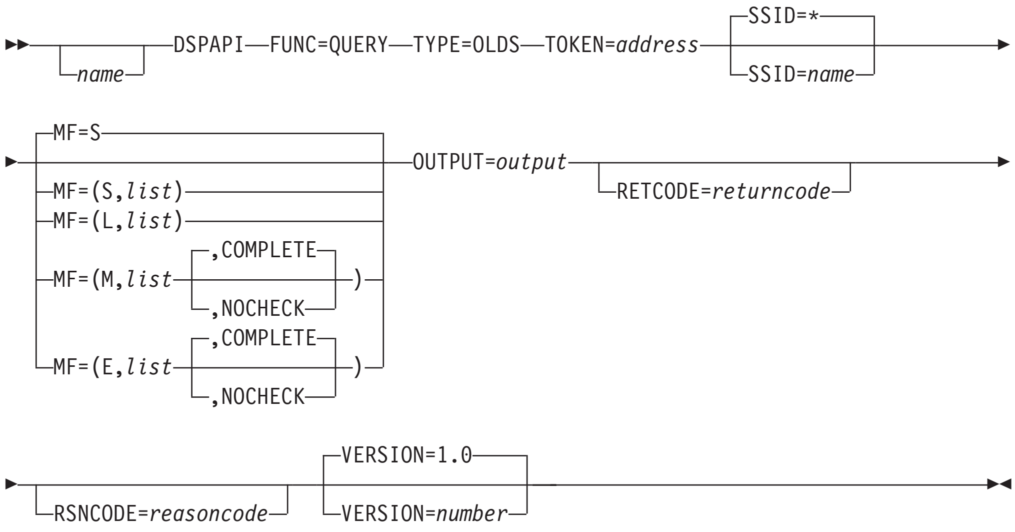
```
  OFFSET   OFFSET
 DECIMAL      HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE       48  DSPAPQLA
       0      (0) UNSIGNED         4   APQLA_DBDSAREAINFO
                                                     Offset to allocated
                                                     DBDS/Area list
       4      (4) SIGNED           4   *(3)           Reserved
      16     (10) CHARACTER       12   APQLA_PRILOGTIME
                                                     PRILOG time
      28     (1C) BIT(8)           1   APQLA_FLAGS   Flags
                  1... ....             APQLA_NONREGD
                                                     Non-registered DB updated
      29     (1D) UNSIGNED         3   APQLA_DBDSAREACOUNT
                                                     Number of DBDS/Areas
                                                     allocated on this log
      32     (20) UNSIGNED         4   APQLA_DBDSAREALEN
                                                     Length of DBDS/Area entry
      36     (24) CHARACTER       12   APQLA_EARLIESTALLOC
                                                     Earliest ALLOC time for
                                                     this log

  OFFSET   OFFSET
 DECIMAL      HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE       32  APQLA_DBDSAREA List of allocated DBDSs and
                                                     Areas
       0      (0) CHARACTER        8   APQLA_DBNAME  Database name
       8      (8) CHARACTER        8   *
       8      (8) CHARACTER        8    APQLA_DDNAME DD name or
       8      (8) CHARACTER        8    APQLA_AREANAME
                                                     AREA name
      16     (10) CHARACTER       12   APQLA_FIRSTALLOC
                                                     Earliest ALLOC time for this
                                                     DBDS/Area on this log
      28     (1C) SIGNED           2   APQLA_ALLNO   Number of ALLOCs for this
                                                     DBDS/Area on this log
      30     (1E) CHARACTER        2   *              Reserved

CONSTANTS

  LEN  TYPE           VALUE           NAME            DESCRIPTION
=====  =========  =================  =============== ===============================
    8  CHARACTER   DSPAPQLA          APQLA_EYECATCHER
```

Figure 50. DSECT of DSPAPQLA

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



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=*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 subsystem name for the backout being queried. ’*’ (the default) indicates a request for 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=*symbol* **| (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 404 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=OLDS Query Request

Table 23 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 Version 9: Diagnosis Guide and Reference*.

Table 23. Return and Reason Codes for TYPE=OLDS Query Requests

Return Codes	Reason Codes	Meaning
X'00000000'	X'00000000'	Request completed successfully.
		Warning
X'00000008'	X'D8500001'	No PRIOLDS records exist.
		Severe Error
X'0000000C'	X'C9000001'	Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.
		Storage Error
X'00000028'	X'D8500001'	Error obtaining storage for OLDS block.
X'00000028'	X'D9100001'	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.
		Internal Error
X'0000002C'	all	DBRC internal error. Refer to “Database Recovery Control Service Aids” in the *IMS Version 9: Diagnosis Guide and Reference* for further information.
		Parameter Error

Table 23. Return and Reason Codes for TYPE=OLDS Query Requests (continued)

Return Codes	Reason Codes	Meaning
X'00000030'	X'C9000001'	Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.
X'00000030'	X'C9000002'	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.
X'00000030'	X'C9000003'	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.
X'00000030'	X'C9000004'	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.
X'00000030'	X'C9000005'	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.
X'00000030'	X'C9000010'	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.
X'00000030'	X'D8000001'	Parameter error. Missing or invalid OUTPUT parameter.

Output for the TYPE=OLDS Query Request

Figure 51 on page 405 illustrates the format of the output from a QUERY TYPE=BACKOUT request. Figure 52 on page 406 is a sample DSECT describes in detail the fields of the storage blocks and their relationship to each other.

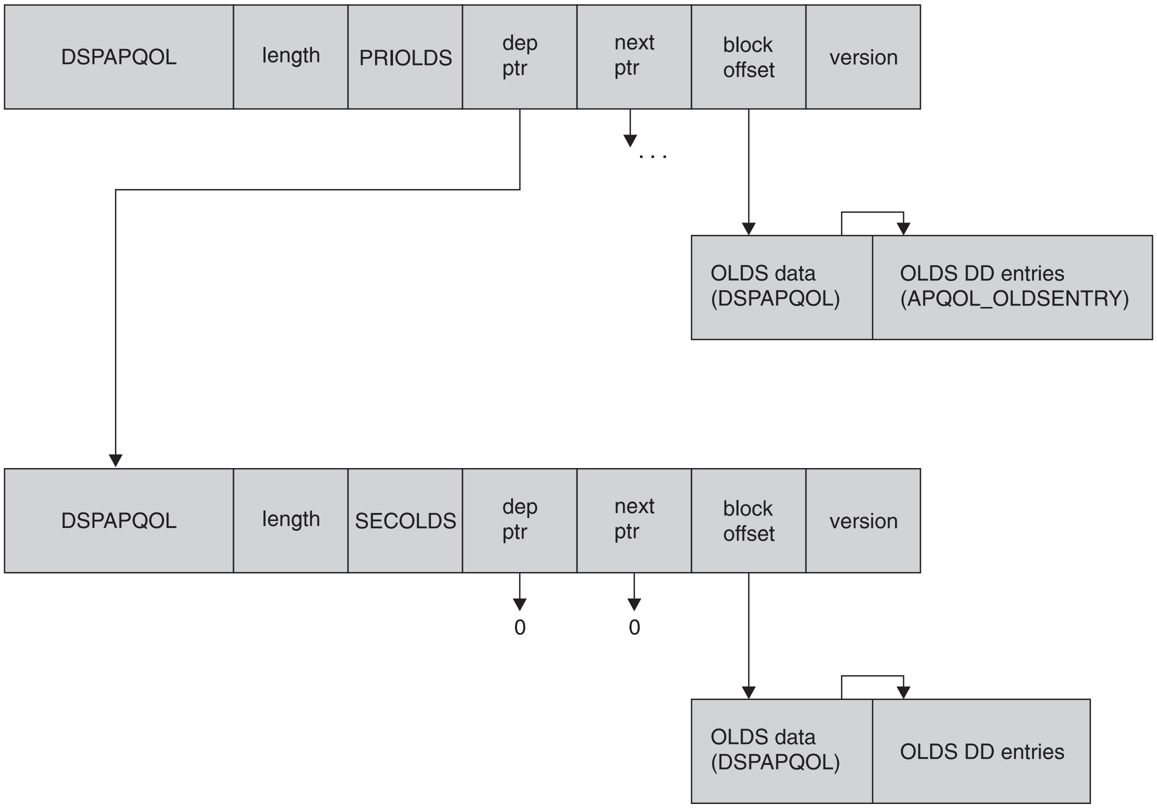


Figure 51. Format for QUERY TYPE=OLDS Output

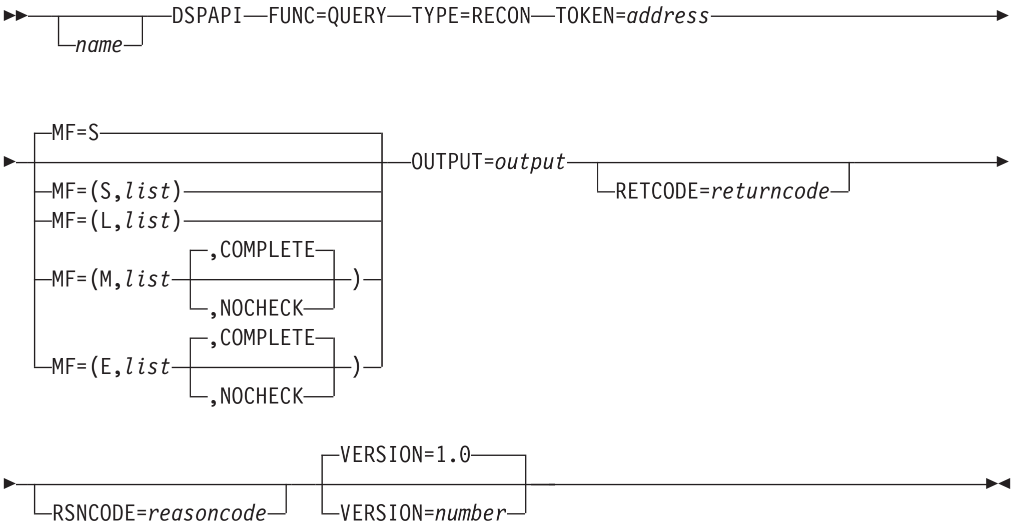
```
  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE       48  DSPAPQOL
       0      (0) UNSIGNED         4   APQOL_OLDSINFO
                                                     Offset to OLDS list
       4      (4) SIGNED           4   *(3)          Reserved
      16     (10) CHARACTER        8   APQOL_SSID    Subsystem ID
      24     (18) UNSIGNED         2   APQOL_OLDSLEN Length of OLDS entry
      26     (1A) SIGNED           2   APQOL_OLDSCOUNT
                                                     Number of OLDS entries
      28     (1C) CHARACTER       12   APQOL_CHKPT0  Checkpoint 0 time
      40     (28) CHARACTER        8   *             Reserved

  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)     DESCRIPTION
======== ======== ========= ========  ============== ===============================
       0      (0) STRUCTURE      128  APQOL_OLDSENTRY
                                                     OLDS entry
       0      (0) CHARACTER        8   APQOL_DDNAME  the OLDS DD name
       8      (8) CHARACTER       44   APQOL_DSNAM   OLDS data set name
      52     (34) CHARACTER       12   APQOL_OPENTIME
                                                     OLDS open time
      64     (40) CHARACTER       12   APQOL_CLOSETIME
                                                     Close time
      76     (4C) CHARACTER       12   APQOL_PRILOGTIME
                                                     Start time of associated
                                                     PRILOG
      88     (58) CHARACTER        8   APQOL_FLSN    LSN of first record in OLDS
      96     (60) CHARACTER        8   APQOL_LLSN    LSN of last record in OLDS
     104     (68) BIT(8)           1   APQOL_FLAG1
                  1... ....             APQOL_RSTBG  Restart begin
                  .1.. ....             APQOL_RSTEN  Restart end
                  ..1. ....             APQOL_COLD   Cold start
                  ...1 ....             APQOL_NOBMP  ERE NOBMP
                  .... 1...             APQOL_SAVER  Backout UORs saved
                  .... .1..             APQOL_NOID   Backouts not identified
                  .... ..1.             APQOL_TRKNG  OLDS created by tracking SS
     105     (69) BIT(8)           1   APQOL_FLAG2   OLDS flags
                  1111 ....             APQOL_STAT   OLDS status
                  1... ....              APQOL_INUSE OLDS is in use
                  .1.. ....              APQOL_ARNED Archive needed
                  ..1. ....              APQOL_ARSCH Archive scheduled (GENJCL)
                  ...1 ....              APQOL_ARSTD Archive job started
                  .... 1...             APQOL_CLERR  Close error on OLDS
                  .... .1..             APQOL_FEOV   Force EOV at archive
                  .... ..1.             APQOL_DUMMY  OLDS not used due to I/O err
                  .... ...1             APQOL_PRVCE  Close error on previous OLDS
     106     (6A) UNSIGNED         1   APQOL_RELVL   Log release level
     107     (6B) UNSIGNED         1   APQOL_GAVER   GENJCL.ARCHIVE version
     108     (6C) BIT(32)          4   APQOL_BLOCKSEQNO
                                                     Block sequence number
     112     (70) CHARACTER        8   APQOL_ARJOB   Name of the archive job
                                                     Generated by GENJCL.ARCHIVE
     120     (78) CHARACTER        6   APQOL_LOCKSEQNO
                                                     Lock sequence number
     126     (7E) CHARACTER        2   *             Reserved

CONSTANTS

  LEN  TYPE           VALUE          NAME           DESCRIPTION
=====  =========  =================  =============== ===============================
    8  CHARACTER   DSPAPQOL          APQOL_EYECATCHER
```

Figure 52. 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



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 409 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=RECON Query Request

Table 24 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 Version 9: Diagnosis Guide and Reference*.

Table 24. Return and Reason Codes for TYPE=RECON Queries

Return Codes	Reason Codes	Meaning
X'00000000'	X'00000000'	Request completed successfully.
	Severe Error	
X'0000000C'	X'C9000001'	Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.
	Storage Error	
X'00000028'	X'D8100001'	Error obtaining storage for the RECON block.
	Internal Error	
X'0000002C'	all	DBRC internal error. Refer to “Database Recovery Control Service Aids” in the *IMS Version 9: Diagnosis Guide and Reference* for further information.
	Parameter Error	
X'00000030'	X'C9000001'	Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.
X'00000030'	X'C9000001'	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.
X'00000030'	X'C9000001'	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.

Table 24. Return and Reason Codes for TYPE=RECON Queries (continued)

Return Codes	Reason Codes	Meaning
X'00000030'	X'C9000001'	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.
X'00000030'	X'C9000001'	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.
X'00000030'	X'D8000001'	Parameter error. Missing or invalid OUTPUT parameter.

Output for the TYPE=RECON Query Request

Figure 53 illustrates the format of the output from a QUERY TYPE=RECON request. Figure 54 on page 410 is a sample DSECT describes in detail the fields of the storage blocks and their relationship to each other.

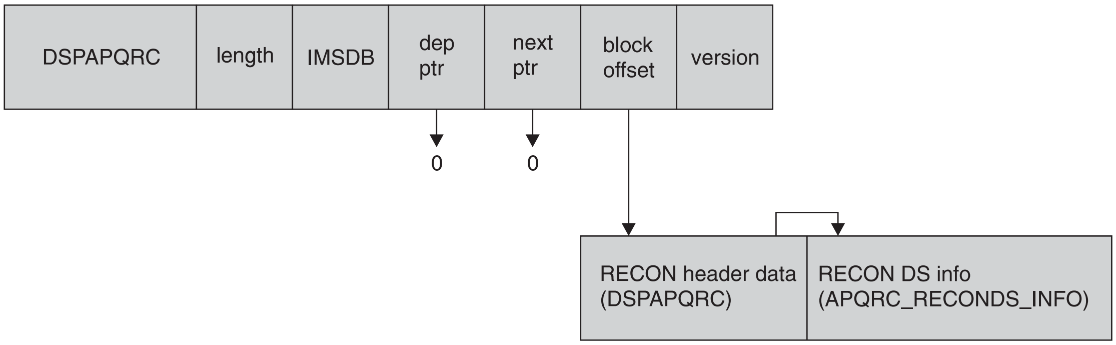

DSPAPQRC
length
IMSDB
dep ptr
next ptr
block offset
version
0
0
RECON header data (DSPAPQRC)
RECON DS info (APQRC_RECONDS_INFO)

Figure 53. Format for QUERY TYPE=RECON Output

```
  OFFSET   OFFSET
 DECIMAL      HEX  TYPE        LENGTH  NAME (DIM)        DESCRIPTION
======== ======== ======== ========  ==============    ===============================
       0      (0) STRUCTURE      560  DSPAPQRC
       0      (0) CHARACTER       44   APQRC_DATA       Initialized with "RECOVERY
                                                        CONTROL DATASET"
      44     (2C) UNSIGNED         4   APQRC_RECONINFO  Offset to RECON data set
                                                        info
      48     (30) SIGNED           4   *(2)             Reserved
      56     (38) UNSIGNED         2   APQAR_RECONINFOLEN  Length of each RECON dataset info element
      58     (3A) UNSIGNED         1   APQRC_RECONCOUNT    # of RECON dataset elements
      59     (3B) CHARACTER        1   *                Reserved
      60     (3C) BIT(8)           1   APQRC_FLAGS      Process flags...
                  1... ....             APQRC_NOCHK     1= NOCHECK log tape dsn check
                  .1.. ....             APQRC_CHK17     1= CHECK17 log tape dsn check
                  ..1. ....             APQRC_CHK44     1= CHECK44 log tape dsn check
                  ...1 ....             APQRC_LSTLG     1= list log DSN
                  .... 1...             APQRC_INUPG     Upgrade in progress.
      61     (3D) BIT(8)           1   APQRC_FLAG2      More flags...
                  1... ....             APQRC_FORCE     1 = FORCER, 0 = NOFORCER
                  .1.. ....             APQRC_CATDS     1=CA|IC|LOGS cataloged
                  ..1. ....             APQRC_TRACE     1 = ext. GTF trace on
                  ...1 ....             APQRC_CMDSAF    SAF enabled
                  .... 1...             APQRC_CMDEXIT   Cmd auth exit enabled
      62     (3E) CHARACTER        2   *                Reserved
      64     (40) CHARACTER      136   APQRC_CLEAN      Fields needed for cleanup
      64     (40) SIGNED           4    APQRC_CSET      0 = updates not in progress,
                                                        >0 = update in progress
      68     (44) SIGNED           4    APQRC_TYPE      Type of update
      72     (48) CHARACTER       32    APQRC_OKEY      Key of original record
     104     (68) CHARACTER       32    APQRC_BKEY      Key of base record that is in
                                                        the process of being changed
     136     (88) CHARACTER       32    APQRC_NKEY      Key of new record
     168     (A8) CHARACTER       16    APQRC_DBID      DBID of DBDS
     168     (A8) CHARACTER        8     APQRC_DBD      DBD name
     176     (B0) CHARACTER        8     APQRC_DDN      DD name
     184     (B8) CHARACTER        8    APQRC_CAGRP     CA group name
     192     (C0) CHARACTER        8    APQRC_DDNEW     New DBDS DD name
     200     (C8) UNSIGNED         2   APQRC_DMBNO      DMB sequence number
     202     (CA) UNSIGNED         2   APQRC_LASTREUSEDDMB#   Last reused DMB number, valid
                                                        only when apqrc_DMBNO is 32767
     204     (CC) CHARACTER        7   APQRC_INITTOKEN  Recon init. token
     211     (D3) CHARACTER        8   APQRC_CMDHLQ     Cmd auth high lvl qual
     219     (DB) UNSIGNED         1   APQRC_MVERS      Minimum IMS version
     220     (DC) BIT(8)           1   APQRC_NWFLG      Fields needed for RECON I/O
                                                        error(s)
```

Figure 54. DSECT of DSPAPQRC (Part 1 of 2)

```
                  1... ....             APQRC_NEW    1=STARTNEW, 0=NONEW
     221     (DD) CHARACTER        3   *                Reserved
     224     (E0) CHARACTER        8   APQRC_SSIDN      SSID for DASD
     232     (E8) CHARACTER        8   APQRC_DASDU      Unit type for DASD
     240     (F0) CHARACTER        8   APQRC_TAPEU      Unit type for tape
     248     (F8) CHARACTER       24   APQRC_TIME
     248     (F8) CHARACTER        2    APQRC_TZDEF    Input offset default
     250     (FA) CHARACTER        5    APQRC_TMFMT     Time format options
     255     (FF) CHARACTER        1    *               Reserved
     256    (100) SIGNED           2    APQRC_TPREC     Time-stamp precision
     258    (102) CHARACTER       12    APQRC_LOGRT     Minimum log retention period
     270    (10E) SIGNED           2    APQRC_TZNUM     Number of entries in time zone label table
     272    (110) CHARACTER        8   APQRC_TZTBL(32)  Time zone label table
     528    (210) BIT(16)          2   APQRC_TROPT      DBRC trace options
     530    (212) CHARACTER        5   APQRC_IMSPLEX    IMSplex name
     535    (217) CHARACTER        5   *                Reserved
     540    (21C) UNSIGNED         4   APQRC_SIZW_DSNUM  SIZEALERT dsnum
     544    (220) UNSIGNED         4   APQRC_SIZW_VOLNUM SIZEALERT volnum
     548    (224) UNSIGNED         4   APQRC_SIZW_PERCENT SIZEALERT percent
     552    (228) UNSIGNED         4   APQRC_LOGW_DSNUM  LOGALERT dsnum
     556    (22C) UNSIGNED         4   APQRC_LOGW_VOLNUM LOGALERT volnum

  OFFSET   OFFSET
 DECIMAL     HEX  TYPE        LENGTH  NAME (DIM)        DESCRIPTION
======== ======== ========= ========  ==============    ===============================
       0      (0) STRUCTURE       53  APQRC_RECONDS_INFO
       0      (0) CHARACTER        8   APQRC_RECONDS_DDNAME  RECON DD name
       8      (8) CHARACTER       44   APQRC_RECONDS_DSNAME  RECON DS name
      52     (34) BIT(8)           1   APQRC_RECONDS_STATUS  RECON DS status
                  1... ....             APQRC_RECONDS_COPY1  COPY 1
                  .1.. ....             APQRC_RECONDS_COPY2  COPY 2
                  ..1. ....             APQRC_RECONDS_SPARE  Spare
                  ...1 ....             APQRC_RECONDS_DISCARDED   Discarded
                  .... 1...             APQRC_RECONDS_UNAVAIL  Unavailable
CONSTANTS

  LEN  TYPE           VALUE          NAME               DESCRIPTION
=====  =========  =================  ===============    ===============================
    8  CHARACTER   DSPAPQRC          APQRC_EYECATCHER
```

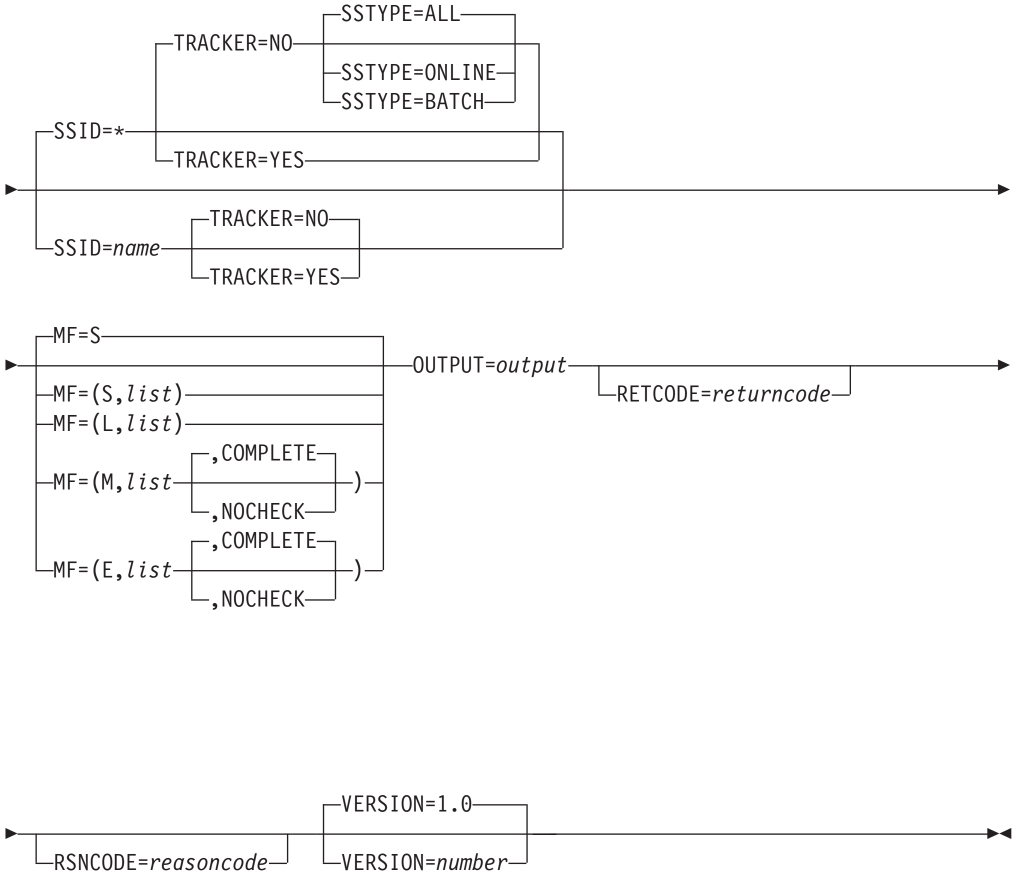
Figure 54. DSECT of DSPAPQRC (Part 2 of 2)

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──────────────►
    └─name─┘
```



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 subsystem name for the backout being queried. '*' (the default) indicates a request for all of the backout information.

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 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 415 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 25 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 Version 9: Diagnosis Guide and Reference*.

Table 25. Return and Reason Codes for TYPE=SUBSYS Query Requests

Return Codes	Reason Codes	Meaning
X'00000000'	X'00000000'	Request completed successfully.
	Warning	

Table 25. Return and Reason Codes for TYPE=SUBSYS Query Requests (continued)

Return Codes	Reason Codes	Meaning
X'00000008'	X'D8600001'	No subsystem record of the requested type, active or tracker, exists.
X'00000008'	X'D8600002'	No batch or online subsystem records exist.
Severe Error		
X'0000000C'	X'C9000001'	Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.
Storage Error		
X'00000028'	X'D8600001'	Error obtaining storage for SUBSYS block.
X'00000028'	X'D9100001'	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.
Internal Error		
X'0000002C'	all	DBRC internal error. Refer to “Database Recovery Control Service Aids” in the *IMS Version 9: Diagnosis Guide and Reference* for further information.
Parameter Error		
X'00000030'	X'C9000001'	Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.
X'00000030'	X'C9000002'	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.
X'00000030'	X'C9000003'	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.
X'00000030'	X'C9000004'	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.
X'00000030'	X'C9000005'	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.
X'00000030'	X'C9000010'	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.
X'00000030'	X'D8000001'	Parameter error. Missing or invalid OUTPUT parameter.
X'00000030'	X'D8600001'	Parameter error. SSTYPE=ONLINE or BATCH can not be specified on requests for a specific SSID or for a tracking subsystem request (TRACKER=YES).

Output for the TYPE=SUBSYS Query Request

Figure 55 illustrates the format of the output from a QUERY TYPE=SUBSYS request. Figure 56 is a sample DSECT describes in detail the fields of the storage blocks and their relationship to each other.

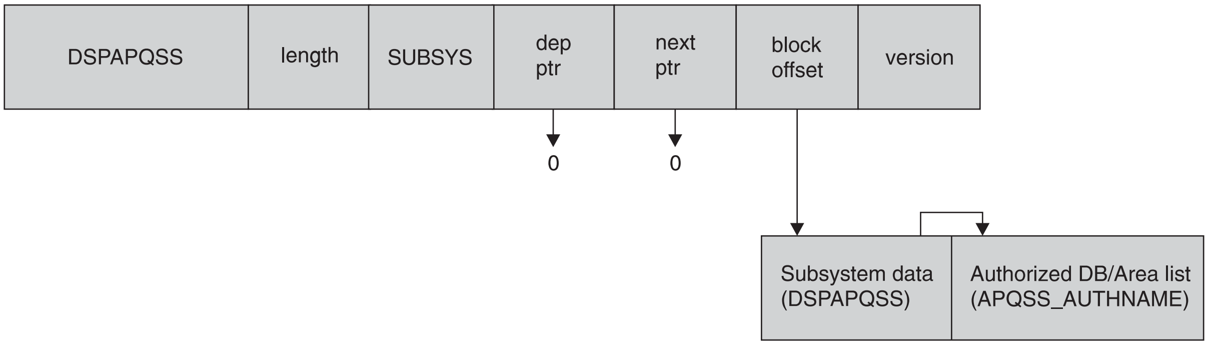

DSPAPQSS
length
SUBSYS
dep ptr
next ptr
block offset
version
0
0
Subsystem data (DSPAPQSS)
Authorized DB/Area list (APQSS_AUTHNAME)

Figure 55. Format for QUERY TYPE=SUBSYS Output

```
  OFFSET   OFFSET
 DECIMAL      HEX TYPE        LENGTH  NAME (DIM)        DESCRIPTION
======== ======== ========= ========  ==============    ===============================
       0      (0) STRUCTURE       64  DSPAPQSS
       0      (0) CHARACTER        8   APQSS_SSID      Subsystem identifier
       8      (8) UNSIGNED         4   APQSS_AUTHLIST  Offset to authd DB/Area list
      12      (C) SIGNED           4   APQSS_AUTHCOUNT Number of authorized DB/Areas
      16     (10) UNSIGNED         2   APQSS_AUTHLEN    Length of auth entry
      18     (12) CHARACTER        6   *                Reserved
      24     (18) CHARACTER       12   APQSS_LOGTIME    Start time of log
      36     (24) UNSIGNED         1   APQSS_RELLVL     Subsystem release level
                                                        '71'X=V7R1,'81'X=V8R1
                                                        '91'X=V9R1, etc.
      37     (25) CHARACTER        1   APQSS_COEXLVL    Coexistence level
      38     (26) UNSIGNED         1   APQSS_IRLMCT     IRLM status count
      39     (27) CHARACTER        1   *                Reserved
      40     (28) CHARACTER        8   APQSS_GSGNAME    Global Service Group name
      48     (30) CHARACTER        5   APQSS_IRLMID     IRLM ID of SS
      53     (35) CHARACTER        5   APQSS_IRLMBK     IRLM ID of backup SS
      58     (3A) BIT(8)           1   APQSS_FLAGS      Flags
                  1... ....             APQSS_TYPE      1=Online | 0=batch
                  .1.. ....             APQSS_ABTERM    Abnormal termination
                  ..1. ....             APQSS_RCVPRC    Recovery processing started
                  ...1 ....             APQSS_BKSIGN    Backup SS signed on
                  .... 1...             APQSS_IRLMFL    IRLM failure
                  .... .1..             APQSS_COMMFL    COMM failure
                  .... ..1.             APQSS_SYSFL     SYS failure
                  .... ...1             APQSS_ACTVST    Status of active SS when backup exists, 1=abterm
      59     (3B) BIT(8)           1   APQSS_FLAGS2     FLAGS 2
                  1... ....             APQSS_SHRING    Sharing covered DBs
                  .1.. ....             APQSS_TRKER     Subsystem is a Tracker
                  ..1. ....             APQSS_TRKTRM   TRACKER has terminated
                  ...1 ....             APQSS_TRCKED   SSID is tracked
                  .... 1...             APQSS_FRSTSO    1ST signon after RSR takeover is in progress
                  .... .1..             APQSS_XRFCAP    SS is XRF capable
                  .... ..11             *               Reserved
      60     (3C) SIGNED           2   APQSS_BCKTKN     Backup recovery token
      62     (3E) CHARACTER        2   *                Reserved
```

Figure 56. DSECT of DSPAPQSS (Part 1 of 2)

```
  OFFSET   OFFSET
 DECIMAL      HEX TYPE        LENGTH  NAME (DIM)        DESCRIPTION
======== ======== ========= ========  ==============    ===============================
       0      (0) STRUCTURE       32  APQSS_AUTHNAME    Names of authd DB/Areas
       0      (0) CHARACTER        8   APQSS_DBNAME     DB name
       8      (8) CHARACTER        8   APQSS_AREANM     If FP, Area name
      16     (10) UNSIGNED         1   APQSS_SHRLVL     Share level
      17     (11) UNSIGNED         1   APQSS_DBACCS     Access intent
      18     (12) UNSIGNED         1   APQSS_DBNCOD     Encoded state
      19     (13) UNSIGNED         1   APQSS_DBSTAT     DB status flags
      20     (14) UNSIGNED         2   APQSS_DBEQCT     DB EQE count
      22     (16) SIGNED           2   APQSS_GLBDMB     Global DMB number
      24     (18) BIT(8)           1   APQSS_AUTHFLAGS  Flags
                  1... ....             APQSS_NRDBUP   Nonrecov DB/Area updated
                  .1.. ....             APQSS_COVRD     DB covered by GSG
                  ..1. ....             APQSS_NRECV     Non-recoverable DB/Area
                  ...1 ....             APQSS_ORDBDS    0=A-J/1=M-V ACTIVE
                  .... 1...             APQSS_OLRON     0 = no OLR active
                  .... .1..             APQSS_OLROWR    0 = no OLR owner
                  .... ..1.             APQSS_OLROWD    0 OLR not owned by SS
      25     (19) CHARACTER        7   *                Reserved

CONSTANTS

  LEN  TYPE           VALUE          NAME               DESCRIPTION
=====  =========  =================  ===============    ===============================
    8  CHARACTER   DSPAPQSS          APQSS_EYECATCHER
```

Figure 56. 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

```
►►──┬──────┬──DSPAPI──FUNC=RELBUF──TOKEN=address──BUFFER=buffer──────────►
    └─name─┘

    ┌─MF=S───────────────────────┐
►───┼────────────────────────────┼──┬──────────────────────┬─────────────►
    ├─MF=(S,list)────────────────┤  └─RETCODE=returncode───┘
    ├─MF=(L,list)────────────────┤
    │            ┌─,COMPLETE─┐   │
    ├─MF=(M,list─┼───────────┼─)─┤
    │            └─,NOCHECK──┘   │
    │            ┌─,COMPLETE─┐   │
    └─MF=(E,list─┼───────────┼─)─┘
                 └─,NOCHECK──┘

                                ┌─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

The following table contains the return and reason codes for the RELBUF request.

Table 26. Return and Reason Codes for RELBUF

Return Codes	Reason Codes	Meaning
X'00000000'	X'00000000'	Request completed successfully.
Partial Success		
X'00000004'	X'D9000001'	There was no storage to free. An address of zero was passed in the BUFFER parameter.
Severe Error		
X'0000000C'	X'C9000001'	Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.
Storage Error		
X'00000028'	X'D9100001'	An error occurred attempting to release storage.
Parameter Error		
X'00000030'	X'C9000001'	Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.
X'00000030'	X'C9000002'	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.
X'00000030'	X'C9000003'	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.
X'00000030'	X'C9000004'	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.

Table 26. Return and Reason Codes for RELBUF (continued)

Return Codes	Reason Codes	Meaning
X'00000030'	X'C9000007'	Invalid BUFFER address. The address of the BUFFER to be released failed validity checking. The address specifies storage not owned by the calling program.
X'00000030'	X'D9100001'	The header of a storage block to be freed is invalid.
X'00000030'	X'D9100002'	The length of a storage block to be freed is not greater than zero.

Chapter 19. DBRC Start Request

Use the DBRC start request (STARTDBRC) to initialize the DBRC API and to start DBRC.

In This Chapter:

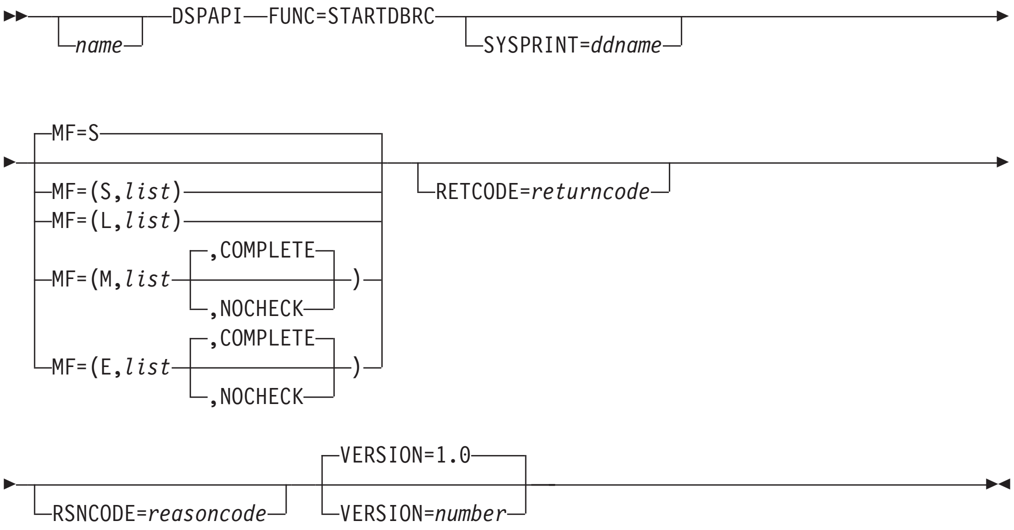
- "STARTDBRC Request"

STARTDBRC Request

"STARTDBRC Usage" contains the details for the DBRC start request.

STARTDBRC Usage

Syntax for the STARTDBRC Request



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.

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 STARTDBRC

Table 27 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 Version 9: Diagnosis Guide and Reference*.

Table 27. Return and Reason Codes for the STARTDBRC Request

Return Codes	Reason Codes	Meaning
X'00000000'	X'00000000'	Request completed successfully.
	Severe Error	
X'0000000C'	X'C9000002'	DBRC could not successfully initialize during phase 0 of DBRC initialization
X'0000000C'	X'C9000003'	DBRC could not successfully initialize during phase 1 of DBRC initialization.
X'0000000C'	X'C9000004'	A z/OS LOAD failed for the DBRC module DSPCRTR0.
X'0000000C'	X'C9000005'	STORAGE request failure. The API cannot obtain storage necessary to complete the request.
X'0000000C'	X'C9000012'	STARTDBRC has been previously issued without an intervening STOPDBRC.
	Parameter Error	
X'00000030'	X'C9000001'	Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.
X'00000030'	X'C9000002'	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.
X'00000030'	X'C9000003'	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.

Table 27. Return and Reason Codes for the STARTDBRC Request (continued)

Return Codes	Reason Codes	Meaning
X'00000030'	X'C9000004'	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.
X'00000030'	X'C9000006'	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.

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──────────────────────►
    └─name─┘


   ┌─MF=S──────────────────────┐
►──┼───────────────────────────┼──┬────────────────────────┬────────────►
   ├─MF=(S,list)───────────────┤  └─RETCODE=returncode─────┘
   ├─MF=(L,list)───────────────┤
   │              ┌─,COMPLETE─┐   │
   ├─MF=(M,list───┼───────────┼─)─┤
   │              └─,NOCHECK──┘   │
   │              ┌─,COMPLETE─┐   │
   └─MF=(E,list───┼───────────┼─)─┘
                  └─,NOCHECK──┘

                                ┌─VERSION=1.0─────┐
►──┬──────────────────────────┬─┼─────────────────┼─────────────────────►◄
   └─RSNCODE=reasoncode───────┘ └─VERSION=number──┘
```

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.

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 28 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 Version 9: Diagnosis Guide and Reference*.

Table 28. Return and Reason Codes for the STOPDBRC Request

Return Codes	Reason Codes	Meaning
X'00000000'	X'00000000'	Request completed successfully.
Severe Error		
X'0000000C'	X'C9000001'	Invalid TOKEN. The TOKEN block passed to the API is not recognized as a TOKEN created by a FUNC=STARTDBRC call.
Parameter Error		
X'00000030'	X'C9000001'	Parameter error. The function (FUNC) specified in the parameter list passed to the API is invalid.
X'00000030'	X'C9000002'	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.
X'00000030'	X'C9000003'	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.
X'00000030'	X'C9000004'	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.

Part 4. Appendixes

Appendix A. Symbolic Keywords Recognized by DBRC

Table 29 through Table 35 on page 437 show the symbolic keywords that DBRC recognizes in the IBM-supplied skeletal JCL execution members.

Table 29 shows the keywords that all supported utilities recognize.

Table 30 on page 430 through Table 35 on page 437 show the keywords that each individual utility recognizes.

These tables and the figures in “Understanding the IBM-Supplied Skeletal JCL Execution Members” on page 438 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 430 for a list of the symbolic keywords supported by the utility and see “GENJCL.ARCHIVE” on page 221 for details using the command.
- `GENJCL.CA`, which generates JCL for the Database Change Accumulation utility. See “Database Change Accumulation Utility (CAJCL)” on page 431 for a list of the symbolic keywords supported by the utility and see “GENJCL.CA” on page 225 for details of the command usage.
- `GENJCL.CLOSE`, which generates JCL for the Log Recovery utility. See “Log Recovery Utility (LOGCLJCL)” on page 432 for a list of the symbolic keywords supported by the utility and see “GENJCL.CLOSE” on page 229 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 433 for a list of the symbolic keywords supported by the utilities and see “GENJCL.IC” on page 231 and “GENJCL.OIC” on page 240 for details of the commands usage.
- `GENJCL.RECEIVE`, which generates JCL for the Database Recovery utility. See “Database Recovery Utility- Receive (ICRCVJCL)” on page 435 for a list of the symbolic keywords supported by the utility and see “GENJCL.RECEIVE” on page 245 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 437 for a list of the symbolic keywords supported by the utility and see “GENJCL.RECOV” on page 248 for details of the command usage.)

All Supported Utilities

Table 29 explains the symbolic keywords recognized by all the supported utilities.

Table 29. Symbolic Keywords for All Supported Utilities

Keyword	Description
`%RCNDSN1`	Name of the RECON1 data set if RECONs are allocated by JCL. Set to null if RECONs are dynamically allocated.
`%RCNDSN2`	Name of the RECON2 data set if RECONs are allocated by JCL. Set to null if RECONs are dynamically allocated.
`%RCNDSN3`	Name of the RECON3 data set if RECONs are allocated by JCL. Set to null if RECONs are dynamically allocated.

Table 29. Symbolic Keywords for All Supported Utilities (continued)

Keyword	Description
%STPNO	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.
%TIME	The time of day, in the form hhmmss.
%GRPINDX	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.
%CNTR	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 1 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.
%DATE	The day of the year, in the form yyddd.
%DATE7	The day of the year, in the form yyyyddd.

Log Archive Utility (ARCHJCL)

Table 30 explains the symbolic keywords recognized by the Log Archive utility.

Table 30. Symbolic Keywords for Log Archive Utility

Keyword	Description
%SSID	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.
%DDNAMES	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.
%OLDSDDN	The ddname of one or more specific OLDS.
%OLDSDSN	The data set name of the one or more OLDS.

Table 30. Symbolic Keywords for Log Archive Utility (continued)

Keyword	Description
%ARDATE	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
%ARDATE7	The date (from the open time stamp) of the first OLDS that is to be archived. The date is in the form yyyyddd 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.
%ARCSLDS	Set to YES when the SLDS parameter is specified.

Database Change Accumulation Utility (CAJCL)

Table 31 explains the symbolic keywords recognized by the Database Change Accumulation utility.

Table 31. Symbolic Keywords for Database Change Accumulation Utility

Keyword	Description
%CAGRP	The CA group name.
%DSLLGTM	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.
%CAODSN	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.
%CAOUNIT	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.
%CAOVOLS	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.
%CAOFSEQ	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.
%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.

Table 31. Symbolic Keywords for Database Change Accumulation Utility (continued)

Keyword	Description
%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.
%CABFSEQ	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.
%LOGDSN	The data set name of the log data set.
%LOGUNIT	The unit type of the log data set.
%LOGVSEQ	The volume sequence number of the log data set.
%LOGVOLS	The volume serial numbers of the log data set.
%LOGFSEQ	The file sequence number of the log data set.
%LOGSEL	Set to YES if any log data sets were selected. Otherwise, set to NO.
%CADB0	This keyword generates the DB0 control statements for the Database Change Accumulation utility.

Log Recovery Utility (LOGCLJCL)

Table 32 explains the symbolic keywords recognized by the Log Recovery utility.

Table 32. Symbolic Keywords for Log Recovery Utility

Keyword	Description
%SSID	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.
%CDDNAME	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.
%OLDSTYP	The type of OLDS, primary or secondary (set to P or S, respectively).
%OLDSDSN	The data set name of the OLDS.
%WADS	If the OLDS to be closed is currently open, this keyword is set to YES. Otherwise, this keyword is set to NO.
%NDDNAME	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.
%PDDNAME	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.

Database Image Copy Utility, Database Image Copy Utility 2, and Online Database Image Copy Utility (ICJCL and OICJCL)

Table 33 explains the symbolic keywords recognized by the Database Image Copy utilities.

Table 33. Symbolic Keywords for Database Image Copy Utilities

<table>
<tr><th>Keyword</th><th>Description</th></tr>
<tr><td>%PSB</td><td>The PSB name, which is set from the PSB parameter on the GENJCLcommand. This keyword is applicable only for the Database Online Image Copy utility.</td></tr>
<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>%DBADDN</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>%DBADSAV</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, %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:
<table>
<tr><th>Column</th><th>Setting</th></tr>
<tr><td><b>1</b></td><td>D</td></tr>
<tr><td><b>2</b></td><td>Number of image copy data sets to be produced (either 1 or 2)</td></tr>
<tr><td><b>4-11</b></td><td>Database name</td></tr>
<tr><td><b>13-20</b></td><td>ddname of the DBDS</td></tr>
<tr><td><b>22-30</b></td><td>ddname of the primary image copy data set</td></tr>
<tr><td><b>31-38</b></td><td>ddname of the duplicate image copy data set, if one is produced.</td></tr>
<tr><td><b>40-43</b></td><td>Checkpoint interval (applicable only for Online Database Image Copy utility).</td></tr>
</table>
All other columns are set to blanks.</td></tr>
</table>

Table 33. Symbolic Keywords for Database Image Copy Utilities (continued)

Keyword	Description

%ICDSN1,
%ICDSN2,
%ICDSN3,
%ICDSN4
	The data set name of the image copy data set is %ICDSN1.
	If NOREUSE is specified for the DBDS, DBRC generates the following data set name:
	`IMS.dbname.ddname.IC.IChhmmss`
	where:
	• dbname is the database name of the DBDS
	• ddname is the ddname of the DBDs
	• hhmmss is the current time of day
	If Sameds is specified for a group of DBDSs, DBRC generates the following data set name:
	`IMSVS.GROUP.grpname.IC.IChhmmss`
	where
	• grpname is the name specified in the GROUP or DBD parameter
	• hhmmss is the current time of day
	If multiple image copy data sets are to be produced, %ICDSN3, or %ICDSN4 are set similarly.

%ICUNIT1, %ICUNIT2, %ICUNIT3, %ICUNIT4	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.
%ICFSEQ1, %ICFSEQ2, %ICFSEQ3, %ICFSEQ4	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.
%ICVOLS1, %ICVOLS2, %ICVOLS3, %ICVOLS4	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.
%ICVCNT1, %ICVCNT2, %ICVCNT3, %ICVCNT4	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.
%SMSGRP	A numeric value indicating the status of group processing: **Column** — **Setting** **0** — 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. **1** — Processing the only member of the group **2** — Processing the first of multiple members **3** — Processing an intermediate member **4** — Processing the last member

Table 33. Symbolic Keywords for Database Image Copy Utilities (continued)

Keyword	Description
%SMS1DS	A numeric value indicating whether Sameds was specified: **Column** — **Setting** **0** — Sameds was not specified **1** — Sameds was specified
%GROUP	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.

Database Recovery Utility- Receive (ICRCVJCL)

Table 34 explains the symbolic keywords recognized by the Database Recovery (Receive) utility.

Table 34. Symbolic Keywords for Database Recovery (Receive) Utility

Keyword	Description
%DBNAME	The database name of the DBDS to be covered. %DBNAME is set from the DBD parameter on the GENJCL.RECEIVE command.
%DBDDN	The ddname of the DBDS, %DBDDN is set from the DDN parameter on the GENJCL.RECEIVE command.
%DBDSN	The data set name of the DBDS, %DBDSN is set from the DBDS record in RECON.
%DBDSAM	Set to VSAM for a VSAM DBDS. Otherwise, set to null.
%DBUSID	The update set identifier for the DBDS.
%ALLUSID	The update set identifier of the most-recent ALLOC record for the DBDS.
%MDBNAME	The HALDB master name, if this is a DBDS of a HALDB partition. This keyword is set to NULL for non-HALDBs.
%DSLLGTM	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.
%ICDSN	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.
%ICUNIT	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.
%ICVOLS	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.
%ICFSEQ	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.
%ICUSID	The update set identifier for the image copy.
%CADSN	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.

Table 34. Symbolic Keywords for Database Recovery (Receive) Utility (continued)

Keyword	Description
%CAUNIT	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.
%CAVOLS	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.
%CAFSEQ	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.
%OLDFLRID	The log record sequence number (log record ID) of the first log record in the OLDS.
%OLDLLRID	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.
%SLDFLRID	The log record sequence number (log record ID) of the first log record in the SLDS.
%SLDFSEQ	The file sequence number of the SLDS.
%SLDLLRID	The log record sequence number (log record ID) of the last log record in the SLDS. If the SLDS has not been closed, %SLDLLRID is set to null.
%SLDREMOT	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.
%SLDUNIT	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.
%SLDVOLS	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.
%LOGDSN	The data set name of the log data set.
%LOGUNIT	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.
%LOGVSEQ	The volume sequence number of the log data set.
%LOGVOLS	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.
%LOGFSEQ	The file sequence number of the log data set.
%LOGSEL	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.
%LOGFLRID	The log record sequence number (log record ID) of the first log record in the RLDS.
%LOGLLRID	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.
%LOGREMOT	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. %LOGREMOT is set to null if the RLDS was created locally by an active IMS subsystem.

Table 34. Symbolic Keywords for Database Recovery (Receive) Utility (continued)

Keyword	Description
%RVSYSIN	The Database Recovery utility control statement. Columns in the statement are set as follows: **Column** — **Setting** **1** — S **4-11** — Database name **13-20** — Data set or area ddname **22-29** — DFSUDUMP All other columns are set to blanks.

Database Recovery Utility-Recovery (RECOVJCL)

Table 35 explains the symbolic keywords recognized by the Database Recovery (Recover) utility.

Table 35. Symbolic Keywords for Database Recovery Utility - Recovery (RECOVJCL)

Keyword	Description
%DBNAME	The database name of the DBDS to be RSR-covered. %DBNAME is set from the DBD parameter on the GENJCL.RECOV command.
%DBDDN	The ddname of the DBDS; %DBDDN is set from the DDN parameter on the GENJCL.RECOV command.
%DBDSN	The data set name of the DBDS; %DBDSN is set from the DBDS record of the DBDSs.
%DBDSAM	Set to VSAM for a VSAM DBDS. Otherwise, set to null.
%MDBNAME	The HALDB master name, if this is a DBDS of a HALDB partition. This keyword is set to NULL for non-HALDBs.
%DSLLGTM	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.
%SMS	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.
%ICDSN	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.
%ICUNIT	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.
%ICVOLS	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.
%ICFSEQ	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.
%CADSN	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.

Table 35. Symbolic Keywords for Database Recovery Utility - Recovery (RECOVJCL) (continued)

Keyword	Description
%CAUNIT	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.
%CAVOLS	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.
%CAFSEQ	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.
%LOGDSN	The data set name of the log data set.
%LOGUNIT	The unit type of the log data set.
%LOGVSEQ	The volume sequence number of the log data set.
%LOGVOLS	The volume serial numbers of the log data set.
%LOGFSEQ	The file sequence number of the log data set.
%LOGSEL	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.
%RCSYSIN	The Database Recovery utility control statement. Columns in the statement are set as follows: **Column** **Setting** **1** S **4-11** Database name **13-20** Data set ddname **31-42** The specified time stamp if the RCVTIME parameter was specified on the GENJCL.RECOV command. Otherwise, blank. **44** C, if the USEDBDS parameter was specified on the GENJCL.RECOV command. Otherwise, blank. All other columns are set to blanks.
%RCVFULL	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.

Understanding the 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.

Related Reading: Instructions on what you must do before using the skeletal JCL execution members are in "Using IBM-Supplied Skeletal JCL" on page 88.

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 `JOBLIB`, `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 57 on page 440 is a listing of ARCHJCL. A description of the statements in ARCHJCL follows Figure 57 on page 440.

```
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
 //*
 //*    THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
 //*    KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
 //*    THE IMS/ESA DATABASE RECOVERY CONTROL FEATURE.
 //*
 //*                 JCL FOR ARCHIVE UTILITY
 //*
 //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
Note: The following lines are used to archive OLDS.
 %DELETE (%ARCSLDS EQ 'YES')
 %SELECT    OLDS(%SSID,(%ddnames))
 //%OLDSDDN DD  DSN=%OLDSDSN,DISP=SHR
 %ENDSEL
 //DFSSLOGP DD  DSN=IMS.SLDSP.%SSID.D%ARDATE.T%ARTIME.V%ARVERS,
 //             UNIT=3400,VOL=(,,,99),
 //             DISP=(NEW,KEEP),LABEL=(1,SL)
 //DFSSLOGS DD  DSN=IMS.SLDSS.%SSID.D%ARDATE.T%ARTIME.V%ARVERS,
 //             UNIT=3400,VOL=(,,,99),
 //             DISP=(NEW,KEEP),LABEL=(1,SL)
 //RLDSDD1  DD  DSN=IMS.RLDSP.%SSID.D%ARDATE.T%ARTIME.V%ARVERS,
 //             UNIT=3400,VOL=(,,,99),
 //             DISP=(NEW,KEEP),LABEL=(1,SL)
 //RLDSDD2  DD  DSN=IMS.RLDSS.%SSID.D%ARDATE.T%ARTIME.V%ARVERS,
 //             UNIT=3400,VOL=(,,,99),
 //             DISP=(NEW,KEEP),LABEL=(1,SL)
 %ENDDEL
```

Figure 57. 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.D%ARDATE.T%ARTIME,
 //             UNIT=3400,VOL=(,,,99),
 //             DISP=(NEW,PASS),LABEL=(1,SL)
 //RLDSDD1   DD DSN=IMSVS.RLDS1.%SSID.D%ARDATE.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.D%ARDATE.T%ARTIME,
 //             UNIT=3400,VOL=(,,,99),
 //             DISP=(NEW,PASS),LABEL=(1,SL)
 //RLDSDD2   DD DSN=IMSVS.RLDS2.%SSID.D%ARDATE.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 DDNOUT1(RLDSDD1) DDNOUT2(RLDSDD2) DBRECOV
 /*
```

Figure 57. IBM-Supplied Skeletal JCL for the Log Archive Utility (Part 2 of 3)

```
Note: The following lines are used for the SLDSs process
 %DELETE  (%ARCSLDS EQ 'NO')
 //****************************************************************
 //*                                                              *
 //* The following optional steps are used to manage the data     *
 //* sets used in the previous SLDS archive step. If the previous *
 //* step completed successfully, the input data sets will be      *
 //* deleted and the output data sets will be cataloged. The      *
 //* output data sets will be deleted if the previous step        *
 //* failed.                                                      *
 //*                                                              *
 //****************************************************************
 //GOODRC%STPNO   EXEC PGM=IEFBR14,COND=(0,NE,AR%STPNO)
 %SELECT  SLDS(%SSID,ALL)
 //PSLDS1   DD DSN=%SLDSDSN,DISP=(OLD,DELETE)
 %ENDSEL
 %SELECT  SSLDS(%SSID,ALL)
 //SSLDS1   DD DSN=%SLDSDSN,DISP=(OLD,DELETE)
 %ENDSEL
 //DD1      DD DSN=*.AR%STPNO.DFSSLOGP,DISP=(OLD,CATLG)
 //DD2      DD DSN=*.AR%STPNO.DFSSLOGS,DISP=(OLD,CATLG)
 //DD3      DD DSN=*.AR%STPNO.RLDSDD1,DISP=(OLD,CATLG)
 //DD4      DD DSN=*.AR%STPNO.RLDSDD2,DISP=(OLD,CATLG)
 /*
 //BADRC%STPNO   EXEC PGM=IEFBR14,COND=(0,EQ,AR%STPNO)
 //DD1      DD DSN=*.AR%STPNO.DFSSLOGP,DISP=(OLD,DELETE)
 //DD2      DD DSN=*.AR%STPNO.DFSSLOGS,DISP=(OLD,DELETE)
 //DD3      DD DSN=*.AR%STPNO.RLDSDD1,DISP=(OLD,DELETE)
 //DD4      DD DSN=*.AR%STPNO.RLDSDD2,DISP=(OLD,DELETE)
 /*
 %ENDDEL
```

Figure 57. IBM-Supplied Skeletal JCL for the Log Archive Utility (Part 3 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 %OLDSDSN 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.

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 dual SLDS logging, delete this DD statement 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 (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.

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 (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 being archived.

If you are not using RLDS logging, delete this statement from the execution member. If this statement is deleted, the DDNOUT2(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 %SLDSDSN 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 %SLDSDSN 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 (hhmmsst) 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.

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 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 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 (hhmmsst) 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 (hhmmsst) 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(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.

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 58 on page 446 is a listing of CAJCL. A description of the statements in CAJCL follows Figure 58 on page 446.

```
//CA%STPNO EXEC PGM=DFSUCUM0,PARM='CORE=100000',REGION=800K
//*
//*  THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
//*  KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
//*  THE IMS/ESA DATABASE RECOVERY CONTROL FEATURE.
//*
//*      JCL FOR CHANGE ACCUMULATION
//*
//STEPLIB  DD  DSN=IMS.RESLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=A
%DELETE    (%RCNDSNl EQ '')
//RECON1   DD  DSN=%RCNDSNl,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
//SYSOUT   DD  SYSOUT=A
//SORTLIB  DD  DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01 DD  UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//SORTWK02 DD  UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//SORTWK03 DD  UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//SORTWK04 DD  UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//SORTWK05 DD  UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//SORTWK06 DD  UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
%DELETE    (%CAODSN EQ '')
//DFSUCUMO DD  DSN=%CAODSN,UNIT=%CAOUNIT,
//             VOL=(PRIVATE,,,,SER=(%CAOVOLS)),
//             LABEL=(%CAOFSEQ,SL),
//             DISP=OLD
%ENDDEL
%DELETE    (%CAODSN NE '')
//DFSUCUMO DD  DUMMY,DCB=BLKSIZE=100
%ENDDEL
//DFSUCUMN DD  DSN=%CANDSN,UNIT=%CANUNIT,
//             VOL=(PRIVATE,,,%CANVCNT,SER=(%CANVOLS)),
//             LABEL=(%CANFSEQ,SL),
//             DISP=(NEW,KEEP)
%SELECT    RLDS((%CAGRP),(FROM(%DSLLGTM)))
//DFSULOG  DD  DSN=%LOGDSN,UNIT=%LOGUNIT,
//             VOL=(PRIVATE,,%LOGVSEQ,,SER=(%LOGVOLS)),
//             LABEL=(%logfseq,SL),
//             DCB=RECFM=VB,
//             DISP=OLD
%ENDSEL
%DELETE    (%LOGSEL EQ 'YES')
//DFSULOG  DD  DUMMY,DCB=BLKSIZE=100
%ENDDEL
//DFSUDD1  DD  DUMMY
//SYSIN    DD  *
%CADB0
/*
```

Figure 58. IBM-Supplied Skeletal JCL for the Database Change Accumulation Utility

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.

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 *IMS Version 9: 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. LOGCLJCL is used when the GENJCL.CLOSE command is issued.

Figure 59 on page 449 is a listing of LOGCLJCL. A description of the statements in LOGCLJCL follows Figure 59 on page 449.

```
//CL%STPNO        EXEC PGM=DFSULTR0,PARM='IMSID=%SSID'
//*
//*  THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
//*  KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
//*  THE IMS/ESA DATABASE RECOVERY CONTROL FEATURE.
//*
//*               JCL FOR LOG RECOVERY UTILITY
//*
//STEPLIB         DD DSN=IMS.RESLIB,DISP=SHR
//SYSPRINT        DD  SYSOUT=A
%DELETE           (%RCNDSNl 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
%SELECT           OLDS(%SSID,(%CDDNAME))
//DFSOL%OLDSTYP  DD  DSN=%OLDSDSN,DISP=SHR
%ENDSEL
%DELETE           (%WADS EQ 'NO')
//DFSWADS0        DD  DSN=IMS.WADS0,DISP=OLD
%ENDDEL
%DELETE           (%WADS EQ 'YES')
%SELECT           OLDS(%SSID,(%NDDNAME))
//DFSNOL%OLDSTYP DD  DSN=%OLDSDSN,DISP=SHR
%ENDSEL
%ENDDEL
%DELETE           (%PDDNAME EQ '')
%SELECT           OLDS(%SSID, (%PDDNAME))
//DFSPOL%OLDSTYP DD  DSN=%OLDSDSN,DISP=SHR
%ENDSEL
%ENDDEL
//SYSIN           DD  *
CLS
/*
```

Figure 59. 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 %CDDNAME identifies the OLDS by its DD name. The OLDS type, primary or secondary, replaces the %OLDSTYP keyword. The resulting ddname is DFSOLP or DFSOLS. The %OLDSDSN 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 DFSWADS*n*. 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 %NDDNAME 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 %OLDSDSN 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 %PDDNAME 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 DFSPOLS. 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:

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 60 on page 452 is a listing of ICJCL. A description of the statements in ICJCL follows Figure 60 on page 452.

```
%DELETE  (%SMSGRP 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:
//*-----|         %SMSGRP = 0 - Not SMS or not Group
//*-----|         %SMS1DS = 0 - Not SMS or not Group or not ONEDS
//*-----|
//*-----|    Multiple DBDSs per step:
//*-----|         %SMSGRP = 1 - SMS + Group + only member
//*-----|         %SMSGRP = 2 - SMS + Group + first member
//*-----|         %SMSGRP = 3 - SMS + Group + not first or last
//*-----|         %SMSGRP = 4 - SMS + Group + last  member
//*-----|         %SMS1DS = 1 - %SMSGRP > 0 + ONEDS
//*-----------------------------------------------------------------
//*-----|  Section 1 is used only once per step
//*-----------------------------------------------------------------
%ENDDEL
%DELETE  (%SMSGRP 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.
//*
//STEPLIB   DD DSN=IMSVS.RESLIB,DISP=SHR
//SYSPRINT  DD SYSOUT=A
%ENDDEL
%DELETE  (%RCNDSN1 EQ '' | %SMSGRP GT '2')
//RECON1    DD DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE  (%RCNDSN2 EQ '' | %SMSGRP GT '2')
//RECON2    DD DSN=%RCNDSN2,DISP=SHR
%ENDDEL
%DELETE  (%RCNDSN3 EQ '' | %SMSGRP GT '2')
//RECON3    DD DSN=%RCNDSN3,DISP=SHR
%ENDDEL
%DELETE  (%SMSGRP GT '2')
//IMS       DD DSN=IMSVS.DBDLIB,DISP=SHR
%ENDDEL
%DELETE  (%SMSGRP NE '')
```

Figure 60. IBM-Supplied Skeletal JCL for the Database Image Copy Utility and the Database Image Copy 2 Utility (Part 1 of 3)

```
//*-----------------------------------------------------------------
//*-----|  Single output data set if SMS and ONEDS
//*-----------------------------------------------------------------
%ENDDEL
%DELETE  (%SMSGRP GT '2' | %SMS1DS NE '1')
//%ICDDN1      DD DSN=%ICDSN1,UNIT=%ICUNIT1,
//             VOL=(PRIVATE,,,%ICVCNT1,SER=(%ICVOLS1)),
//             LABEL=(%ICFSEQ1,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%COPIES LT '2' | %SMSGRP GT '2' | %SMS1DS NE '1')
//%ICDDN2      DD DSN=%ICDSN2,UNIT=%ICUNIT2,
//             VOL=(PRIVATE,,,%ICVCNT2,SER=(%ICVOLS2)),
//             LABEL=(%ICFSEQ2,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%COPIES LT '3' |%SMSGRP GT '2' | %SMS1DS NE '1')
//%ICDDN3      DD DSN=%ICDSN3,UNIT=%ICUNIT3,
//             VOL=(PRIVATE,,,%ICVCNT3,SER=(%ICVOLS3)),
//             LABEL=(%ICFSEQ3,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%COPIES LT '4' |%SMSGRP GT '2' | %SMS1DS NE '1')
//%ICDDN4      DD DSN=%ICDSN4,UNIT=%ICUNIT4,
//             VOL=(PRIVATE,,,%ICVCNT4,SER=(%ICVOLS4)),
//             LABEL=(%ICFSEQ4,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%SMSGRP NE '')
//*-----------------------------------------------------------------
//*-----|  Section 2 is used once per DBDS (any value of %SMSGRP)
//*-----------------------------------------------------------------
%ENDDEL
%SELECT  DBDS((%DBNAME,%DBDDN))
%DELETE  (%DBADSAV NE 'AVAIL')
//%DBADDN   DD DSN=%DBDSN,DISP=%CICDISP
%ENDDEL
%DELETE  (%DBADSAV NE '' | %MDBNAME NE'')
//%DBDDN    DD DSN=%DBDSN,DISP=%CICDISP
%ENDDEL
%ENDSEL
%DELETE  (%SMSGRP NE '')
//*-----------------------------------------------------------------
//*-----|  Output data sets not included if SMS + Group + ONEDS
//*-----------------------------------------------------------------
%ENDDEL
%DELETE  (%SMS1DS EQ '1')
//%ICDDN1      DD DSN=%ICDSN1,UNIT=%ICUNIT1,
//             VOL=(PRIVATE,,,%ICVCNT1,SER=(%ICVOLS1)),
//             LABEL=(%ICFSEQ1,SL),
%ENDDEL
%DELETE (%SMS EQ '1' | %SMS1DS EQ '1')
//             DISP=(NEW,KEEP),DCB=BUFNO=10
%ENDDEL
%DELETE (%SMS NE '1' | %SMS1DS EQ '1')
//             DISP=(NEW,KEEP)
%ENDDEL
```

Figure 60. IBM-Supplied Skeletal JCL for the Database Image Copy Utility and the Database Image Copy 2 Utility (Part 2 of 3)

```
%DELETE  (%COPIES EQ '1' | %SMS EQ '1' | %SMS1DS EQ '1')
//%ICDDN2      DD DSN=%ICDSN2,UNIT=%ICUNIT2,
//             VOL=(PRIVATE,,,%ICVCNT2,SER=(%ICVOLS2)),
//             LABEL=(%ICFSEQ2,SL),
//             DISP=(NEW,KEEP),DCB=BUFNO=10
%ENDDEL
%DELETE  (%COPIES EQ '1' | %SMS NE '1' | %SMS1DS EQ '1')
//%ICDDN2      DD DSN=%ICDSN2,UNIT=%ICUNIT2,
//             VOL=(PRIVATE,,,%ICVCNT2,SER=(%ICVOLS2)),
//             LABEL=(%ICFSEQ2,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%COPIES LE '2' | %SMS EQ '1' | %SMS1DS EQ '1')
//%ICDDN3      DD DSN=%ICDSN3,UNIT=%ICUNIT3,
//             VOL=(PRIVATE,,,%ICVCNT3,SER=(%ICVOLS3)),
//             LABEL=(%ICFSEQ3,SL),
//             DISP=(NEW,KEEP),DCB=BUFNO=10
%ENDDEL
%DELETE  (%COPIES LE '2' | %SMS NE '1' | %SMS1DS EQ '1')
//%ICDDN3      DD DSN=%ICDSN3,UNIT=%ICUNIT3,
//             VOL=(PRIVATE,,,%ICVCNT3,SER=(%ICVOLS3)),
//             LABEL=(%ICFSEQ3,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%COPIES LE '3' | %SMS EQ '1' | %SMS1DS EQ '1')
//%ICDDN4      DD DSN=%ICDSN4,UNIT=%ICUNIT4,
//             VOL=(PRIVATE,,,%ICVCNT4,SER=(%ICVOLS4)),
//             LABEL=(%ICFSEQ4,SL),
//             DISP=(NEW,KEEP),DCB=BUFNO=10
%ENDDEL
%DELETE  (%COPIES LE '3' | %SMS NE '1' | %SMS1DS EQ '1')
//%ICDDN4      DD DSN=%ICDSN4,UNIT=%ICUNIT4,
//             VOL=(PRIVATE,,,%ICVCNT4,SER=(%ICVOLS4)),
//             LABEL=(%ICFSEQ4,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%SMSGRP NE '')
//*-----------------------------------------------------------------
//*-----|  Section 3 is used once per step (%SMSGRP = 0, 1, or 4
//*-----------------------------------------------------------------
%ENDDEL
%DELETE  (%SMSGRP EQ '2'| %SMSGRP EQ '3')
//SYSIN     DD *
%ICSYSIN
/*
%ENDDEL
```

Figure 60. IBM-Supplied Skeletal JCL for the Database Image Copy Utility and the Database Image Copy 2 Utility (Part 3 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 %SMS1DS 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 %ICDDNn DD statements are deleted if Sameds was specified on the GENJCL command.
- 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 %STPNO 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 %DBDDN 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 %ICDSNx, %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.OIC 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 61 on page 457 is a listing of OICJCL. A description of the statements in OICJCL follows Figure 61 on page 457.

```
 //OIC%STPNO EXEC PGM=DFSRRC00,PARM='BMP,DFSUICP0,%PSB,,MASTER',
//             REGION=700K
//*     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 ONLINE IMAGE COPY.
//*
//STEPLIB   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
//%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 *
%ICSYSIN
/*
```

Figure 61. 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.OIC 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 DFSUCKPT 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 62 on page 459 is a listing of ICRCVJCL. A description of the statements in ICRCVJCL follows Figure 62 on page 459.

```
//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.DBDLIB,DISP=SHR
//%DBDDN    DD DSN=%DBDSN,
%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 62. 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 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 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.

DFSVSAMP 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 63 on page 461 is a listing of RECOVJCL. A description of the statements in RECOVJCL follows Figure 63 on page 461.

```
%DELETE   (%MDBNAME NE '')
//RCV%STPNO EXEC PGM=DFSRRC00,REGION=1300K,
//             PARM='UDR,DFSURDB0,%DBNAME,,,,,,,,,,,,,,,,,,,%GSGNAME'
%ENDDEL
%DELETE   (%MDBNAME EQ  '')
//RCV%STPNO EXEC PGM=DFSRRC00,REGION=1300K,
//            PARM='UDR,DFSURDB0,%MDBNAME,,,,,,,,,,,,,,,,,,%GSGNAME'
%ENDDEL
//*
//*
//*      THIS JCL ORIGINIATES 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.
//*
//STEPLIB   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
//%DBDDN    DD DSN=%DBDSN,
%DELETE  (%DBDSAM EQ 'VSAM' | %SMS EQ '1')
//             UNIT=SYSDA,
//             VOL=SER=VOLSER,
//             SPACE=(CYL,(20,2)),
//             DISP=(NEW,KEEP),
//             DCB=BUFNO=10
%ENDDEL
%DELETE  (%DBDSAM EQ 'VSAM' | %SMS EQ '0')
//             UNIT=SYSDA,
//             VOL=SER=VOLSER,
//             SPACE=(CYL,(20,2)),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%DBDSAM NE 'VSAM')
//             DISP=OLD
%ENDDEL
%DELETE  (%ICDSN EQ '')
//DFSUDUMP  DD DSN=%ICDSN,
%ENDDEL
```

Figure 63. IBM-Supplied Skeletal JCL for the Database Recovery Utility (Part 1 of 2)

```
%DELETE  (%ICCAT EQ 'YES')
//             UNIT=%ICUNIT,
//             VOL=(PRIVATE,,,,SER=(%ICVOLS)),
//             LABEL=(%ICFSEQ,SL),
%ENDDEL
%DELETE  (%ICDSN EQ '' | %SMS EQ '1')
//             DISP=(OLD,KEEP),DCB=BUFNO=10
%ENDDEL
%DELETE  (%ICDSN EQ '' | %SMS EQ '0')
//             DISP=(OLD,KEEP)
%ENDDEL
%DELETE  (%ICDSN NE '')
//DFSUDUMP  DD DUMMY
%ENDDEL
//DFSVDUMP  DD DUMMY
%DELETE  (%CADSN EQ '')
//DFSUCUM   DD DSN=%CADSN,UNIT=%CAUNIT,
//             VOL=(PRIVATE,,,,SER=(%CAVOLS)),
//             LABEL=(%CAFSEQ,SL),
//             DISP=(OLD,KEEP),DCB=BUFNO=10
%ENDDEL
%DELETE  (%CADSN NE '')
//DFSUCUM   DD DUMMY
%ENDDEL
%SELECT  RLDS((%DBNAME,%DBDDN),FROM(%DSLLGTM))
%DELETE (%LOGVOLS EQ '')
//DFSULOG   DD DSN=%LOGDSN,UNIT=%LOGUNIT,
//             VOL=(PRIVATE,,%LOGVSEQ,,SER=(%LOGVOLS)),
//             LABEL=(%LOGFSEQ,SL),
//             DCB=RECFM=VB,
//             DISP=OLD
%ENDDEL
%DELETE (%LOGVOLS NE '')
//DFSULOG   DD DSN=%LOGDSN,DISP=OLD
%ENDDEL
%ENDSEL
%DELETE  (%LOGSEL EQ 'YES')
//DFSULOG   DD DUMMY
%ENDDEL
%DELETE  (%TRACK EQ 'NO')
//DFSTRCV   DD DSN=??????,
//             DISP=OLD
%ENDDEL
//DFSVSAMP  DD *
1024,4
4096,4
//SYSIN     DD *
%RCSYSIN
/*
```

Figure 63. IBM-Supplied Skeletal JCL for the Database Recovery Utility (Part 2 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

%CAFSEQ File sequence number

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.

```
%DELETE (%RCVFULL EQ 'YES')
//RCV%STPNO EXEC PGM=DSPURX00
//STEPLIB   DD  DSN=IMS.RESLIB,DISP=SHR
%ENDDEL
%DELETE (%RCVFULL EQ 'YES' | %RCNDSN1 EQ '')
//RECON1    DD  DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE (%RCVFULL EQ 'YES' | %RCNDSN2 EQ '')
//RECON2    DD  DSN=%RCNDSN2,DISP=SHR
```

```
%ENDDEL
%DELETE (%RCVFULL EQ 'YES' | %RCNDSN3 EQ '')
//RECON3    DD  DSN=%RCNDSN3,DISP=SHR
%ENDDEL
%DELETE (%RCVFULL EQ 'YES')
//SYSIN     DD *
CHANGE.DBDS DBD(%DBNAME) DDN(%DBDDN) ICON
/*
%ENDDEL
```

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 64 on page 466 is a listing of DSPUPJCL. A description of the statements in DSPUPJCL follows Figure 64 on page 466.

```
//UPREC%STPNO EXEC PGM=DFSRRC00,REGION=1300K,
//             PARM='ULU,DFSPREC0,%MDBNAME,,,,,,,,,,,,,,,,,,,Y,N'
//*
//*************************************************************************
//*
//*     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
/*
```

Figure 64. IBM-Supplied Skeletal JCL for the HALDB Index/ILDS Rebuild Utility

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.

SYSIN DD Statement
This DD statement contains database recovery statements that control the processing.

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 Version 9: Utilities Reference: Database and Transaction Manager*.

RECOVTYP=%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 Version 9: Utilities Reference: Database and Transaction Manager*.

Appendix B. Sample Listings of RECONs

The following listings show the format and content of various records of RECONs as they were listed by LIST.RECON commands.

In This Appendix:

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, CAGRPs 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:

Member Name	**RECON Record DSECTS**
DSPALLRC	ALLOC record
DSPBKORC	BACKOUT record
DSPCAGRC	CAGRP record
DSPCHGRC	CA record
DSPDBHRC	DB record
DSPDGRC	DBDSGRP record
DSPDSHRC	DBDS record
DSPGSGRC	Global Service Group record
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
DSPRRGRC	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 65 is an example of `LIST.HISTORY` output that includes a time line.

```
         IMS VERSION 9 RELEASE 1  DATA BASE RECOVERY CONTROL      PAGE 0001
  ------------------------------- Referenced Note ---------------------------
    LIST.HISTORY DBD(DIVNTZ04)
02.070 08:30:25.9                   LISTING OF RECON                  PAGE 0002
-------------------------------------------------------------------------------
 DB
  DBD=DIVNTZ04                 IRLMID=*NULL        DMB#=1       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
-------------------------------------------------------------------------------
02.070 08:30:25.9                   LISTING OF RECON                  PAGE 0003
-------------------------------------------------------------------------------
 DBDS
  DSN=DIVNTZ04.JJXXS01K                                         TYPE=IMS
  DBD=DIVNTZ04  DDN=DBHVSAM1 DSID=001 DBORG=HISAM  DSORG=VSAM
  CAGRP=CAGRP1    GENMAX=3     IC AVAIL=0     IC USED=1     DSSN=00000000
  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
-------------------------------------------------------------------------------
```

Figure 65. Sample LIST.HISTORY Output (Part 1 of 4)

```
-------------------------------------------------------------------------------
 ALLOC
  ALLOC   =01.111 12:15:00.0              *  ALLOC LRID =0000000000000000
  DSSN=0000000001 USID=0000000002 START = 01.111 12:00:00.0
  DEALLOC =01.111 12:59:59.9               DEALLOC LRID =0000000000000000

 IMAGE
  RUN     = 01.111 12:30:00.0              *  RECORD COUNT =0
  STOP    = 01.111 12:45:00.0                 ONLINE     USID=0000000001

 IC1
  DSN=DIVNTZ04.DBHVSAM1.IC1                       FILE SEQ=0001
  UNIT=3400                          VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=USER01

 ALLOC
  ALLOC   =01.111 13:15:00.0              *  ALLOC LRID =0000000000000000
  DSSN=0000000002 USID=0000000003 START = 01.111 12:00:00.0
  DEALLOC =01.111 13:59:59.9               DEALLOC LRID =0000000000000000

 ALLOC
  ALLOC   =01.112 12:15:00.0              *  ALLOC LRID =0000000000000000
  DSSN=0000000003 USID=0000000004 START = 01.112 12:00:00.0
  DEALLOC =01.112 12:59:59.9               DEALLOC LRID =0000000000000000

 ALLOC
  ALLOC   =01.112 13:15:00.0              *  ALLOC LRID =0000000000000000
  DSSN=0000000004 USID=0000000005 START = 01.112 12:00:00.0
  DEALLOC =01.112 13:59:59.9               DEALLOC LRID =0000000000000000
DSP0181I  NO CA             RECORD FOUND
DSP0181I  NO REORG          RECORD FOUND
DSP0181I  NO RECOV          RECORD FOUND
```

Figure 65. Sample LIST.HISTORY Output (Part 2 of 4)

```
02.070 08:30:25.9                   LISTING OF RECON                  PAGE 0004
-------------------------------------------------------------------------------
 PRILOG                                        RECORD SIZE=     464
  START = 01.111 12:00:00.0              *    SSID=IMS1     Version=9.1
  STOP  = 01.111 14:00:00.0                   #DSN=2
  GSGNAME=**NULL**
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 0

  DSN=PRILOG1.DSN1                                         UNIT=3400
  START = 01.111 12:00:00.0                FIRST DS LSN= 0000000000000001
  STOP  = 01.111 13:00:00.0                LAST  DS LSN= 0000000000000100
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=USER01 STOPTIME = 01.111 13:00:00.0
      CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
      LOCK SEQUENCE#=  2E9800000000

  DSN=PRILOG1.DSN2                                         UNIT=3400
  START = 01.111 13:00:00.0                FIRST DS LSN= 0000000000000101
  STOP  = 01.111 14:00:00.0                LAST  DS LSN= 0000000000000200
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=USER01 STOPTIME = 01.111 14:00:00.0
      CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
      LOCK SEQUENCE#=  2E9800000000
-------------------------------------------------------------------------------
02.070 08:30:25.9                   LISTING OF RECON                  PAGE 0005
-------------------------------------------------------------------------------
 PRILOG                                        RECORD SIZE=     464
  START = 01.112 12:00:00.0              *    SSID=IMS2     Version=9.1
  STOP  = 01.112 14:00:00.0                   #DSN=2
  GSGNAME=**NULL**
  FIRST RECORD ID= 0000000000000201       PRILOG TOKEN= 0

  DSN=PRILOG2.DSN1                                         UNIT=3400
  START = 01.112 12:00:00.0                FIRST DS LSN= 0000000000000201
  STOP  = 01.112 13:00:00.0                LAST  DS LSN= 0000000000000300
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=USER01 STOPTIME = 01.112 13:00:00.0
      CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
      LOCK SEQUENCE#=  2E9800000000

  DSN=PRILOG2.DSN2                                         UNIT=3400
  START = 01.112 13:00:00.0                FIRST DS LSN= 0000000000000301
  STOP  = 01.112 14:00:00.0                LAST  DS LSN= 0000000000000400
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=USER01 STOPTIME = 01.112 14:00:00.0
      CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
      LOCK SEQUENCE#=  2E9800000000
```

Figure 65. Sample LIST.HISTORY Output (Part 3 of 4)

```
-------------------------------------------------------------------------------
02.070 08:30:25.9                   LISTING OF RECON                  PAGE 0006
+------------------------------------------------------------------------------+
|Timeline for DBDS: DIVNTZ04 DBHVSAM1                                      Ref |
|                       USID=00000001   AUTHORIZED=00000000                Page|
|                    RECEIVE=00000000         HARD=00000000                0002|
+-Time------------+Events----+---+--+------------------------------------------+
|                 |IC        |   |  |                                          |
|                 |REORG     |   |US|Subsystem                                 |
|                 |RECOV     |CA |ID|Logs and Allocs                           |
+-----------------+----------+---+--+------------------------------------------+
01.111 12:00:00.0                    IMS1                                  0004
01.111 12:15:00.0                 2  A                                     0003
01.111 12:30:00.0  O              |  |                                     0003
01.111 12:45:00.0  s              |  |
01.111 12:59:59.9                 2  D
01.111 13:00:00.0                    s
01.111 13:15:00.0                 3  A                                     0003
01.111 13:59:59.9                 3  D
01.111 14:00:00.0                    C
01.112 12:00:00.0                    IMS2                                  0005
01.112 12:15:00.0                 4  A                                     0003
01.112 12:59:59.9                 4  D
01.112 13:00:00.0                    s
01.112 13:15:00.0                 5  A                                     0003
01.112 13:59:59.9                 5  D
01.112 14:00:00.0                    C
+-----------------+----------+---+--+------------------------------------------+
In the timeline, only the last digit of USID is shown.
            IC: B = Batch, U = UIC, O = OIC, C = CIC, s =  Stop
          Logs: SSID = Open time, C = Log Close,
                v = Vol close, s = DS close
        Allocs: D = Dealloc time, A = Alloc time
+------------------------------------------------------------------------------+
DSP0180I  NUMBER OF RECORDS LISTED IS        9
DSP0203I  COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I  COMMAND COMPLETION TIME 02.070 08:30:26.7
         IMS VERSION 8 RELEASE 1  DATA BASE RECOVERY CONTROL          PAGE 0007
DSP0211I  COMMAND PROCESSING COMPLETE
DSP0211I  HIGHEST CONDITION CODE = 00
```

Figure 65. Sample LIST.HISTORY Output (Part 4 of 4)

Related Reading: See “LIST.HISTORY” on page 294 for more information about the LIST.HISTORY command.

Sample Listing of a RECON at the Active Site

The following sections contain RECON record listings from an active site in an RSR environment (Figure 66 on page 474 through Figure 85 on page 522). “Fields Present in a Listing of a RECON by Record Type” on page 522 describes the fields that can be present in a listing of the RECON.

In This Section:

- “RECON Status Record” on page 474
- “Log Records” on page 475
- “GSG Record” on page 483
- “SSYS Record” on page 484
- “BACKOUT Record” on page 484
- “CAGRP and CA Records” on page 485

- “DBGRP, DBDSGRP, and RECOVGRP Records” on page 486
- “DB (IMS) and Related Records” on page 487
- “DB (HALDB and PART) and Related Records” on page 490
- “DB (FP) and Related Records” on page 499

RECON Status Record

```
03.224 12:07:43.9                   LISTING OF RECON                  PAGE 0002
-------------------------------------------------------------------------------
RECON
  RECOVERY CONTROL DATA SET, IMS V9R1
  DMB#=32767   LAST USED DMB#=32763  INIT TOKEN=03224F1825550F
  NOFORCER  LOG DSN CHECK=CHECK17    STARTNEW=NO
  TAPE UNIT=3400      DASD UNIT=3400      TRACEOFF   SSID=**NULL**
  LIST DLOG=NO                 CA/IC/LOG DATA SETS CATALOGED=NO
  MINIMUM VERSION = 7.1
  LOG RETENTION PERIOD=00.001 00:00:00.0
  COMMAND AUTH=NONE  HLQ=**NULL**
  SIZALERT DSNUM=15      VOLNUM=16     PERCENT= 95
  LOGALERT DSNUM=3       VOLNUM=16

  TIME-STAMP INFORMATION:

    TIMEZIN = %SYS

    OUTPUT FORMAT:  DEFAULT = LOCORG NONE   PUNC YY
                    CURRENT = LOCORG NONE   PUNC YY


  IMSPLEX = ** NONE **

 -DDNAME-      -STATUS-       -DATA SET NAME-
  RECON1        COPY1          IMSTESTS.DSHR.RECON1
  RECON2        COPY2          IMSTESTS.DSHR.RECON2
  RECON3        SPARE          IMSTESTS.DSHR.RECON3
```

Figure 66. Sample Listing of a RECON at the Active Site - RECON Status

Log Records

```
02.231 10:32:42.8                   LISTING OF RECON                 PAGE 0003
-------------------------------------------------------------------------------
 PRILOG                                         RECORD SIZE=     464
  START = 02.231 10:43:16.8             *    SSID=SYS3     Version=9.1
  STOP  = 02.231 10:51:11.9                  #DSN=2
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 1
  EARLIEST CHECKPOINT = 02.231 10:53:22.3

  DSN=IMSVS.RLDSP.SYS3.D02231.T1043168.V00                UNIT=SYSDA
  START = 02.231 10:43:16.8                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:47:18.9                LAST  DS LSN= 0000000000000445
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 02.231 10:47:18.9
      CKPTCT=1    CHKPT ID = 02.231 10:43:21.1
      LOCK SEQUENCE#=  000000000000

  DSN=IMSVS.RLDSP.SYS3.D02231.T1047189.V00                UNIT=SYSDA
  START = 02.231 10:47:18.9                FIRST DS LSN= 0000000000000446
  STOP  = 02.231 10:51:11.9                LAST  DS LSN= 000000000000066A
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 02.231 10:51:11.9
      CKPTCT=2    CHKPT ID = 02.231 10:51:11.5
      LOCK SEQUENCE#=  000000000000

 LOGALL
  START   = 02.231 10:43:16.8              *
  EARLIEST ALLOC TIME = 02.231 10:45:06.1
  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 67. Sample Listing of a RECON at the Active Site - Log Records (Part 1 of 9)

```
02.231 10:32:42.8                   LISTING OF RECON                  PAGE 0004
------------------------------------------------------------------------------
 PRISLD                                       RECORD SIZE=     464
  START = 02.231 10:43:16.8              *    SSID=SYS3     Version=9.1
  STOP  = 02.231 10:51:11.9                   #DSN=2
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 1

  DSN=IMSVS.SLDSP.SYS3.D02231.T1043168.V00                 UNIT=SYSDA
  START = 02.231 10:43:16.8                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:47:18.9                LAST  DS LSN= 0000000000000445
  FILE SEQ=0001    #VOLUMES=0001
  CHECKPOINT TYPES=80: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=N

    VOLSER=000000 STOPTIME = 02.231 10:47:18.9
      CKPTCT=1    CHKPT ID = 02.231 10:43:21.1
      LOCK SEQUENCE#=  000000000000

  DSN=IMSVS.SLDSP.SYS3.D02231.T1047189.V00                 UNIT=SYSDA
  START = 02.231 10:47:18.9                FIRST DS LSN= 0000000000000446
  STOP  = 02.231 10:51:11.9                LAST  DS LSN= 000000000000066A
  FILE SEQ=0001    #VOLUMES=0001
  CHECKPOINT TYPES=88: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=Y

    VOLSER=000000 STOPTIME = 02.231 10:51:11.9
      CKPTCT=2    CHKPT ID = 02.231 10:51:11.5
      LOCK SEQUENCE#=  000000000000
02.231 10:32:42.8                   LISTING OF RECON                  PAGE 0005
-------------------------------------------------------------------------------
 PRILOG                                       RECORD SIZE=     744
  START = 02.231 10:53:20.0              *    SSID=SYS3     Version=9.1
  STOP  = 02.231 11:11:22.1                   #DSN=4
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 2
  EARLIEST CHECKPOINT = 02.231 10:55:10.3

  DSN=**** COMPRESSED DATA SET ****                        UNIT=
  START = 02.231 10:53:20.0                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:54:15.1                LAST  DS LSN= 0000000000000316
  FILE SEQ=0000    #VOLUMES=0000
```

Figure 67. Sample Listing of a RECON at the Active Site - Log Records (Part 2 of 9)

```
DSN=IMSVS.RLDSP.SYS3.D02231.T1054151.V01              UNIT=SYSDA
  START = 02.231 10:54:15.1                FIRST DS LSN= 0000000000000317
  STOP  = 02.231 10:54:55.1                LAST  DS LSN= 00000000000004DE
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 02.231 10:54:55.1
      CKPTCT=1    CHKPT ID = 02.231 10:54:15.1
      LOCK SEQUENCE#=  000000000000

  DSN=IMSVS.RLDSP.SYS3.D02231.T1054551.V00                UNIT=SYSDA
  START = 02.231 10:54:55.1                FIRST DS LSN= 00000000000004DF
  STOP  = 02.231 10:55:16.8                LAST  DS LSN= 00000000000005F6
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 02.231 10:55:16.8
      CKPTCT=1    CHKPT ID = 02.231 10:55:10.3
      LOCK SEQUENCE#=  000000000000

  DSN=IMSVS.RLDSP.SYS3.D02231.T1055168.V01                UNIT=SYSDA
  START = 02.231 10:55:16.8                FIRST DS LSN= 00000000000005F7
  STOP  = 02.231 11:11:22.1                LAST  DS LSN= 000000000000080D
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 02.231 11:11:22.1
      CKPTCT=2    CHKPT ID = 02.231 11:11:21.2
      LOCK SEQUENCE#=  000000000000

 LOGALL
  START   = 02.231 10:53:20.0              *
  EARLIEST ALLOC TIME = 02.231 10:54:51.4
  DBDS ALLOC=1                        -DBD-    -DDN-    -ALLOC-
                                      DBVHDJ05 CJVHDG1E  1
02.231 10:32:42.8                   LISTING OF RECON                  PAGE 0006
-------------------------------------------------------------------------------
 PRISLD                                        RECORD SIZE=     744
  START = 02.231 10:53:20.0             *    SSID=SYS3     Version=9.1
  STOP  = 02.231 11:11:22.1                  #DSN=4
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 2

  DSN=**** COMPRESSED DATA SET ****                       UNIT=
  START = 02.231 10:53:20.0                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:54:15.1                LAST  DS LSN= 0000000000000316
  FILE SEQ=0000    #VOLUMES=0000
  CHECKPOINT TYPES=80: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=N

  DSN=IMSVS.SLDSP.SYS3.D02231.T1054151.V01                UNIT=SYSDA
  START = 02.231 10:54:15.1                FIRST DS LSN= 0000000000000317
  STOP  = 02.231 10:54:55.1                LAST  DS LSN= 00000000000004DE
  FILE SEQ=0001    #VOLUMES=0001
  CHECKPOINT TYPES=80: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=N

    VOLSER=000000 STOPTIME = 02.231 10:54:55.1
      CKPTCT=1    CHKPT ID = 02.231 10:54:15.1
      LOCK SEQUENCE#=  000000000000
```

Figure 67. Sample Listing of a RECON at the Active Site - Log Records (Part 3 of 9)

```
DSN=IMSVS.SLDSP.SYS3.D02231.T1054551.V00               UNIT=SYSDA
  START = 02.231 10:54:55.1                FIRST DS LSN= 00000000000004DF
  STOP  = 02.231 10:55:16.8                LAST  DS LSN= 00000000000005F6
  FILE SEQ=0001    #VOLUMES=0001
  CHECKPOINT TYPES=40: SIMPLE=N SNAPQ=Y DUMPQ=N PURGE=N FREEZE=N

    VOLSER=000000 STOPTIME = 02.231 10:55:16.8
      CKPTCT=1    CHKPT ID = 02.231 10:55:10.3
      LOCK SEQUENCE#=  000000000000

  DSN=IMSVS.SLDSP.SYS3.D02231.T1055168.V01                 UNIT=SYSDA
  START = 02.231 10:55:16.8                FIRST DS LSN= 00000000000005F7
  STOP  = 02.231 11:11:22.1                LAST  DS LSN= 000000000000080D
  FILE SEQ=0001    #VOLUMES=0001
  CHECKPOINT TYPES=88: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=Y

    VOLSER=000000 STOPTIME = 02.231 11:11:22.1
      CKPTCT=2    CHKPT ID = 02.231 11:11:21.2
      LOCK SEQUENCE#=  000000000000
02.231 10:32:42.8                   LISTING OF RECON                  PAGE 0007
-------------------------------------------------------------------------------
 PRILOG                                       RECORD SIZE=     304
  START = 02.231 11:11:58.9              *    SSID=BATCH1   Version=9.1
  STOP  = 02.231 11:12:00.5                   #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 3

  DSN=BATCH1.UPDATEF.LOG                                   UNIT=SYSDA
  START = 02.231 11:11:58.9                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 11:12:00.5                LAST  DS LSN= 000000000000009C
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 02.231 11:12:00.5
     CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
      LOCK SEQUENCE#=  000000000000

 LOGALL
  START   = 02.231 11:11:58.9              *
  EARLIEST ALLOC TIME = 02.231 11:11:59.7
  DBDS ALLOC=3                        -DBD-    -DDN-    -ALLOC-
                                      DHVNTZ02 HIDAM     1
                                      DXVNTZ02 XDLBT04I  1
                                      DIVNTZ02 DBHVSAM1  1
```

Figure 67. Sample Listing of a RECON at the Active Site - Log Records (Part 4 of 9)

```
02.231 10:32:42.8                   LISTING OF RECON                 PAGE 0008
-------------------------------------------------------------------------------
 SECLOG                                       RECORD SIZE=     304
  START = 02.231 11:11:58.9              *    SSID=BATCH1   Version=9.1
  STOP  = 02.231 11:12:00.5                   #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 3

  DSN=IMSTESTL.IMS01.OLDSP1                                UNIT=SYSDA
  START = 02.231 11:11:58.9                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 11:12:00.5                LAST  DS LSN= 000000000000009C
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=USER03 STOPTIME = 02.231 11:12:00.5
      CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
      LOCK SEQUENCE#=  000000000000
02.231 10:32:42.8                   LISTING OF RECON                  PAGE 0009
-------------------------------------------------------------------------------
 PRILOG                                       RECORD SIZE=     304
  START = 02.231 11:12:19.1              *    SSID=BATFPC   Version=9.1
  STOP  = 02.231 11:12:27.2                   #DSN=1
  GSGNAME=**NULL**
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 0

  DSN=IMSTESTM.BAT01.DBLOG1                                UNIT=SYSDA
  START = 02.231 11:12:19.1                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 11:12:27.2                LAST  DS LSN= 00000000000000E2
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=222222 STOPTIME = 02.231 11:12:27.2
      CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
      LOCK SEQUENCE#=  000000000000

 LOGALL
  START   = 02.231 11:12:19.1              *
  EARLIEST ALLOC TIME = 02.231 11:12:19.9
  DBDS ALLOC=1                        -DBD-    -DDN-    -ALLOC-
                                      DBOVLFPC VLOSAM01  1
02.231 10:32:42.8                   LISTING OF RECON                  PAGE 0010
-------------------------------------------------------------------------------
 PRILOG                                       RECORD SIZE=     304
  START = 02.231 11:12:43.6              *    SSID=BATFPC   Version=9.1
  STOP  = 02.231 11:12:44.7                   #DSN=1
  GSGNAME=**NULL**                             BBO
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 0

  DSN=IMSTESTM.BBOFPC.BB1                                  UNIT=SYSDA
  START = 02.231 11:12:43.6                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 11:12:44.7                LAST  DS LSN= 000000000000002E
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=333333 STOPTIME = 02.231 11:12:44.7
      CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
      LOCK SEQUENCE#=  000000000000

 LOGALL
  START   = 02.231 11:12:43.6              *
  EARLIEST ALLOC TIME = 02.231 11:12:44.4
  DBDS ALLOC=1                        -DBD-    -DDN-    -ALLOC-
                                      DBOVLFPC VLOSAM01  1
```

Figure 67. Sample Listing of a RECON at the Active Site - Log Records (Part 5 of 9)

```
02.231 10:32:42.8                   LISTING OF RECON                 PAGE 0011
-------------------------------------------------------------------------------
 PRILOG                                       RECORD SIZE=     304
  START = 02.231 11:14:06.8              *    SSID=SYS3     Version=9.1
  STOP  = 02.231 10:28:00.1                   #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001      PRILOG TOKEN= 4
  EARLIEST CHECKPOINT = 02.231 10:55:10.3

  DSN=IMSVS.RLDSP.SYS3.D02231.T1114068.V02                 UNIT=SYSDA
  START = 02.231 11:14:06.8                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:28:00.1                LAST  DS LSN= 000000000000027A
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 02.231 10:28:00.1
      CKPTCT=1    CHKPT ID = 02.231 11:14:09.5
      LOCK SEQUENCE#=  000000000000

 LOGALL
  START   = 02.231 11:14:06.8              *
  EARLIEST ALLOC TIME = 02.231 11:14:10.0
  DBDS ALLOC=7                        -DBD-    -DDN-    -ALLOC-
                                      DEDBDD01 DD01AR0   1
                                      DHVNTZ02 HIDAM     1
                                      DXVNTZ02 XDLBT04I  1
                                      DIVNTZ02 DBHVSAM1  1
                                      DBOHIDK5 CKOHIG1O  1
                                      DXVHIDK5 CKVHIIXK  1
                                      DBVHDJ05 CJVHDG1E  1
02.231 10:32:42.8                   LISTING OF RECON                  PAGE 0012
-------------------------------------------------------------------------------
 PRISLD                                       RECORD SIZE=     304
  START = 02.231 11:14:06.8              *    SSID=SYS3     Version=9.1
  STOP  = 02.231 10:28:00.1                   #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001      PRILOG TOKEN= 4

  DSN=IMSVS.SLDSP.SYS3.D02231.T1114068.V02                 UNIT=SYSDA
  START = 02.231 11:14:06.8                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:28:00.1                LAST  DS LSN= 000000000000027A
  FILE SEQ=0001    #VOLUMES=0001
  CHECKPOINT TYPES=80: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=N

    VOLSER=000000 STOPTIME = 02.231 10:28:00.1
      CKPTCT=1    CHKPT ID = 02.231 11:14:09.5
      LOCK SEQUENCE#=  000000000000
```

Figure 67. Sample Listing of a RECON at the Active Site - Log Records (Part 6 of 9)

```
02.231 10:32:42.8                   LISTING OF RECON                  PAGE 0013
-------------------------------------------------------------------------------
 PRILOG                                       RECORD SIZE=     304
  START = 02.231 10:28:00.2              *    SSID=SYS3     Version=9.1
  STOP  = 00.000 00:00:00.0                   #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 000000000000027B      PRILOG TOKEN= 5
  EARLIEST CHECKPOINT = 02.231 10:28:03.7

  DSN=IMSVS.RLDSP.SYS3.D02231.T1028002.V01                 UNIT=SYSDA
  START = 02.231 10:28:00.2                FIRST DS LSN= 000000000000027B
  STOP  = 02.231 10:30:16.3                LAST  DS LSN= 0000000000000400
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 02.231 10:30:16.3
      CKPTCT=2    CHKPT ID = 02.231 10:28:03.7
      LOCK SEQUENCE#=  000000000000

 LOGALL
  START   = 02.231 10:28:00.2              *
  EARLIEST ALLOC TIME = 02.231 10:28:05.9
  DBDS ALLOC=1                        -DBD-    -DDN-    -ALLOC-
                                      DEDBDD01 DD01AR0   1
02.231 10:32:42.8                   LISTING OF RECON                  PAGE 0014
-------------------------------------------------------------------------------
 PRISLD                                       RECORD SIZE=     304
  START = 02.231 10:28:00.2              *    SSID=SYS3     Version=9.1
  STOP  = 00.000 00:00:00.0                   #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 000000000000027B      PRILOG TOKEN= 5

  DSN=IMSVS.SLDSP.SYS3.D02231.T1028002.V01                 UNIT=SYSDA
  START = 02.231 10:28:00.2                FIRST DS LSN= 000000000000027B
  STOP  = 02.231 10:30:16.3                LAST  DS LSN= 0000000000000400
  FILE SEQ=0001    #VOLUMES=0001
  CHECKPOINT TYPES=C0: SIMPLE=Y SNAPQ=Y DUMPQ=N PURGE=N FREEZE=N

    VOLSER=000000 STOPTIME = 02.231 10:30:16.3
      CKPTCT=2    CHKPT ID = 02.231 10:28:03.7
      LOCK SEQUENCE#=  000000000000
```

Figure 67. Sample Listing of a RECON at the Active Site - Log Records (Part 7 of 9)

```
02.231 10:32:42.8                   LISTING OF RECON                 PAGE 0015
-------------------------------------------------------------------------------
 PRILOG                                       RECORD SIZE=     304
  START = 02.231 10:31:36.3              *    SSID=IMS2     Version=9.1
  STOP  = 02.231 10:32:14.6                   #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001      PRILOG TOKEN= 6
  EARLIEST CHECKPOINT = 02.231 10:31:40.5

  DSN=IMSVS.RLDSP.IMS2.D02231.T1031363.V00                 UNIT=SYSDA
  START = 02.231 10:31:36.3                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:32:14.6                LAST  DS LSN= 00000000000001E0
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 02.231 10:32:14.6
      CKPTCT=2    CHKPT ID = 02.231 10:32:14.1
      LOCK SEQUENCE#=  000000000000

 LOGALL
  START   = 02.231 10:31:36.3              *
  EARLIEST ALLOC TIME = 00.000 00:00:00.0
  DBDS ALLOC=0
02.231 10:32:42.8                   LISTING OF RECON                  PAGE 0016
-------------------------------------------------------------------------------
 PRISLD                                       RECORD SIZE=     304
  START = 02.231 10:31:36.3              *    SSID=IMS2     Version=9.1
  STOP  = 02.231 10:32:14.6                   #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001      PRILOG TOKEN= 6

  DSN=IMSVS.SLDSP.IMS2.D02231.T1031363.V00                 UNIT=SYSDA
  START = 02.231 10:31:36.3                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:32:14.6                LAST  DS LSN= 00000000000001E0
  FILE SEQ=0001    #VOLUMES=0001
  CHECKPOINT TYPES=88: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=Y

    VOLSER=000000 STOPTIME = 02.231 10:32:14.6
      CKPTCT=2    CHKPT ID = 02.231 10:32:14.1
      LOCK SEQUENCE#=  000000000000
02.231 10:32:42.8                   LISTING OF RECON                  PAGE 0017
-------------------------------------------------------------------------------
 PRIOLD
  SSID=IMS2            # DD ENTRIES=1
  EARLIEST CHECKPOINT = 02.231 10:31:40.5

  DDNAME=DFSOLP00   DSN=IMSTESTL.IMS02.OLDSP0
  START = 02.231 10:31:36.3                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:32:14.6                LAST  DS LSN= 00000000000001E0
  LOCK SEQUENCE# = 000000000000
  STATUS=ARC COMPLT                                FEOV=NO    AVAIL
  PRILOG TIME=02.231 10:31:36.3                ARCHIVE JOB NAME=JT103215
  Version=9.1
```

Figure 67. Sample Listing of a RECON at the Active Site - Log Records (Part 8 of 9)

```
02.231 10:32:42.8                  LISTING OF RECON                PAGE 0018
-------------------------------------------------------------------------------
 PRIOLD
  SSID=SYS3            # DD ENTRIES=4
  EARLIEST CHECKPOINT = 02.231 10:28:03.7

  DDNAME=DFSOLP03   DSN=IMSTESTL.IMS01.OLDSP3
  START = 02.231 10:55:16.8                FIRST DS LSN= 00000000000005F7
  STOP  = 02.231 11:11:21.4                LAST  DS LSN= 000000000000079E
  LOCK SEQUENCE# = 000000000000
  STATUS=ARC COMPLT                                FEOV=NO    AVAIL
  PRILOG TIME=02.231 10:53:20.0                ARCHIVE JOB NAME=JT102801
  Version=9.1

  DDNAME=DFSOLP01   DSN=IMSTESTL.IMS01.OLDSP1
  START = 02.231 11:14:06.8                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:28:00.1                LAST  DS LSN= 000000000000027A
  LOCK SEQUENCE# = 000000000000
  STATUS=ARC COMPLT                                FEOV=NO    AVAIL
  PRILOG TIME=02.231 11:14:06.8                ARCHIVE JOB NAME=JT102801
  Version=9.1

  DDNAME=DFSOLP02   DSN=IMSTESTL.IMS01.OLDSP2
  START = 02.231 10:28:00.2                FIRST DS LSN= 000000000000027B
  STOP  = 02.231 10:30:16.3                LAST  DS LSN= 0000000000000400
  LOCK SEQUENCE# = 000000000000
  STATUS=ARC COMPLT                                FEOV=NO    AVAIL
  PRILOG TIME=02.231 10:28:00.2                ARCHIVE JOB NAME=JT103016
  Version=9.1

  DDNAME=DFSOLP00   DSN=IMSTESTL.IMS01.OLDSP0
  START = 02.231 10:30:16.3                FIRST DS LSN= 0000000000000401
  STOP  = 00.000 00:00:00.0                LAST  DS LSN= 0000000000000000
  LOCK SEQUENCE# = 000000000000
  STATUS=ACTIVE                                    FEOV=NO    AVAIL
  PRILOG TIME=02.231 10:28:00.2
  Version=9.1
DSP0260I  NO INTERIM RLDS/SLDS RECORDS FOUND IN RECON
DSP0260I  NO INT-ONLINE LOG RECORDS FOUND IN RECON
```

Figure 67. Sample Listing of a RECON at the Active Site - Log Records (Part 9 of 9)

GSG Record

```
1999.251 11:46:24.1 -08:00          LISTING OF RECON
-------------------------------------------------------------------------------
 GSG
  GSGNAME=IMSGSG1      #SGS=2      -SGNAME-      -ROLE-
                                   STLSITE1     ACTIVE       LOCAL
                                   STLSITE2     TRACKING
  CURRENT PRILOG TOKEN      = 6           TAKEOVER TOKEN = 0
  MINIMUM PRILOG TOKEN      = 1           DSN SEQ NUMBER = 0
  START TIME OF CURRENT LOG = 1999.251 11:44:49.0 -08:00
  HIGHEST ACTIVE SITE TIME  = 0000.000 00:00:00.0 +00:00
  TRACKING SUBSYSTEM ID     = **NULL**
  TAKEOVER IN PROGRESS
```

Figure 68. Sample Listing of a RECON at the Active Site - GSG Record

SSYS Record

```
02.022 15:55:01.7                   LISTING OF RECON                  PAGE 0020
--------------------------------------------------------------------------------
 SSYS
  SSID=SYS3      LOG START=02.022 15:49:42.8
  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
                                                    ENCODED
    -DBD-        -AREA-   -LEVEL-  -ACCESS INTENT-  -STATE-
    DEDBDD01     DD01AR0     1        UPDATE           6
```

Figure 69. Sample Listing of a RECON at the Active Site - SSYS Record

BACKOUT Record

```
1999.251 11:46:24.1 -08:00          LISTING OF RECON
--------------------------------------------------------------------------------
 BACKOUT
  SSID=SYS3      #UORS=2

    RECOVERY TOKEN=E2E8E2F3404040400000000300000002
    TIME=1999.251 11:38:22.5 -08:00       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=E2E8E2F3404040400000000400000000
    TIME=1999.251 11:38:22.8 -08:00       PSB=PSBEJK05
                INFLT   BMP   COLDEND
      ASSOCIATED DATA BASES=2

                   BACKED   DYN BKOUT
      -DBD-        -OUT -   -FAILURE-
      DBOHIDK5      NO         NO
      DXVHIDK5      NO         NO
```

Figure 70. Sample Listing of a RECON at the Active Site - BACKOUT Record

CAGRP and CA Records

```
1999.251 11:46:24.1 -08:00          LISTING OF RECON
-------------------------------------------------------------------------------
 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=IMSVS.CAGRP1.CA.CA194501                     FILE SEQ=1
  CAGRP=CAGRP1       STOP    = 1999.251 11:10:59.7 -08:00     *
                     UNIT=SYSDA      VOLS DEF=1    VOLS USED=1
                                     VOLSER=222222
  RUN     = 1999.251 11:45:45.0 -08:00
   DBD=DEDBJN21 DDN=DB21AR1  PURGETIME = 1999.251 11:08:45.9 -08:00
     CHANGES ACCUMULATED=YES COMPLETE CA=YES INDOUBT EEQES=NO
     LSN    = 000000000000      DSSN   = 0000000001
     LRID   = 0000000000000424  USID   = 0000000002
   DBD=DEDBJN21 DDN=DB21AR3  PURGETIME = 1999.251 11:08:47.9 -08:00
     CHANGES ACCUMULATED=YES COMPLETE CA=YES INDOUBT EEQES=NO
     LSN    = 000000000000      DSSN   = 0000000001
     LRID   = 000000000000043A  USID   = 0000000002
   DBD=DEDBJN21 DDN=DB21AR6  PURGETIME = 1999.251 11:02:53.0 -08:00
     CHANGES ACCUMULATED=NO  COMPLETE CA=YES INDOUBT EEQES=NO
     LSN    = 000000000000      DSSN   = 0000000000
     LRID   = 0000000000000000  USID   = 0000000000
   DBD=DEDBJN21 DDN=DB21AR7  PURGETIME = 1999.251 11:02:53.0 -08:00
     CHANGES ACCUMULATED=NO  COMPLETE CA=YES INDOUBT EEQES=NO
     LSN    = 000000000000      DSSN   = 0000000000
     LRID   = 0000000000000000  USID   = 0000000000
```

Figure 71. Sample Listing of a RECON at the Active Site - CAGRP and CA Records

DBGRP, DBDSGRP, and RECOVGRP Records

```
1999.251 11:46:24.1 -08:00          LISTING OF RECON
--------------------------------------------------------------------------------
 DBGRP
  GRPNAME=DBGRP1               #MEMBERS=6    -DBD/AREA-
                                              DIVNTZ02
                                              DHVNTZ02
                                              DXVNTZ02
                                              DB21AR0
                                              DB21AR1
                                              DB21AR2

 DBDSGRP
  GRPNAME=FJKGRP               #MEMBERS=5     -DBD-    -DDN/AREA-
                                              DIVNTZ02  DBHVSAM1
                                              DIVNTZ02  DBHVSAM2
                                              DHVNTZ02  HIDAM
                                              DHVNTZ02  HIDAM2
                                              DXVNTZ02  XDLBT04I

 RECOVGRP
  GRPNAME=RCVGRP1              #MEMBERS=5     -DBD-     -AREA-
                                              DIVNTZ02
                                              DHVNTZ02
                                              DXVNTZ02
                                              DEDBJN21  DB21AR0
                                              DEDBJN21  DB21AR1
```

Figure 72. Sample Listing of a RECON at the Active Site - DBGRP, DBDSGRP, and RECOVGRP Records

DB (IMS) and Related Records

```
1999.251 11:46:24.1 -08:00          LISTING OF RECON
-------------------------------------------------------------------------------
 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        =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

 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     DSSN=00000005
  NOREUSE         RECOVPD=0
  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   =1999.251 11:18:26.4 -08:00     *  ALLOC LRID =0000000000000000
  DSSN=0000000004 USID=0000000005 START = 1999.251 11:16:52.6 -08:00
  DEALLOC =1999.251 11:18:52.6 -08:00      DEALLOC LRID =0000000000000000

 ALLOC
  ALLOC   =1999.251 11:39:00.8 -08:00     *  ALLOC LRID =0000000000000000
  DSSN=0000000005 USID=0000000006 START = 1999.251 11:37:33.0 -08:00

 IMAGE
  RUN     = 1999.251 11:18:13.1 -08:00     *  RECORD COUNT =33
  STOP    = 0000.000 00:00:00.0 +00:00        BATCH      USID=0000000004
```

Figure 73. Sample Listing of a RECON at the Active Site - DB (IMS) and Related Records (Part 1 of 3)

```
IC1
 DSN=IMSVS.DBVHDJ05.CJVHDG1E.IC.IC121810          FILE SEQ=0001
 UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                    VOLSER=222222


IMAGE
 RUN     = 1999.251 11:19:26.0 -08:00     *  RECORD COUNT =33
 STOP    = 1999.251 11:19:26.6 -08:00        CONCUR     USID=0000000005

IC1
 DSN=IMSVS.DBVHDJ05.CJVHDG1E.IC.IC121920          FILE SEQ=0001
 UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                    VOLSER=222222
```

Figure 73. Sample Listing of a RECON at the Active Site - DB (IMS) and Related Records (Part 2 of 3)

```
------------------------------------------------------
01.150 14:03:52.9                   LISTING OF RECON
----------------------------------------------------------------
DB
 DBD=DHONTZ04          IRLMID=*NULL        DMB#=1       TYPE=IMS
 SHARE LEVEL=3        GSGNAME=**NULL**     USID=0000000001
 AUTHORIZED USID=0000000  RECEIVE USID=0000000 HARD USID=0000000
 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           =NO          HELD AUTHORIZATION STATE=0
                                      EEQE COUNT              =0
   TRACKING SUSPENDED    =NO          RECEIVE REQUIRED COUNT  =0
   OFR REQUIRED          =NO
   IC NEEDED DISABLED
----------------------------------------------------------------
01.150 14:03:52.9               LISTING OF RECON
----------------------------------------------------------------
 DBDS
  DSN=DHONTZ04.JKXXI01O                                 TYPE=IMS
  DBD=DHONTZ04  DDN=HIDAM    DSID=001 DBORG=HIDAM  DSORG=OSAM
  CAGRP=**NULL**  GENMAX=3  IC AVAIL=0  IC USED=0  DSSN=00000000
  NOREUSE         RECOVPD=0
  DEFLTJCL=**NULL** ICJCL=ICJCL OICJCL=OICJCL  RECOVJCL=RECOVJCL
  RECVJCL=ICRCVJCL
  FLAGS:                             COUNTERS:
    IC NEEDED      =OFF
    RECOV NEEDED   =OFF
    RECEIVE NEEDED =OFF                EEQE COUNT             =0
----------------------------------------------------------------
01.150 14:03:52.9                   LISTING OF RECON
----------------------------------------------------------------
 DBDS
  DSN=DHONTZ04.JKXXI02O                                 TYPE=IMS
  DBD=DHONTZ04  DDN=HIDAM2   DSID=002 DBORG=HIDAM  DSORG=OSAM
  CAGRP=**NULL**  GENMAX=3  IC AVAIL=0  IC USED=0  DSSN=00000000
  NOREUSE         RECOVPD=0
  DEFLTJCL=**NULL**  ICJCL=ICJCL OICJCL=OICJCL RECOVJCL=RECOVJCL
  RECVJCL=ICRCVJCL
  FLAGS:                             COUNTERS:
    IC NEEDED      =OFF
    RECOV NEEDED   =OFF
    RECEIVE NEEDED =OFF                EEQE COUNT             =0
```

Figure 73. Sample Listing of a RECON at the Active Site - DB (IMS) and Related Records (Part 3 of 3)

DB (HALDB and PART) and Related Records

```
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0014
-------------------------------------------------------------------------------
 DB
  DBD=DBHIDJ03                        DMB#=2      CHANGE#=5     TYPE=HALDB
  SHARE LEVEL=3               GSGNAME=IMSGSG1
  PSNAME=**NULL**  DBORG=PHIDAM   DSORG=OSAM    CURRENT PARTITION ID=00004
  FLAGS:                             COUNTERS:
    RECOVERABLE           =YES         PARTITIONS              =4
    ONLINE REORG CAPABLE  =YES         DATA SET GROUP MEMBERS  =2
-------------------------------------------------------------------------------
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0015
-------------------------------------------------------------------------------
 DB
  DBD=POHIDJA   MASTER DB=DBHIDJ03  IRLMID=*NULL  CHANGE#=2     TYPE=PART
  USID=0000000003  AUTHORIZED USID=0000000003  HARD USID=0000000003
  RECEIVE USID=0000000003  RECEIVE NEEDED USID=0000000000
  DBRCVGRP=**NULL**
  DSN PREFIX=IMSTESTS.DBHIDJ03                          PARTITION ID=00001
  PREVIOUS PARTITION=**NULL**    NEXT PARTITION=POHIDJB
  OLRIMSID=**NULL**    ACTIVE DBDS=A-J

  FREE SPACE:
    FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0

  PARTITION HIGH KEY/STRING (CHAR):               (LENGTH=5  )
    J2000
  PARTITION HIGH KEY/STRING (HEX):
    D1F2F0F0F0404040404040404040404040404040404040404040404040404040

  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
                                       HELD AUTHORIZATION STATE=0
    DATABASE LEVEL TRACK  =YES         EEQE COUNT              =0
    TRACKING SUSPENDED    =NO          RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =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
```

Figure 74. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 1 of 9)

```
-------------------------------------------------------------------------------
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0016
-------------------------------------------------------------------------------
 DBDS
  DSN=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=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
-------------------------------------------------------------------------------

 IMAGE
  RUN     = 03.287 11:22:19.0              *  RECORD COUNT =120
  STOP    = 00.000 00:00:00.0                 BATCH      USID=0000000003

 IC1
  DSN=IMSVS.POHIDJA.POHIDJAA.IC.IC112153           FILE SEQ=0001
  UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=222222


03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0017
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.B00001                                  TYPE=PART
  DBD=POHIDJA   DDN=POHIDJAB DSID=006 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
-------------------------------------------------------------------------------

 IMAGE
  RUN     = 03.287 11:23:05.7              *  RECORD COUNT =120
  STOP    = 00.000 00:00:00.0                 BATCH      USID=0000000003

 IC1
  DSN=IMSVS.POHIDJA.POHIDJAB.IC.IC112154           FILE SEQ=0001
  UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=222222


03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0018
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.L00001                                  TYPE=PART
  DBD=POHIDJA   DDN=POHIDJAL DSID=003 DBORG=INDEX  DSORG=VSAM
                                  OTHER DDN=**NULL**
  FLAGS:                             COUNTERS:
    RECOV NEEDED   =OFF                EEQE COUNT              =0
```

Figure 74. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 2 of 9)

```
-------------------------------------------------------------------------------
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0019
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.X00001                                  TYPE=PART
  DBD=POHIDJA   DDN=POHIDJAX DSID=005 DBORG=INDEX  DSORG=VSAM
                                  OTHER DDN=**NULL**
  FLAGS:                             COUNTERS:
    RECOV NEEDED   =OFF                EEQE COUNT              =0
-------------------------------------------------------------------------------
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0020
-------------------------------------------------------------------------------
 DB
  DBD=POHIDJB   MASTER DB=DBHIDJ03  IRLMID=*NULL  CHANGE#=3     TYPE=PART
  USID=0000000003  AUTHORIZED USID=0000000003  HARD USID=0000000003
  RECEIVE USID=0000000003  RECEIVE NEEDED USID=0000000000
  DBRCVGRP=**NULL**
  DSN PREFIX=IMSTESTS.DBHIDJ03                          PARTITION ID=00002
  PREVIOUS PARTITION=POHIDJA     NEXT PARTITION=POHIDJC
  OLRIMSID=**NULL**    ACTIVE DBDS=A-J

  FREE SPACE:
    FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0

  PARTITION HIGH KEY/STRING (CHAR):               (LENGTH=5  )
    J4000
  PARTITION HIGH KEY/STRING (HEX):
    D1F4F0F0F0404040404040404040404040404040404040404040404040404040

  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
                                       HELD AUTHORIZATION STATE=0
    DATABASE LEVEL TRACK  =YES         EEQE COUNT              =0
    TRACKING SUSPENDED    =NO          RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =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
```

Figure 74. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 3 of 9)

```
-------------------------------------------------------------------------------
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0021
-------------------------------------------------------------------------------
 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
-------------------------------------------------------------------------------

 IMAGE
  RUN     = 03.287 11:23:42.4              *  RECORD COUNT =60
  STOP    = 00.000 00:00:00.0                 BATCH      USID=0000000003

 IC1
  DSN=IMSVS.POHIDJB.POHIDJBA.IC.IC112155           FILE SEQ=0001
  UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=222222


03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0022
-------------------------------------------------------------------------------
 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     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
-------------------------------------------------------------------------------

 IMAGE
  RUN     = 03.287 11:24:19.6              *  RECORD COUNT =60
  STOP    = 00.000 00:00:00.0                 BATCH      USID=0000000003

 IC1
  DSN=IMSVS.POHIDJB.POHIDJBB.IC.IC112155           FILE SEQ=0001
  UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=222222


03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0023
-------------------------------------------------------------------------------
 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
```

Figure 74. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 4 of 9)

```
-------------------------------------------------------------------------------
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0024
-------------------------------------------------------------------------------
 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
-------------------------------------------------------------------------------
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0025
-------------------------------------------------------------------------------
 DB
  DBD=POHIDJC   MASTER DB=DBHIDJ03  IRLMID=*NULL  CHANGE#=4     TYPE=PART
  USID=0000000001  AUTHORIZED USID=0000000001  HARD USID=0000000001
  RECEIVE USID=0000000001  RECEIVE NEEDED USID=0000000000
  DBRCVGRP=**NULL**
  DSN PREFIX=IMSTESTS.DBHIDJ03                          PARTITION ID=00003
  PREVIOUS PARTITION=POHIDJB     NEXT PARTITION=POHIDJD
  OLRIMSID=**NULL**    ACTIVE DBDS=A-J

  FREE SPACE:
    FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0

  PARTITION HIGH KEY/STRING (CHAR):               (LENGTH=5  )
    J6000
  PARTITION HIGH KEY/STRING (HEX):
    D1F6F0F0F0404040404040404040404040404040404040404040404040404040

  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
                                       HELD AUTHORIZATION STATE=0
    DATABASE LEVEL TRACK  =YES         EEQE COUNT              =0
    TRACKING SUSPENDED    =NO          RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =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
```

Figure 74. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 5 of 9)

```
-------------------------------------------------------------------------------
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0026
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.A00003                                  TYPE=PART
  DBD=POHIDJC   DDN=POHIDJCA DSID=001 DBORG=HIDAM  DSORG=OSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     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
-------------------------------------------------------------------------------

 IMAGE
  RUN     = 03.287 11:24:57.2              *  RECORD COUNT =2
  STOP    = 00.000 00:00:00.0                 BATCH      USID=0000000001

 IC1
  DSN=IMSVS.POHIDJC.POHIDJCA.IC.IC112156           FILE SEQ=0001
  UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=222222



 REORG
  RUN     = 03.287 10:47:56.4              *    USID = 0000000001
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0027
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.B00003                                  TYPE=PART
  DBD=POHIDJC   DDN=POHIDJCB DSID=006 DBORG=HIDAM  DSORG=OSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     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
-------------------------------------------------------------------------------

 IMAGE
  RUN     = 03.287 11:25:37.6              *  RECORD COUNT =0
  STOP    = 00.000 00:00:00.0                 BATCH      USID=0000000001

 IC1
  DSN=IMSVS.POHIDJC.POHIDJCB.IC.IC112156           FILE SEQ=0001
  UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=222222



 REORG
  RUN     = 03.287 10:47:56.4              *    USID = 0000000001
```

Figure 74. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 6 of 9)

```
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0028
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.L00003                                  TYPE=PART
  DBD=POHIDJC   DDN=POHIDJCL DSID=003 DBORG=INDEX  DSORG=VSAM
                                  OTHER DDN=**NULL**
  FLAGS:                             COUNTERS:
    RECOV NEEDED   =OFF                EEQE COUNT              =0
-------------------------------------------------------------------------------
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0029
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.X00003                                  TYPE=PART
  DBD=POHIDJC   DDN=POHIDJCX DSID=005 DBORG=INDEX  DSORG=VSAM
OTHER DDN=**NULL**
  FLAGS:                             COUNTERS:
    RECOV NEEDED   =OFF                EEQE COUNT              =0
-------------------------------------------------------------------------------
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0030
-------------------------------------------------------------------------------
 DB
  DBD=POHIDJD   MASTER DB=DBHIDJ03  IRLMID=*NULL  CHANGE#=5     TYPE=PART
  USID=0000000001  AUTHORIZED USID=0000000001  HARD USID=0000000001
  RECEIVE USID=0000000001  RECEIVE NEEDED USID=0000000000
  DBRCVGRP=**NULL**
  DSN PREFIX=IMSTESTS.DBHIDJ03                          PARTITION ID=00004
  PREVIOUS PARTITION=POHIDJC     NEXT PARTITION=**NULL**
  OLRIMSID=**NULL**    ACTIVE DBDS=A-J

  FREE SPACE:
    FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0

  PARTITION HIGH KEY/STRING (CHAR):               (LENGTH=5  )
    .....
  PARTITION HIGH KEY/STRING (HEX):
    FFFFFFFFFF404040404040404040404040404040404040404040404040404040

  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
                                       HELD AUTHORIZATION STATE=0
    DATABASE LEVEL TRACK  =YES         EEQE COUNT              =0
    TRACKING SUSPENDED    =NO          RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =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
```

Figure 74. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 7 of 9)

```
-------------------------------------------------------------------------------
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0031
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.A00004                                  TYPE=PART
  DBD=POHIDJD   DDN=POHIDJDA DSID=001 DBORG=HIDAM  DSORG=OSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     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
-------------------------------------------------------------------------------

 IMAGE
  RUN     = 03.287 11:26:12.4              *  RECORD COUNT =2
  STOP    = 00.000 00:00:00.0                 BATCH      USID=0000000001

 IC1
  DSN=IMSVS.POHIDJD.POHIDJDA.IC.IC112157           FILE SEQ=0001
  UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=222222



 REORG
  RUN     = 03.287 10:47:56.8              *    USID = 0000000001
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0032
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.B00004                                  TYPE=PART
  DBD=POHIDJD   DDN=POHIDJDB DSID=006 DBORG=HIDAM  DSORG=OSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     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
-------------------------------------------------------------------------------

 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.POHIDJDB.IC.IC112157           FILE SEQ=0001
  UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=222222



 REORG
  RUN     = 03.287 10:47:56.8              *    USID = 0000000001
```

Figure 74. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 8 of 9)

```
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0033
-------------------------------------------------------------------------------
 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
-------------------------------------------------------------------------------
03.289 12:42:49.3                   LISTING OF RECON                  PAGE 0034
-------------------------------------------------------------------------------
 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 74. Sample Listing of a RECON at the Active Site - DB (HALDB and PART) and Related Records (Part 9 of 9)

DB (FP) and Related Records

```
1999.251 11:46:24.1 -08:00          LISTING OF RECON                  PAGE 0361
-------------------------------------------------------------------------------
 DB
  DBD=DEDBDD01                                     DMB#=8       TYPE=FP
  SHARE LEVEL=1
  FLAGS:                             COUNTERS:
                                       RECOVERY NEEDED COUNT   =0
                                       IMAGE COPY NEEDED COUNT =0
    PROHIBIT AUTHORIZATION=OFF         AUTHORIZED AREAS        =1
    RECOVERABLE           =YES         EEQE COUNT              =0
-------------------------------------------------------------------------------
1999.251 11:46:24.1 -08:00          LISTING OF RECON                  PAGE 0362
-------------------------------------------------------------------------------
 DBDS
  DBD=DEDBDD01  AREA=DD01AR0                                    TYPE=FP
  SHARE LEVEL=1            DSID=001 DBORG=DEDB   DSORG=VSAM
  GSGNAME=IMSGSG1             USID=0000000003
  AUTHORIZED USID=0000000003  RECEIVE USID=0000000003 HARD USID=0000000003
  RECEIVE NEEDED USID=0000000000
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     DSSN=00000002
  NOREUSE         RECOVPD=0      NOVSO  PREOPEN    NOPRELOAD
  CFSTR1=**NULL**         CFSTR2=**NULL**          NOLKASID    NOMAS
  DEFLTJCL=**NULL**  ICJCL=ICJCL    RECVJCL=ICRCVJCL  RECOVJCL=RECOVJCL
  DBRCVGRP=**NULL**

FLAGS:                               COUNTERS:
    PROHIBIT AUTHORIZATION=OFF         AUTHORIZED SUBSYSTEMS   =1
                                       HELD AUTHORIZATION STATE=6
    IC NEEDED             =OFF         ADS AVAIL #             =1
    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:
                                                                  CREATE
    -ADS DDN--ADS DSN-                                       -STAT- -RUNNING-
    DD01AR0  DD01AR0                                          AVAIL     NO

ASSOCIATED SUBSYSTEM INFORMATION:
                               ENCODED
     -SSID-   -ACCESS INTENT-  -STATE-  -SS ROLE-
    SYS3         UPDATE           6      ACTIVE
-------------------------------------------------------------------------------

 ALLOC
  ALLOC   =1999.251 11:37:42.5 -08:00     *  ALLOC LRID =0000000000000000
  DSSN=0000000001 USID=0000000002 START = 1999.251 11:37:33.0 -08:00

 ALLOC
  ALLOC   =1999.251 11:40:57.1 -08:00     *  ALLOC LRID =0000000000000000
  DSSN=0000000002 USID=0000000003 START = 1999.251 11:40:50.0 -08:00
```

Figure 75. Sample Listing of a RECON at the Active Site - DB (FP) and Related Records (Part 1 of 2)

```
IMAGE
 RUN     = 1996.100 07:15:11.2 -08:00     *
 STOP    = 1996.100 07:16:12.3 -08:00        SMSCIC     USID=0000000001

IC1
 DSN=IMSVS.DEDBDD01.SMSCIC.DSN1                   FILE SEQ=0001
 UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                    VOLSER=IMSCC1
```

Figure 75. Sample Listing of a RECON at the Active Site - DB (FP) and Related Records (Part 2 of 2)

Sample Listing of a RECON at the Tracking Site

Beginning with Figure 76 on page 501, 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 522 describes the fields that can be present in a listing of the RECON.

In This Section:

- “RECON Status Record” on page 501
- “Log Records” on page 502
- “GSG Record” on page 507
- “SSYS Record” on page 508
- “BACKOUT Record” on page 508
- “CAGRP and CA Records” on page 509
- “DBDSGRP Records” on page 510
- “DB (IMS) and Related Records” on page 510
- “DB (HALDB and PART) and Related Records” on page 511
- “DB (FP) and Related Records” on page 522

RECON Status Record

```
03.287 13:23:51.7                  LISTING OF RECON                 PAGE 0002
--------------------------------------------------------------------------------
 RECON
  RECOVERY CONTROL DATA SET, IMS V9R1
  DMB#=6                             INIT TOKEN=03287F1803058F
  NOFORCER  LOG DSN CHECK=CHECK17    STARTNEW=NO
  TAPE UNIT=3480      DASD UNIT=SYSDA     TRACEOFF   SSID=**NULL**
  LIST DLOG=NO                 CA/IC/LOG DATA SETS CATALOGED=NO
  MINIMUM VERSION = 8.1
  LOG RETENTION PERIOD=00.001 00:00:00.0
  COMMAND AUTH=NONE  HLQ=**NULL**
  SIZALERT DSNUM=15      VOLNUM=16     PERCENT= 95
  LOGALERT DSNUM=3       VOLNUM=16

  TIME STAMP INFORMATION:

    TIMEZIN = %SYS

    OUTPUT FORMAT:  DEFAULT = LOCORG NONE   PUNC YY
                    CURRENT = LOCORG NONE   PUNC YY


  IMSPLEX = ** NONE **

 -DDNAME-      -STATUS-       -DATA SET NAME-
  RECON1        COPY1          IMSTESTS.DSHR.RECON1
  RECON2        COPY2          IMSTESTS.DSHR.RECON2
  RECON3        SPARE          IMSTESTS.DSHR.RECON3
```

Figure 76. Sample Listing of a RECON at the Tracking Site - RECON Status Record

Log Records

```
02.231 10:32:50.5                   LISTING OF RECON                  PAGE 0003
--------------------------------------------------------------------------------
 PRILOG                                         RECORD SIZE=     624
  START = 02.231 10:43:16.8              *    SSID=SYS3     Version=9.1
  STOP  = 02.231 10:51:11.8                   #DSN=3
  GSGNAME=IMSGSG1     TRACKING
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 1
  EARLIEST CHECKPOINT = 02.231 10:43:21.1

  DSN=IMSTESTL.RSR.RLDS1.N0000005
  START = 02.231 10:43:16.8                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:47:18.9                LAST  DS LSN= 0000000000000445
  #DS CHECKPOINTS=  1  CHKPT ID = 00.000 00:00:00.0

  DSN=IMSTESTL.RSR.RLDS1.N0000008
  START = 02.231 10:47:18.9                FIRST DS LSN= 0000000000000446
  STOP  = 02.231 10:51:11.6                LAST  DS LSN= 000000000000060E
  #DS CHECKPOINTS=  2  CHKPT ID = 00.000 00:00:00.0

  DSN=IMSTESTL.RSR.RLDS1.N0000010
  START = 02.231 10:51:11.6                FIRST DS LSN= 000000000000060F
  STOP  = 02.231 10:51:11.8                LAST  DS LSN= 000000000000066A
  #DS CHECKPOINTS=  0  CHKPT ID = 02.231 10:51:11.5

 LOGALL
  START   = 02.231 10:43:16.8              *
  EARLIEST ALLOC TIME = 02.231 10:45:06.1
  DBDS ALLOC=5                         -DBD-    -DDN-    -ALLOC-
                                       DEDBJN21 DB21AR1   1
                                       DEDBJN21 DB21AR2   1
                                       DEDBJN21 DB21AR3   1
                                       DEDBJN21 DB21AR4   1
                                       DEDBJN21 DB21AR5   1
02.231 10:32:50.5                   LISTING OF RECON                  PAGE 0004
--------------------------------------------------------------------------------
 PRISLD                                         RECORD SIZE=     624
  START = 02.231 10:43:16.8              *    SSID=SYS3     Version=9.1
  STOP  = 02.231 10:51:11.8                   #DSN=3
  GSGNAME=IMSGSG1     TRACKING
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 1

  DSN=IMSTESTL.RSR.ARCH1.N0000004
  START = 02.231 10:43:16.8                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:47:18.9                LAST  DS LSN= 0000000000000445
  #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 = 02.231 10:47:18.9                FIRST DS LSN= 0000000000000446
  STOP  = 02.231 10:51:11.6                LAST  DS LSN= 000000000000060E
  #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.N0000009
  START = 02.231 10:51:11.6                FIRST DS LSN= 000000000000060F
  STOP  = 02.231 10:51:11.8                LAST  DS LSN= 000000000000066A
  #DS CHECKPOINTS=  0  CHKPT ID = 02.231 10:51:11.5
  CHECKPOINT TYPES=00: SIMPLE=N SNAPQ=N DUMPQ=N PURGE=N FREEZE=N
```

Figure 77. Sample Listing of a RECON at the Tracking Site - Log Records (Part 1 of 6)

```
02.231 10:32:50.5                  LISTING OF RECON                  PAGE 0005
--------------------------------------------------------------------------------
 PRILOG                                      RECORD SIZE=     744
  START = 02.231 10:53:20.0             *    SSID=SYS3     Version=9.1
  STOP  = 02.231 11:11:21.8                  #DSN=4
  GSGNAME=IMSGSG1     TRACKING
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 2
  EARLIEST CHECKPOINT = 02.231 10:55:10.3

  DSN=**** COMPRESSED DATA SET ****
  START = 02.231 10:53:20.0                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:54:55.1                LAST  DS LSN= 00000000000004DE
  #DS CHECKPOINTS=  1

  DSN=IMSTESTL.RSR.RLDS1.N0000023
  START = 02.231 10:54:55.1                FIRST DS LSN= 00000000000004DF
  STOP  = 02.231 10:55:16.8                LAST  DS LSN= 00000000000005F6
  #DS CHECKPOINTS=  1  CHKPT ID = 02.231 10:55:10.3

  DSN=IMSTESTL.RSR.RLDS1.N0000026
  START = 02.231 10:55:16.8                FIRST DS LSN= 00000000000005F7
  STOP  = 02.231 11:11:21.4                LAST  DS LSN= 00000000000007B1
  #DS CHECKPOINTS=  2  CHKPT ID = 02.231 10:55:16.8

  DSN=IMSTESTL.RSR.RLDS1.N0000028
  START = 02.231 11:11:21.4                FIRST DS LSN= 00000000000007B2
  STOP  = 02.231 11:11:21.8                LAST  DS LSN= 000000000000080D
  #DS CHECKPOINTS=  0  CHKPT ID = 02.231 11:11:21.2

 LOGALL
  START   = 02.231 10:53:20.0              *
  EARLIEST ALLOC TIME = 00.000 00:00:00.0
  DBDS ALLOC=0
02.231 10:32:50.5                  LISTING OF RECON                  PAGE 0006
--------------------------------------------------------------------------------
 PRISLD                                      RECORD SIZE=     744
  START = 02.231 10:53:20.0             *    SSID=SYS3     Version=9.1
  STOP  = 02.231 11:11:21.8                  #DSN=4
  GSGNAME=IMSGSG1     TRACKING
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 2

  DSN=**** COMPRESSED DATA SET ****
  START = 02.231 10:53:20.0                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:54:55.1                LAST  DS LSN= 00000000000004DE
  #DS CHECKPOINTS=  1
  CHECKPOINT TYPES=80: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=N

  DSN=IMSTESTL.RSR.ARCH1.N0000022
  START = 02.231 10:54:55.1                FIRST DS LSN= 00000000000004DF
  STOP  = 02.231 10:55:16.8                LAST  DS LSN= 00000000000005F6
  #DS CHECKPOINTS=  1  CHKPT ID = 02.231 10:55:10.3
  CHECKPOINT TYPES=40: SIMPLE=N SNAPQ=Y DUMPQ=N PURGE=N FREEZE=N

  DSN=IMSTESTL.RSR.ARCH1.N0000025
  START = 02.231 10:55:16.8                FIRST DS LSN= 00000000000005F7
  STOP  = 02.231 11:11:21.4                LAST  DS LSN= 00000000000007B1
  #DS CHECKPOINTS=  2  CHKPT ID = 02.231 10:55:16.8
  CHECKPOINT TYPES=88: SIMPLE=Y SNAPQ=N DUMPQ=N PURGE=N FREEZE=Y

  DSN=IMSTESTL.RSR.ARCH1.N0000027
  START = 02.231 11:11:21.4                FIRST DS LSN= 00000000000007B2
  STOP  = 02.231 11:11:21.8                LAST  DS LSN= 000000000000080D
  #DS CHECKPOINTS=  0  CHKPT ID = 02.231 11:11:21.2
  CHECKPOINT TYPES=00: SIMPLE=N SNAPQ=N DUMPQ=N PURGE=N FREEZE=N
```

Figure 77. Sample Listing of a RECON at the Tracking Site - Log Records (Part 2 of 6)

```
02.231 10:32:50.5                   LISTING OF RECON                  PAGE 0007
--------------------------------------------------------------------------------
 PRILOG                                       RECORD SIZE=     304
  START = 02.231 11:11:58.9              *    SSID=BATCH1   Version=9.1
  STOP  = 02.231 11:12:00.4                   #DSN=1
  GSGNAME=IMSGSG1     TRACKING
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 3

  DSN=IMSTESTL.RSR.ARCH1.N0000030
  START = 02.231 11:11:58.9                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 11:12:00.4                LAST  DS LSN= 000000000000009C
  #DS CHECKPOINTS=  0  CHKPT ID = 00.000 00:00:00.0

 LOGALL
  START   = 02.231 11:11:58.9              *
  EARLIEST ALLOC TIME = 02.231 11:11:59.7
  DBDS ALLOC=3                        -DBD-    -DDN-    -ALLOC-
                                      DHVNTZ02 HIDAM     1
                                      DXVNTZ02 XDLBT04I  1
                                      DIVNTZ02 DBHVSAM1  1
02.231 10:32:50.5                   LISTING OF RECON                  PAGE 0008
--------------------------------------------------------------------------------
 PRILOG                                       RECORD SIZE=     464
  START = 02.231 11:14:06.8              *    SSID=SYS3     Version=9.1
  STOP  = 02.231 10:28:00.1                   #DSN=2
  GSGNAME=IMSGSG1     TRACKING
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 4
  EARLIEST CHECKPOINT = 02.231 11:14:09.5

  DSN=IMSTESTL.RSR.RLDS1.N0000033
  START = 02.231 11:14:06.8                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:25:58.3                LAST  DS LSN= 000000000000025F
  #DS CHECKPOINTS=  1  CHKPT ID = 00.000 00:00:00.0

  DSN=IMSTESTL.RSR.RLDS1.N0000037
  START = 02.231 10:25:58.3                FIRST DS LSN= 0000000000000260
  STOP  = 02.231 10:28:00.1                LAST  DS LSN= 000000000000027A
  #DS CHECKPOINTS=  0  CHKPT ID = 00.000 00:00:00.0

 LOGALL
  START   = 02.231 11:14:06.8              *
  EARLIEST ALLOC TIME = 02.231 11:14:10.0
  DBDS ALLOC=4                        -DBD-    -DDN-    -ALLOC-
                                      DEDBDD01 DD01AR0   1
                                      DHVNTZ02 HIDAM     1
                                      DXVNTZ02 XDLBT04I  1
                                      DIVNTZ02 DBHVSAM1  1
```

Figure 77. Sample Listing of a RECON at the Tracking Site - Log Records (Part 3 of 6)

```
02.231 10:32:50.5                   LISTING OF RECON                  PAGE 0009
-------------------------------------------------------------------------------
 PRISLD                                       RECORD SIZE=     464
  START = 02.231 11:14:06.8              *    SSID=SYS3     Version=9.1
  STOP  = 02.231 10:28:00.1                   #DSN=2
  GSGNAME=IMSGSG1     TRACKING
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 4

  DSN=IMSTESTL.RSR.ARCH1.N0000032
  START = 02.231 11:14:06.8                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:25:58.3                LAST  DS LSN= 000000000000025F
  #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.N0000036
  START = 02.231 10:25:58.3                FIRST DS LSN= 0000000000000260
  STOP  = 02.231 10:28:00.1                LAST  DS LSN= 000000000000027A
  #DS CHECKPOINTS=  0  CHKPT ID = 00.000 00:00:00.0
  CHECKPOINT TYPES=00: SIMPLE=N SNAPQ=N DUMPQ=N PURGE=N FREEZE=N
02.231 10:32:50.5                   LISTING OF RECON                  PAGE 0010
-------------------------------------------------------------------------------
 PRILOG                                       RECORD SIZE=     464
  START = 02.231 10:28:00.2              *    SSID=SYS3     Version=9.1
  STOP  = 00.000 00:00:00.0                   #DSN=2
  GSGNAME=IMSGSG1     TRACKING
  FIRST RECORD ID= 000000000000027B       PRILOG TOKEN= 5
  EARLIEST CHECKPOINT = 02.231 11:14:09.5

  DSN=IMSTESTL.RSR.RLDS1.N0000039
  START = 02.231 10:28:00.2                FIRST DS LSN= 000000000000027B
  STOP  = 02.231 10:28:03.8                LAST  DS LSN= 0000000000000334
  #DS CHECKPOINTS=  1  CHKPT ID = 02.231 11:14:09.5

  DSN=IMSTESTL.RSR.SLDS1.N0000035
  START = 00.000 00:00:00.0                FIRST DS LSN= 0000000000000480
  STOP  = 00.000 00:00:00.0                LAST  DS LSN= 0000000000000000
  #DS CHECKPOINTS=  0  CHKPT ID = 00.000 00:00:00.0

 LOGALL
  START   = 02.231 10:28:00.2              *
  EARLIEST ALLOC TIME = 00.000 00:00:00.0
  DBDS ALLOC=0
```

Figure 77. Sample Listing of a RECON at the Tracking Site - Log Records (Part 4 of 6)

```
02.231 10:32:50.5                   LISTING OF RECON                  PAGE 0011
--------------------------------------------------------------------------------
 PRISLD                                       RECORD SIZE=     464
  START = 02.231 10:28:00.2              *    SSID=SYS3     Version=9.1
  STOP  = 00.000 00:00:00.0                   #DSN=2
  GSGNAME=IMSGSG1     TRACKING
  FIRST RECORD ID= 000000000000027B       PRILOG TOKEN= 5

  DSN=IMSTESTL.RSR.ARCH1.N0000038
  START = 02.231 10:28:00.2                FIRST DS LSN= 000000000000027B
  STOP  = 02.231 10:28:03.8                LAST  DS LSN= 0000000000000334
  #DS CHECKPOINTS=  1  CHKPT ID = 02.231 11:14:09.5
  CHECKPOINT TYPES=40: SIMPLE=N SNAPQ=Y DUMPQ=N PURGE=N FREEZE=N

  DSN=IMSTESTL.RSR.SLDS1.N0000035
  START = 00.000 00:00:00.0                FIRST DS LSN= 0000000000000480
  STOP  = 00.000 00:00:00.0                LAST  DS LSN= 0000000000000000
  #DS CHECKPOINTS=  0  CHKPT ID = 00.000 00:00:00.0
  CHECKPOINT TYPES=00: SIMPLE=N SNAPQ=N DUMPQ=N PURGE=N FREEZE=N
02.231 10:32:50.5                   LISTING OF RECON                  PAGE 0012
--------------------------------------------------------------------------------
 PRILOG                                       RECORD SIZE=     464
  START = 02.231 10:31:36.3              *    SSID=IMS2     Version=9.1
  STOP  = 02.231 10:32:14.3                   #DSN=2
  GSGNAME=IMSGSG1     TRACKING       PREV-GAP
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 6
  EARLIEST CHECKPOINT = 00.000 00:00:00.0

  DSN=IMSTESTL.RSR.RLDS1.N0000044
  START = 02.231 10:31:36.3                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:32:14.2                LAST  DS LSN= 0000000000000199
  #DS CHECKPOINTS=  2  CHKPT ID = 00.000 00:00:00.0

  DSN=IMSTESTL.RSR.RLDS1.N0000046
  START = 02.231 10:32:14.2                FIRST DS LSN= 000000000000019A
  STOP  = 02.231 10:32:14.3                LAST  DS LSN= 00000000000001E0
  #DS CHECKPOINTS=  0  CHKPT ID = 00.000 00:00:00.0

 LOGALL
  START   = 02.231 10:31:36.3              *
  EARLIEST ALLOC TIME = 00.000 00:00:00.0
  DBDS ALLOC=0
02.231 10:32:50.5                   LISTING OF RECON                  PAGE 0013
--------------------------------------------------------------------------------
 PRISLD                                       RECORD SIZE=     464
  START = 02.231 10:31:36.3              *    SSID=IMS2     Version=9.1
  STOP  = 02.231 10:32:14.3                   #DSN=2
  GSGNAME=IMSGSG1     TRACKING
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 6

  DSN=IMSTESTL.RSR.ARCH1.N0000043
  START = 02.231 10:31:36.3                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:32:14.2                LAST  DS LSN= 0000000000000199
  #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.N0000045
  START = 02.231 10:32:14.2                FIRST DS LSN= 000000000000019A
  STOP  = 02.231 10:32:14.3                LAST  DS LSN= 00000000000001E0
  #DS CHECKPOINTS=  0  CHKPT ID = 00.000 00:00:00.0
  CHECKPOINT TYPES=00: SIMPLE=N SNAPQ=N DUMPQ=N PURGE=N FREEZE=N
```

Figure 77. Sample Listing of a RECON at the Tracking Site - Log Records (Part 5 of 6)

```
02.231 10:32:50.5                   LISTING OF RECON                  PAGE 0014
-------------------------------------------------------------------------------
 PRIOLD
  SSID=SYS3            # DD ENTRIES=2
  EARLIEST CHECKPOINT = 02.231 10:42:09.9

  DDNAME=DFSOLP00   DSN=IMSTESTL.IMS01.OLDSP0
  START = 02.231 10:42:08.7                FIRST DS LSN= 0000000000000001
  STOP  = 02.231 10:30:30.2                LAST  DS LSN= 000000000000055D
  LOCK SEQUENCE# = 000000000000
  STATUS=ARC COMPLT                                FEOV=YES   AVAIL
  PRILOG TIME=02.231 10:42:08.7                ARCHIVE JOB NAME=JT103030
  Version=9.1
  DDNAME=DFSOLP01   DSN=IMSTESTL.IMS01.OLDSP1
  START = 02.231 10:30:30.2                FIRST DS LSN= 000000000000055E
  STOP  = 00.000 00:00:00.0                LAST  DS LSN= 0000000000000000
  LOCK SEQUENCE# = 000000000000
  STATUS=ACTIVE                                    FEOV=NO    AVAIL
  PRILOG TIME=02.231 10:42:08.7
  Version=9.1
DSP0260I  NO INTERIM RLDS/SLDS RECORDS FOUND IN RECON
DSP0260I  NO INT-ONLINE LOG RECORDS FOUND IN RECON
```

Figure 77. Sample Listing of a RECON at the Tracking Site - Log Records (Part 6 of 6)

GSG Record

```
1999.251 11:46:45.8 -08:00          LISTING OF RECON
-------------------------------------------------------------------------------
 GSG
  GSGNAME=IMSGSG1      #SGS=2      -SGNAME-      -ROLE-
                                   STLSITE1     ACTIVE
                                   STLSITE2     TRACKING     LOCAL
  CURRENT PRILOG TOKEN      = 6           TAKEOVER TOKEN = 0
  MINIMUM PRILOG TOKEN      = 1           DSN SEQ NUMBER = 36
  START TIME OF CURRENT LOG = 1999.251 11:44:49.0 -08:00
  HIGHEST ACTIVE SITE TIME  = 1999.251 11:45:29.4 -08:00
  TRACKING SUBSYSTEM ID     = SYS3
  TAKEOVER IN PROGRESS
```

Figure 78. Sample Listing of a RECON at the Tracking Site - GSG Record

SSYS Record

```
02.022 15:55:11.5                   LISTING OF RECON                  PAGE 0017
-------------------------------------------------------------------------------
 SSYS
  SSID=SYS3      LOG START=02.022 15:06:38.3
  SSTYPE=TRACKER ABNORMAL TERM=OFF  RECOVERY STARTED=NO   BACKUP=NO
  TRACKED=NO     TRACKER TERM=OFF   SHARING COVERED DBS=NO
                                             GSGNAME=IMSGSG1

  AUTHORIZED DATA BASES/AREAS=3      Version=9.1   XRF CAPABLE=NO
                                                    ENCODED
    -DBD-        -AREA-   -LEVEL-  -ACCESS INTENT-  -STATE-
    DHVNTZ02     **NULL**    3        EXCLUSIVE        7
    DXVNTZ02     **NULL**    3        EXCLUSIVE        7
    DIVNTZ02     **NULL**    3        EXCLUSIVE        7

 SSYS
  SSID=SYS3      LOG START=02.022 16:05:27.4
  SSTYPE=ONLINE  ABNORMAL TERM=ON   RECOVERY STARTED=NO   BACKUP=NO
  TRACKED=YES    TRACKER TERM=OFF   SHARING COVERED DBS=NO
  IRLMID=**NULL**  IRLM STATUS=NORMAL        GSGNAME=IMSGSG1

  AUTHORIZED DATA BASES/AREAS=4      Version=9.1   XRF CAPABLE=NO
                                                    ENCODED
    -DBD-        -AREA-   -LEVEL-  -ACCESS INTENT-  -STATE-
    DEDBDD01     DD01AR0     1        UPDATE           6
    DHVNTZ02     **NULL**    3        UPDATE           6
    DXVNTZ02     **NULL**    3        UPDATE           6
    DIVNTZ02     **NULL**    3        UPDATE           6
```

Figure 79. Sample Listing of a RECON at the Tracking Site - SSYS Record

BACKOUT Record

```
1999.251 11:46:45.8 -08:00          LISTING OF RECON
-------------------------------------------------------------------------------
 BACKOUT
  SSID=SYS3      #UORS=1

    RECOVERY TOKEN=E2E8E2F3404040400000000300000002
    TIME=1999.251 11:38:22.5 -08:00       PSB=PLVAPZ12
                INFLT   BMP
      ASSOCIATED DATA BASES=3

                  BACKED   DYN BKOUT
      -DBD-        -OUT -   -FAILURE-
      DHVNTZ02      NO         NO
      DXVNTZ02      NO         NO
      DIVNTZ02      NO         NO
```

Figure 80. Sample Listing of a RECON at the Tracking Site - BACKOUT Record

CAGRP and CA Records

```
1999.274 09:27:48.3 -09:00          LISTING OF RECON
-------------------------------------------------------------------------------
 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=IMSVS.CAGRP1.CA.CA182601                     FILE SEQ=1
  CAGRP=CAGRP1       STOP    = 1999.274 08:40:44.3 -09:00     *
                     UNIT=SYSDA      VOLS DEF=1    VOLS USED=1
                                     VOLSER=222222
  RUN     = 1999.274 09:26:39.6 -09:00
   DBD=DEDBJN21 DDN=DB21AR1  PURGETIME = 1999.274 08:31:07.0 -09:00
     CHANGES ACCUMULATED=YES COMPLETE CA=YES INDOUBT EEQES=NO
     LSN    = 000000000000      DSSN   = 0000000001
     LRID   = 0000000000000414  USID   = 0000000002
   DBD=DEDBJN21 DDN=DB21AR3  PURGETIME = 1999.274 08:31:07.0 -09:00
     CHANGES ACCUMULATED=YES COMPLETE CA=YES INDOUBT EEQES=NO
     LSN    = 000000000000      DSSN   = 0000000001
     LRID   = 0000000000000428  USID   = 0000000002
   DBD=DEDBJN21 DDN=DB21AR6  PURGETIME = 1999.274 08:31:07.0 -09:00
     CHANGES ACCUMULATED=NO  COMPLETE CA=YES INDOUBT EEQES=NO
     LSN    = 000000000000      DSSN   = 0000000000
     LRID   = 0000000000000000  USID   = 0000000000
   DBD=DEDBJN21 DDN=DB21AR7  PURGETIME = 1999.274 08:31:07.0 -09:00
     CHANGES ACCUMULATED=NO  COMPLETE CA=YES INDOUBT EEQES=NO
     LSN    = 000000000000      DSSN   = 0000000000
     LRID   = 0000000000000000  USID   = 0000000000
-------------------------------------------------------------------------------
 CAGRP
  GRPNAME=CAGRP2   GRPMAX=5    CA AVAIL=0    CA USED=0
  NOREUSE CAJCL=CAJCL          DEFLTJCL=**NULL**
                               #MEMBERS=3    -DBD-     -DDN-
                                             DBVHDJ05  CJVHDG1E
                                             DBOHIDK5  CKOHIG1O
                                             DXVHIDK5  CKVHIIXK
```

Figure 81. Sample Listing of a RECON at the Tracking Site - CAGRP and CA Records

DBDSGRP Records

```
1999.274 09:27:48.3 -09:00          LISTING OF RECON
-------------------------------------------------------------------------------
 DBDSGRP
  GRPNAME=DBGRP1               #MEMBERS=6    -DBD-    -DDN/AREA-
                                             DIVNTZ02  **NULL**
                                             DHVNTZ02  **NULL**
                                             DXVNTZ02  **NULL**
                                             DB21AR0   **NULL**
                                             DB21AR1   **NULL**
                                             DB21AR2   **NULL**

 DBDSGRP
  GRPNAME=FJKGRP               #MEMBERS=5    -DBD-    -DDN/AREA-
                                             DIVNTZ02  DBHVSAM1
                                             DIVNTZ02  DBHVSAM2
                                             DHVNTZ02  HIDAM
                                             DHVNTZ02  HIDAM2
                                             DXVNTZ02  XDLBT04I
```

Figure 82. Sample Listing of a RECON at the Tracking Site - DBDSGRP Records

DB (IMS) and Related Records

```
1999.251 11:46:45.8 -08:00          LISTING OF RECON
-------------------------------------------------------------------------------
 DB
  DBD=DBVHDJ05                 IRLMID=*NULL        DMB#=1       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


 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=0     DSSN=00000000
  NOREUSE         RECOVPD=0
  DEFLTJCL=**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
```

Figure 83. Sample Listing of a RECON at the Tracking Site - DB (IMS) and Related Records

DB (HALDB and PART) and Related Records

```
03.287 13:23:51.7                   LISTING OF RECON                  PAGE 0013
-------------------------------------------------------------------------------
 DB
  DBD=DBHIDJ03                        DMB#=6      CHANGE#=5     TYPE=HALDB
  SHARE LEVEL=3               GSGNAME=IMSGSG1
  PSNAME=**NULL**  DBORG=PHIDAM   DSORG=OSAM    CURRENT PARTITION ID=00004
  FLAGS:                             COUNTERS:
    RECOVERABLE           =YES         PARTITIONS              =4
    ONLINE REORG CAPABLE  =YES         DATA SET GROUP MEMBERS  =2
-------------------------------------------------------------------------------
03.287 13:23:51.7                   LISTING OF RECON                  PAGE 0014
-------------------------------------------------------------------------------
  DBD=POHIDJA   MASTER DB=DBHIDJ03  IRLMID=*NULL  CHANGE#=2     TYPE=PART
  USID=0000000003  AUTHORIZED USID=0000000003  HARD USID=0000000003
  RECEIVE USID=0000000003  RECEIVE NEEDED USID=0000000000
  DBRCVGRP=**NULL**
  DSN PREFIX=IMSTESTS.DBHIDJ03                          PARTITION ID=00001
  PREVIOUS PARTITION=**NULL**    NEXT PARTITION=POHIDJB
  OLRIMSID=**NULL**    ACTIVE DBDS=A-J

    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

  FLAGS:                             COUNTERS:
    BACKOUT NEEDED        =OFF         RECOVERY NEEDED COUNT   =0
    READ ONLY             =OFF         IMAGE COPY NEEDED COUNT =0
    PROHIBIT AUTHORIZATION=OFF         AUTHORIZED SUBSYSTEMS   =1
                                       HELD AUTHORIZATION STATE=0
    DATABASE LEVEL TRACK  =YES         EEQE COUNT              =0
    TRACKING SUSPENDED    =NO          RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =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

  ASSOCIATED SUBSYSTEM INFORMATION:
                               ENCODED  B/O NEEDED
     -SSID-   -ACCESS INTENT-  -STATE-   -COUNT-  -SS ROLE-
    IMS2         EXCLUSIVE       7         0      TRACKING
```

Figure 84. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 1 of 11)

```
--------------------------------------------------------------------------------
03.287 13:23:51.7                   LISTING OF RECON                  PAGE 0015
--------------------------------------------------------------------------------
 DBDS
  DSN=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=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 10:49:51.9              *  ALLOC LRID =000000000000000C
  DSSN=0000000001 USID=0000000002 START = 03.287 10:49:50.0

 ALLOC
  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.POHIDJA.POHIDJAA.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 84. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 2 of 11)

```
03.287 13:23:51.7                   LISTING OF RECON                  PAGE 0016
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.B00001                                  TYPE=PART
DDN=POHIDJAB DSID=006 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 10:49:51.8              *  ALLOC LRID =0000000000000008
  DSSN=0000000001 USID=0000000002 START = 03.287 10:49:50.0

 ALLOC
  ALLOC   =03.287 10:55:40.1              *  ALLOC LRID =000000000000000D
  DSSN=0000000002 USID=0000000003 START = 03.287 10:55:36.8

 IMAGE
  RUN     = 03.287 11:23:05.7              *  RECORD COUNT =0
  STOP    = 00.000 00:00:00.0                 BATCH      USID=0000000003

 IC1
  DSN=IMSVS.POHIDJA.POHIDJAB.IC.IC112154           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 11:54:13.3              *  RUN USID      = 0000000000
03.287 13:23:51.7                   LISTING OF RECON                  PAGE 0017
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.L00001                                  TYPE=PART
  DBD=POHIDJA   DDN=POHIDJAL 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 0018
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.X00001                                  TYPE=PART
  DBD=POHIDJA   DDN=POHIDJAX DSID=005 DBORG=INDEX  DSORG=VSAM
                                  OTHER DDN=**NULL**
  FLAGS:                             COUNTERS:
    RECOV NEEDED   =OFF                EEQE COUNT              =0
```

Figure 84. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 3 of 11)

```
-------------------------------------------------------------------------------
03.287 13:23:51.7                   LISTING OF RECON                  PAGE 0019
-------------------------------------------------------------------------------
 DB
  DBD=POHIDJB   MASTER DB=DBHIDJ03  IRLMID=*NULL  CHANGE#=3     TYPE=PART
  USID=0000000003  AUTHORIZED USID=0000000003  HARD USID=0000000003
  RECEIVE USID=0000000003  RECEIVE NEEDED USID=0000000000
  DBRCVGRP=**NULL**
  DSN PREFIX=IMSTESTS.DBHIDJ03                          PARTITION ID=00002
  PREVIOUS PARTITION=POHIDJA     NEXT PARTITION=POHIDJC
  OLRIMSID=**NULL**    ACTIVE DBDS=A-J

  FREE SPACE:
    FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0

  PARTITION HIGH KEY/STRING (CHAR):               (LENGTH=5  )
    J0400
  PARTITION HIGH KEY/STRING (HEX):
    D1F0F4F0F040404040404040404040404040404040404040404040404040404040

  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   =1
                                       HELD AUTHORIZATION STATE=0
    DATABASE LEVEL TRACK  =YES         EEQE COUNT              =0
    TRACKING SUSPENDED    =NO          RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =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

  ASSOCIATED SUBSYSTEM INFORMATION:
                               ENCODED  B/O NEEDED
     -SSID-   -ACCESS INTENT-  -STATE-   -COUNT-  -SS ROLE-
    IMS2         EXCLUSIVE       7         0      TRACKING
```

Figure 84. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 4 of 11)

```
-------------------------------------------------------------------------------
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.POHIDJB.POHIDJBA.IC.IC112155            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 84. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 5 of 11)

```
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     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 10:49:55.4              *  ALLOC LRID =0000000000000C2A
  DSSN=0000000001 USID=0000000002 START = 03.287 10:49:50.0
 ALLOC
  ALLOC   =03.287 10:55:44.4              *  ALLOC LRID =000000000000092E
  DSSN=0000000002 USID=0000000003 START = 03.287 10:55:36.8

 IMAGE
  RUN     = 03.287 11:24:19.6              *  RECORD COUNT =0
  STOP    = 00.000 00:00:00.0                 BATCH      USID=0000000003

 IC1
  DSN=IMSVS.POHIDJB.POHIDJBB.IC.IC112155           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 12:10:36.0              *  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
--------------------------------------------------------------------------------
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
```

Figure 84. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 6 of 11)

```
-------------------------------------------------------------------------------
03.287 13:23:51.7                   LISTING OF RECON                  PAGE 0024
-------------------------------------------------------------------------------
 DB
  DBD=POHIDJC   MASTER DB=DBHIDJ03  IRLMID=*NULL  CHANGE#=4     TYPE=PART
  USID=0000000001  AUTHORIZED USID=0000000001  HARD USID=0000000001
  RECEIVE USID=0000000001  RECEIVE NEEDED USID=0000000000
  DBRCVGRP=**NULL**
  DSN PREFIX=IMSTESTS.DBHIDJ03                          PARTITION ID=00003
  PREVIOUS PARTITION=POHIDJB     NEXT PARTITION=POHIDJD
  OLRIMSID=**NULL**    ACTIVE DBDS=A-J

  FREE SPACE:
    FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0
  PARTITION HIGH KEY/STRING (CHAR):               (LENGTH=5  )
    J0600
  PARTITION HIGH KEY/STRING (HEX):
    D1F0F6F0F0404040404040404040404040404040404040404040404040404040

  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
                                       HELD AUTHORIZATION STATE=0
    DATABASE LEVEL TRACK  =YES         EEQE COUNT              =0
    TRACKING SUSPENDED    =NO          RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =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
```

Figure 84. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 7 of 11)

```
-------------------------------------------------------------------------------
03.287 13:23:51.7                   LISTING OF RECON                  PAGE 0025
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.A00003                                  TYPE=PART
  DBD=POHIDJC   DDN=POHIDJCA DSID=001 DBORG=HIDAM  DSORG=OSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     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
-------------------------------------------------------------------------------

 IMAGE
  RUN     = 03.287 11:24:57.2              *  RECORD COUNT =0
  STOP    = 00.000 00:00:00.0                 BATCH      USID=0000000001

 IC1
  DSN=IMSVS.POHIDJC.POHIDJCA.IC.IC112156           FILE SEQ=0001
  UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=222222



 RECOV
  RUN     = 03.287 12:16:02.1              *  RUN USID      = 0000000000
03.287 13:23:51.7                   LISTING OF RECON                  PAGE 0026
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.B00003                                  TYPE=PART
  DBD=POHIDJC   DDN=POHIDJCB DSID=006 DBORG=HIDAM  DSORG=OSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     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
-------------------------------------------------------------------------------

 IMAGE
  RUN     = 03.287 11:25:37.6              *  RECORD COUNT =0
  STOP    = 00.000 00:00:00.0                 BATCH      USID=0000000001

 IC1
  DSN=IMSVS.POHIDJC.POHIDJCB.IC.IC112156           FILE SEQ=0001
  UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=222222



 RECOV
  RUN     = 03.287 12:20:52.3              *  RUN USID      = 0000000000
```

Figure 84. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 8 of 11)

```
03.287 13:23:51.7                   LISTING OF RECON                 PAGE 0027
--------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.L00003                                   TYPE=PART
  DBD=POHIDJC   DDN=POHIDJCL 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 0028
--------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.X00003                                   TYPE=PART
  DBD=POHIDJC   DDN=POHIDJCX DSID=005 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 0029
--------------------------------------------------------------------------------
 DB
  DBD=POHIDJD   MASTER DB=DBHIDJ03  IRLMID=*NULL  CHANGE#=5      TYPE=PART
  USID=0000000001  AUTHORIZED USID=0000000001  HARD USID=0000000001
  RECEIVE USID=0000000001  RECEIVE NEEDED USID=0000000000
  DBRCVGRP=**NULL**
  DSN PREFIX=IMSTESTS.DBHIDJ03                           PARTITION ID=00004
  PREVIOUS PARTITION=POHIDJC     NEXT PARTITION=**NULL**
  OLRIMSID=**NULL**    ACTIVE DBDS=A-J

  FREE SPACE:
    FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0

  PARTITION HIGH KEY/STRING (CHAR):               (LENGTH=5  )
    .....
  PARTITION HIGH KEY/STRING (HEX):
    FFFFFFFFFF404040404040404040404040404040404040404040404040404040

  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
                                       HELD AUTHORIZATION STATE=0
    DATABASE LEVEL TRACK  =YES         EEQE COUNT              =0
    TRACKING SUSPENDED    =NO          RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =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
```

Figure 84. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 9 of 11)

```
--------------------------------------------------------------------------------
03.287 13:23:51.7                    LISTING OF RECON                  PAGE 0030
--------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBHIDJ03.A00004                                   TYPE=PART
  DBD=POHIDJD   DDN=POHIDJDA DSID=001 DBORG=HIDAM  DSORG=OSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0      IC USED=1     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
--------------------------------------------------------------------------------

 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.POHIDJDA.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=POHIDJDB DSID=006 DBORG=HIDAM  DSORG=OSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0      IC USED=1     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
--------------------------------------------------------------------------------

 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.POHIDJDB.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 84. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 10 of 11)

```
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 84. Sample Listing of a RECON at the Tracking Site - DB (HALDB and PART) and Related Records (Part 11 of 11)

DB (FP) and Related Records

```
1999.251 11:46:45.8 -08:00          LISTING OF RECON
-------------------------------------------------------------------------------
 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        =1
    RECOVERABLE           =YES         EEQE COUNT              =0


 DBDS
  DBD=DEDBDD01  AREA=DD01AR0                                    TYPE=FP
  SHARE LEVEL=1             DSID=001 DBORG=DEDB   DSORG=VSAM
  GSGNAME=IMSGSG1              USID=0000000002
  AUTHORIZED USID=0000000000  RECEIVE USID=0000000000 HARD USID=0000000000
  RECEIVE NEEDED USID=0000000000
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=0     DSSN=00000001
  NOREUSE         RECOVPD=0      NOVSO  PREOPEN    NOPRELOAD
  CFSTR1=**NULL**         CFSTR2=**NULL**          NOLKASID
  DEFLTJCL=**NULL**  ICJCL=ICJCL    RECVJCL=ICRCVJCL  RECOVJCL=RECOVJCL
  DBRCVGRP=**NULL**
  FLAGS:                             COUNTERS:
    PROHIBIT AUTHORIZATION=OFF         AUTHORIZED SUBSYSTEMS   =1
                                       HELD AUTHORIZATION STATE=6
    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:
                                                                  CREATE
    -ADS DDN--ADS DSN-                                    -STAT- -RUNNING-
    DD01AR0  DD01AR0                                      AVAIL     NO

  ASSOCIATED SUBSYSTEM INFORMATION:
                               ENCODED
     -SSID-   -ACCESS INTENT-  -STATE-  -SS ROLE-
    SYS3         UPDATE           6      ACTIVE
-------------------------------------------------------------------------------

 ALLOC
  ALLOC   =1999.251 11:37:42.5 -08:00     *  ALLOC LRID =00000000000000D8
  DSSN=0000000001 USID=0000000002 START = 1999.251 11:37:33.0 -08:00
```

Figure 85. 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 523
- “Fields Present in a Log Record” on page 526
- “Fields Present in a LOGALL Record” on page 529
- “Fields Present in an Online Log Record” on page 529

- “Fields Present in a GSG Record” on page 531
- “Fields Present in a SSYS Record” on page 531
- “Fields Present in a BACKOUT Record” on page 532
- “Fields Present in a CAGRP Record” on page 533
- “Fields Present in a CA Record” on page 534
- “Fields Present in a Data Group Record” on page 536
- “Fields Present in a DB (IMS) Record” on page 536
- | “Fields Present in a DB (HALDB) Record” on page 538
- “Fields Present in a DB (PART) Record” on page 539
- “Fields Present in a DB (Fast Path) Record” on page 543
- “Fields Present in a DBDS (non-Fast Path) Record” on page 543
- “Fields Present in a DBDS (Fast Path) Record” on page 545
- “Fields Present in an ALLOC Record” on page 548
- “Fields Present in an IMAGE Record” on page 549
- “Fields Present in a REORG Record” on page 551
- “Fields Present in a RECOV Record” on page 552

Examples of many of these fields appear in Figure 66 on page 474 and Figure 76 on page 501

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 36.

Table 36. Fields Present in the RECON Record

Record Type	Line Number	Field	Contents
RECON	1		
	2	RECOVERY CONTROL DATA SET, IMS VxRx	Identifies the version and release of the RECON.
	3	LAST USED DMB#= *nnn*	*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.
		INIT TOKEN= *token*	The token assigned to this RECON when it was initialized.
	4	FORCER | NOFORCER	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.
		LOG DSN CHECK= *xxxxxx*	The type of log data set name checking; *xxxxxx* is CHECK17, CHECK44, or NOCHECK.

Table 36. Fields Present in the RECON Record (continued)

Record Type	Line Number	Field	Contents
		STARTNEW= YES | NO	When I/O errors exist on one of the RECON, YES indicates that new jobs are to start. NO indicates that no new jobs are to start.
	5	TAPE UNIT= *unittype*	The default unit type for log data sets, NOREUSE image copy data sets, and NOREUSE change accumulation data sets that reside on tape devices.
		DASD UNIT= *unittype*	The default unit type for log data sets, NOREUSE image copy data sets, and NOREUSE change accumulation data sets that reside on DASD devices.
		TRACEON | TRACEOFF	TRACEON indicates DBRC is to produce external GTF trace entries.
		SSID= *xxxxxxxx*	The IMS subsystem default name.
	6	LISTDLOG= YES | NO	Indicates whether the names of log data sets deleted from RECON are to be listed.
		CA | IC | LOG DATA SETS CATALOGED= YES | NO	Indicates whether CA, IC, and log data sets are to be treated as if they are cataloged. To bypass volume serial checking of utility DD statements, specify YES and the data set is cataloged. If you specify NO (or the data set is not cataloged), volume serial checking takes place.
	7	MINIMUM VERSION = *x.x*	Indicates the minimum version of IMS that is allowed to sign on using these RECONs.
	8	LOG RETENTION PERIOD= *yy.ddd hh:mm:ss.t*	Indicates the minimum amount of time that DBRC is to keep log records in RECON.
	9	COMMAND AUTH=SAF | EXIT | BOTH | NONE	Indication of the level of DBRC command authorization being performed.
		HLQ=*hlq name*	High level qualifier of the resource name used to determine command authorization. **NULL** is printed when command authorization is not enabled.
	10	TRACE OPTIONS= *options*	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.
		SIZALERT= *xxxxxxxx xxxxxxxx xx*	The (optional) thresholds that trigger messages to warn you that a record has grown unnaturally large. Field values can be dsnum, volnum and percent.

Table 36. Fields Present in the RECON Record (continued)

Record Type	Line Number	Field	Contents
	11	LOGALERT= *xxxxxxxx xxxxxxxx*	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.
The following line is printed only if a failure occurred during a multiple update to the RECON:			
	12	UPDATE TYPE= *nnnn*	The type of multiple update that was in progress.
		DBID= *xxxxxxxx xxxxxxxx*	Data set name and data set ddname of the DBDS involved in the multiple update. Can be blank.
		CAGRP= *xxxxxxxx*	Name of the CA group involved in the multiple update. Contains blanks if no group was involved.
		NEW DDN= *xxxxxxxx*	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.
	13	OLD RECORD KEY	This field is printed only if the multiple update involves a key change.
	14	KEY TYPE= **xxxxx*	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.

		DBD= *xxxxxxxx*,
		DDN= *xxxxxxxx*
			The database name and data set ddname of a DBDS. If the record is PRILOG, SECLOG, IPRI, or ISEC, the fields are printed as `**NULL**`. If the record is a CAGRP or DB record, the DBD field is `**NULL**`, and the DDN field contains the name of the CA group or database.

		TIME= *time_stamp*	The time stamp of the key of the record.
	15-16	NEW RECORD KEY	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.
	17-18	BASE RECORD KEY	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.
	19-20	SSID= *ssidname* IRLMID= *irlmidname* SHARE LVL #= *n* #DB= *n* B/O#= *n*	Information about the multiple update.
	21	FLAGS:	A heading for the following lines.
	22	RECOV= ON | OFF NORECOV= ON | OFF ICON= ON | OFF ICOFF= ON | OFF	Information about the multiple update.

Table 36. Fields Present in the RECON Record (continued)

Record Type	Line Number	Field	Contents
	23	NORMAL= ON | OFF ABNORMAL= ON | OFF STARTRCV= ON | OFF ENDRECOV= ON | OFF	Information about the multiple update.
	24	READON= ON | OFF READOFF= ON | OFF AUTH= ON | OFF NOAUTH= ON | OFF	Information about the multiple update.
	25	B/O DONE= ON | OFF PASS 1= ON | OFF BACKOUT= ON | OFF SHARELVL= ON | OFF	Information about the multiple update.
	26	TIME STAMP INFORMATION:	Heading for section containing RECON time stamp information.
	27	TIMEZIN= *offset*	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.
		-LABEL- -OFFSET-	These column headings are printed if a Time Zone Label Table has been defined.
	28-nnn	*label offset*	The list of defined time zone offset labels with their corresponding offset values.
	29	OUTPUT FORMAT:	Time-stamp output format settings.
		DEFAULT= *offset display form yearsize*	Default time-stamp output format settings.
	30	CURRENT= *offset display form yearsize*	Current time-stamp output format settings.
	31	IMSPLEX=*imsplex_name*	The IMSplex name associated with the RECON.
	32	-DDNAME- -STATUS- -DATA SET NAME-	Column headings for the following one to three lines.
	33	RECON1 | RECON2 | RECON3 COPY1 *dsname*	The ddname, status, and data set name of the Copy1 RECON.
	34-35	RECON1 | RECON2 | RECON3 | COPY2 | DISCARDED | SPARE | UNAVAILABLE *dsname* or blank	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).

Fields Present in a Log Record

A log record's fields, including types, and their corresponding line numbers are described in Table 37 on page 527.

Table 37. Fields Present in a Log Record

Record Type	Line Number	Field	Contents
PRILOG or SECLOG or PRISLD or SECSLD or IPRI or ISEC or IPRISL or ISECSL or PRITSLDS or SECTSLDS or IPRITSLD or ISECTSLD	1		
		RECORD SIZE=*nnnnnn*	The logical record size for PRILOG, SECLOG, PRISLD, SECSLD, PRITSLDS, and SECTSLDS records. This field is not present for other log record types.
	2	START= *time_stamp**	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.
		SSID= *ssidname*	The name of the IMS subsystem.
		VERSION=*version*	The version of the IMS subsystem that created the log.
	3	STOP= *time_stamp*	Time stamp of the stop time (that is, close time) of the log data set. Zeros indicate that the data set is still open.
		#DSN= *nn*	The number of data sets in the log data set. A value of zero indicates that no data set has been created.
	4	GSGNAME= *gsgname*	Identifies the name of the GSG to which the subsystem producing this log belongs.
The following 4 fields are printed if the condition in the contents column is true:			
		TRACKING	This log data set was originally created by an active IMS subsystem of the nonlocal service group and transported to the tracking site.
		GAP	There is a gap in the log data sets of this record.
		PREV-GAP	There is a gap in a previous log record of the same global service group.
		BBO	Identifies the log record as it was created by batch backout. If the record was not created by batch backout, this field is blank.
	5	FIRST RECORD ID= *lsn*	The log record sequence number of the first log record that was written during initialization of the IMS subsystem.
		PRILOG TOKEN= *n*	The numeric log token assigned sequentially to PRILOG records for the same GSG.
	6	EARLIEST CHECKPOINT= *time_stamp*	This line is printed only for online PRILOGs. It indicates the earliest checkpoint required by IMS emergency restart (/ERE or /ERE BUILDQ).
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.			

Table 37. Fields Present in a Log Record (continued)

Record Type	Line Number	Field	Contents
	7	DSN= *log.dsname*	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.
		ERR	This field is printed if a previous command was used to indicate that an error exists in the log data set.
		UNIT= *unittype*	Unit type to be used for substitution during the `GENJCL` process.
	8	START= *time_stamp*	Start time of the data set entry in the log data set.
		FIRST RECORD LSN= *lsn*	The log record sequence number of the first log record of the data set.
	9	STOP= *time_stamp*	Stop time of the data set entry in the log data set. Zeros indicate that the data set is still open.
		LAST RECORD LSN= *lsn*	The log record sequence number of the last log record of the data set.
	10	FILE SEQ= *nnnn*	The file sequence number of the log data set. This field is not printed for tracking logs.
		#VOLUMES= *nnn*	The number of volumes for each data set entry. This field is not printed for tracking logs.
		#DS CHECKPOINTS= *n* CHKPT ID= *time_stamp*	Count of checkpoint records in the log data set, and the checkpoint ID value. This field is printed only for tracking logs.

		CHECKPOINT TYPES=*xx*
		SIMPLE= Y | N
		SNAPQ = Y | N
		DUMPQ = Y | N
		PURGE = Y | N
		FREEZE = Y | N
			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:
			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
			For more information, see the note that follows this table.

	11 (ACTIVE)	VOLSER= *volser* STOPTIME= *time_stamp*	Volume serial number and stop time. This field is not printed for tracking logs.
	12 (ACTIVE)	CKPTCT= *n* CHKPT ID= *time_stamp*	Checkpoint count and ID value.
	13 (ACTIVE)	LOCK SEQUENCE # =	A unique number, issued sequentially, given to a database resource to identify that it is in use by (i.e. ″locked″ by) a requestor. A locked resource cannot be used until the current requestor has finished using it.

Table 37. Fields Present in a Log Record (continued)

Record Type	Line Number	Field	Contents
		FEOV	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.

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 38.

Table 38. Fields Present in the LOGALL Record

Record Type	Line Number	Field	Contents
LOGALL	1		
	2	START= *time_stamp**	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.
	3	EARLIEST ALLOC TIME=*time_stamp*	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.
	4	DBDS ALLOC= *nnnn*	The number of allocated DBDSs for this log data set.
		-DBD- -DDN- -ALLOC-	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.
	5-*nnn*	*dbdname ddname nnnn*	The database name, data set ddname, and number of allocations of the DBDS during the time span of the allocated log.

Fields Present in an Online Log Record

An online log record's fields and their corresponding line numbers are described in Table 39 on page 530.

Table 39. Fields Present in an Online Log Record

Record Type	Line Number	Field	Contents
PRIOLD or SECOLD or IPRIOL or ISECOL	1		
	2	SSID= *xxxxxxxx*	The name of the IMS subsystem.
		# DD ENTRIES= *n*	The number of ddnames in the OLDS.
	3	EARLIEST CHECKPOINT= *time_stamp*	This line is printed only for PRIOLD records. It indicates the earliest checkpoint required by IMS emergency restart (/ERE or /ERE BUILDQ).
	4	DDNAME= *ddname*	The ddname of the OLDS.
		DSN=log-dsname	The data set name of the OLDS.
	5	START= *time_stamp*	Time stamp of the start time (that is, original open time) of the log data set.
		FIRST DS LSN= *lsn*	The log record sequence number of the first log record of the data set.
	6	STOP= *time_stamp*	Time stamp of the stop time (that is, close time) of the log data set. Zeros indicate that the data set is not closed.
		LAST DS LSN= *lsn*	The log record sequence number of the last log record of the data set.
	7	STATUS= ARC NEEDED| ARC STARTED | ARC SCHED | ARC COMPLT | ACTIVE	The status of the DD entry in the OLDS record. This field applies to the PRIOLDS record only.
		ERROR= CLOSE | PREV	Indicates a close error on an OLDS or on a previous OLDS record. This field applies to the PRIOLDS record only.
		FEOV= YES | NO	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.
		AVAIL | UNAVAIL	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 SECOLDS records only.
	8	PRILOG TIME= *time_stamp*	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.
		ARCHIVE JOB NAME= *jobname*	Name of the archive job generated by the GENJCL ARCHIVE function.
	blank line		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.

Fields Present in a GSG Record

A global service group record's fields and their corresponding line numbers are described in Table 40.

Table 40. Fields Present in a GSG Record

Record Type	Line Number	Field	Contents
GSG	1		
	2	GSGNAME= *gsgname*	Identifies the name of the GSG.
		#SGS=	The number of service groups in this GSG.
		-SGNAME- -ROLE-	Headings for the following lines 3 through 10.
	3-4	*sgnamesgrole*	Service group name and service group role (active or tracking).
		LOCAL	Identifies which service group is the local one of the GSG.
	5	CURRENT PRILOG TOKEN= *token*	The highest PRILOG token at the local site of this GSG.
		TAKEOVER TOKEN= *token*	The current PRILOG token at the time an RSR takeover was initiated at the local site.
	6	MINIMUM PRILOG TOKEN= *token*	The minimum PRILOG that is maintained at the local site of the GSG.
		DSN SEQ NUMBER= *number*	The initial DSN sequence number value for the GSG group.
	7	START TIME OF CURRENT LOG= *time_stamp*	The start time of the PRILOG with the current log token.
	8	HIGHEST ACTIVE SITE TIME= *time_stamp*	The highest time received from the active site. This field is only meaningful to the tracking site.
	9	TRACKING SUBSYSTEM ID= *ssid*	Subsystem identification of the tracking subsystem. This field is null at the active site.
	10	TAKEOVER IN PROGRESS	This is printed when an RSR takeover has been initiated at the local site.

Fields Present in a SSYS Record

A subsystem record's fields and their corresponding line numbers are described in Table 41.

Table 41. Fields Present in a SSYS Record

Record Type	Line Number	Field	Contents
SSYS	1		
	2	SSID= *ssidname*	The name of the IMS subsystem.
		LOG START= *time_stamp*	The earliest log data set start time associated with this system or the log data set start time created by batch backout.
	3	SSTYPE= ONLINE | BATCH | TRACKER	Indication of whether this is an IMS online, batch, or tracking subsystem.
		ABNORMAL TERM= ON | OFF	The value of this flag field is normally OFF. If it is ON, the system has been abnormally ended and database recovery is required.

Table 41. Fields Present in a SSYS Record (continued)

Record Type	Line Number	Field	Contents
		RECOVERY STARTED= YES | NO	If the value of this field is YES, this subsystem has signed on for recovery-started processing. Normally, this value is NO.
		BACKUP= YES | NO	If the value of this field is YES, there is an alternate subsystem.
	4	TRACKED= YES | NO	Indicates whether this is the record of an active subsystem being tracked by the tracking site.
		TRACKER TERM= ON | OFF	Indicates whether the tracking subsystem has terminated. This field only applies to the tracking subsystem record.
		SHARING COVERED DBS= YES | NO	Indicates that this active subsystem is sharing RSR-covered databases.
	5	IRLMID= *xxxxxx*	The IRLM with which this subsystem is communicating.
		IRLM STATUS= NORMAL | IRLM FAILURE | COMM FAILURE | SYS FAILURE	Indicates the status of the IRLM. NORMAL indicates no failure. This field does not apply to an RSR tracking subsystem.
		GSG NAME= *gsgname*	Identifies the name of the global service group to which the subsystem belongs.
	6	BACKUP IRLMID= *irlmname*	Identifies the IRLM with which the alternate subsystem is communicating. Listed only if an alternate subsystem exists.
		BACKUP TOKEN= *nnnn*	Identifies the backup token. Listed only if an alternate subsystem exists.
	7	AUTHORIZED DATABASES | AREAS=*nnn*	Indicates that this subsystem is currently authorized to *n* databases or areas.
	8	VERSION= *n*	Identifies the IMS release level through which the subsystem signed on.
		XRF CAPABLE=NO|YES	Identifies if the subsystem is XRF capable.
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:			
	9-10	-DBD- -AREA- -LEVEL- -ACCESSINTENT- ENCODED -STATE-	Column headings for the following line or lines that describe the currently authorized databases or areas.
	11-*nnn*	*dbdname 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 42 on page 533.

Table 42. Fields Present in a BACKOUT Record

Record Type	Line Number	Field	Contents
BACKOUT	1		
	2	SSID= *xxxxxxxx*	The name of the associated IMS subsystem.
		#UORS= *nn*	The number of units of recovery (UORs) in the BACKOUT record.
	3	RECOVERY TOKEN= *32 hexadecimal digits*	Describes a specific UOR.
	4	TIME= *time_stamp*	The time stamp of the beginning of the UOR (found in the X'5607' log record).
		PSB= *psbname*	Name of the PSB associated with the UOR.
	5	CANDIDATE INFLT | INDT BMP COLDEND CMD BATCH	UOR indicators, one or more of which might be listed. CANDIDATE: UOR identified by BBO utility prior to restart (COLDSTART | ACTIVE control statement). Reset (to null) when promoted to backout-needed status by IMS restart. INFLT, In-flight UOR due to an IMS failure. INDT, In-doubt UOR due to a CCTL or DBCTL failure. BMP, UOR due to a BMP. COLDEND, Cold start has ended. UOR can only be resolved by BBO. CMD, UOR entry has been modified by commands (CHANGE, DELETE, NOTIFY). BATCH, UOR due to a dynamic backout failure of DL/I batch.
	6	ASSOCIATED DATABASES= *nn*	Number of databases associated with the UOR.
	7, 8	BACKED DYN BKOUT -DBD- -OUT- -FAILURE	Heading for the following list of associated databases.
	9-*nnn*	*dbname* YES | NO YES | NO	Database name. The first YES or NO indicates whether the UOR has been backed out for the database. The second YES or NO indicates whether the UOR is the result of a dynamic backout failure.

Fields Present in a CAGRP Record

A change accumulation group record’s fields and their corresponding line numbers are described in Table 43.

Table 43. Fields Present in a CAGRP Record

Record Type	Line Number	Field	Contents
CAGRP	1		
	2	GRPNAME= *cagrpnam*	The name of the CA group.
		GRPMAX= *nnnn*	The maximum number of change accumulation run records that can be associated with this group, whether available or in use.
		CA AVAIL= *nnnn*	The number of available change accumulation run records currently in RECON for this group.
		CA USED= *nnnn*	The number of in-use change accumulation run records currently in RECON for this group.

Table 43. Fields Present in a CAGRP Record (continued)

Record Type	Line Number	Field	Contents
	3	REUSE ǀ NOREUSE	Indicates whether change accumulation data sets can be reused and whether empty ones can be created for subsequent use (REUSE) or not (NOREUSE).
		CAJCL= *cajclmem*	The name of the member of a partitioned data set that contains skeletal JCL for this CA group.
		DEFLTJCL= *member*	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.
	4	#MEMBERS= *nnnn*	This number of DBDSs and areas that belong to this CA group.
		-DBD- -DDN-	These headings and the following lines are printed only if some DBDSs are members of this group.
	5-*nnn*	dbdname ddname	The database name and data set ddname of a DBDS that is a member of this CA group.

Fields Present in a CA Record

A change accumulation record's fields and their corresponding line numbers are described in Table 44.

Table 44. Fields Present in a CA Record

Record Type	Line Number	Field	Contents
CA	1		
	2	*DSN= *chge.accum.dsname*	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.
		FILE SEQ= *nnnn*	The file sequence number of the first volume of this data set.
	3	CAGRP= *cagrpname*	Name of the CA group to which the change accumulation record belongs.
		UNIT= *unittype*	Unit type to be used for substitution during the GENJCL process.
	4	CREATE ǀ STOP= *time_stamp**	CREATE is printed if the data set is available for future use. In this case, the time stamp is the time that the record was created or made available for recycling. STOP is printed for in-use records. In this case, the time stamp is the stop time of the last log volume that was processed by the Database Change Accumulation utility. If SUBSET is printed (see line 6), the time stamp is the start time of the first unselected log. The asterisk (*) indicates that this time stamp is in the record key.

Table 44. Fields Present in a CA Record (continued)

Record Type	Line Number	Field	Contents
		VOLS DEF= *nnnn* VOLS USED= *nnnn*	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.
	5-*nnn*	VOLSER= *volser1, volser2,...*	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.
The following lines are printed only if the change accumulation data set has been used during a run of the Change Accumulation utility (the change accumulation data set is no longer available):			
	6	RUN= *time_stamp*	This time stamp represents the time of the run of the Change Accumulation utility during which this record was used.
		ERR	This indicates that you have marked the change accumulation data set as unusable because of a previous error.
		SUBSET	This indicates that a subset of logs were processed when the CA was created.
		*	The asterisk (*) indicates that COMP or SUBSET were specified in the `CHANGE.CA` or `NOTIFY.CA` commands.
These headings are printed only if purge times exist in the record. If purge times exist, there is one line in the format labeled below (7-9) for each DBDS in the CA group.			

	7	DBD= *dbdname*
		DDN= *ddname*
			The database name and data set ddname of the DBDS.

		PURGETIME= *time_stamp*	Change records occurring before this time stamp for the corresponding DBDS have been ignored by the Database Change Accumulation utility.
	8	CHANGES ACCUMULATED= YES | NO	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.
		COMPLETE CA= YES | NO	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.
		INDOUBT EEQES= YES | NO	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.
	9	LSN= *lsn*	The lock sequence number of the last change accumulated for the DBDS is listed.
		DSSN= *dssn*	The data set sequence number of the last change accumulated for the DBDS is listed.
	10	LRID= *log_record_ID*	The last log record ID of the last change accumulated for the DBDS is listed in this field.

Table 44. Fields Present in a CA Record (continued)

Record Type	Line Number	Field	Contents
		USID= *update_set_ID*	The last update set ID of the last change accumulated for the DBDS is listed in this field.

Fields Present in a Data Group Record

A database data group record's fields and their corresponding line numbers are described in Table 45.

Table 45. Fields Present in the DBDSGRP Record

Record Type	Line Number	Field	Contents
DBDSGRP DBGRP or RECOVGRP	1		Type of group: DBDS, DB, or Recovery.
	2	GRPNAME= *grpname*	Name of the group.
		#MEMBERS= *nnn*	Number of members belonging to the group.
		-DBD- -DDN/AREA- -DBD/AREA- or -DBD- -AREA-	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-`.
	3-*nnn*	*dbdname ddname/areaname*	For DBDSGRP, *dbdname* = DB name, *ddname/areaname* = DBDS name or area name. For DBGRP, *dbdname* = DL/I DB name or DEDB area name. *ddname/areaname* is blank. For RECOVGRP, *dbdname* = DB name, *ddname/areaname* = area name if DB is a DEDB, or blank if DB is DL/I.

Fields Present in a DB (IMS) Record

An IMS database record's fields and their corresponding line numbers are described in Table 46.

Table 46. Fields Present in the DB (IMS) Record

Record Type	Line Number	Field	Contents
DB (IMS)	1		
	2	DBD= *dbdname*	Name of the database.
		IRLMID= *xxxxx*	Identifies the IRLM when the share level of this database is 2. This field is printed only when SHARE LEVEL=2.
		DMB#= *nnn*	The value assigned when this database was registered.
		TYPE= IMS	Indicates that this is a DL/I database (rather than a Fast Path DEDB).
	3	SHARE LEVEL= *n*	The level of data sharing for which authorized subsystems can share this database.
		GSGNAME= *gsgname*	Identifies the name of the GSG to which this database belongs.
		USID= *n*	The highest update set identifier for this database.

Table 46. Fields Present in the DB (IMS) Record (continued)

Record Type	Line Number	Field	Contents
	4	AUTHORIZED USID= *n*	The identifier of the current update set that is being applied to the database.
		RECEIVE USID= *n*	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.
		HARD USID= *n*	The update set identifier of the latest changes that were written to the database.
	5	RECEIVE NEEDED USID= *n*	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".
	6	DBRCVGRP=*rcvgrpnm*	The name of the recovery group to which this database belongs, if any. Otherwise, `**NULL**` is printed.
	7	FLAGS: COUNTERS:	This heading line is printed for lines 8 through 14, which describe the status of this database.
	8	BACKOUT NEEDED= ON | OFF	This flag indicates whether this database needs to be backed-out by any subsystem.
		RECOVERY NEEDED COUNT= *n*	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.
	9	READ ONLY= ON | OFF	Indicates whether this database can be authorized for read processing only, or authorized for read and update processing.
		IMAGE COPY NEEDED COUNT= *n*	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.
	10	PROHIBIT AUTHORIZATION= ON | OFF	This flag indicates whether this database is available for authorization processing. If the database can be authorized, the value is OFF.
		AUTHORIZED SUBSYSTEMS= *n*	The count of subsystems that have current authorization to this database.
	11	RECOVERABLE= YES | NO	An indication of whether the database is recoverable (YES) or non-recoverable (NO).
		HELD AUTHORIZATION STATE= *n*	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*> 0, see line the 17 `encoded state` field below for the subsystem encoded state. If *n*> 128, block-level data sharing is in effect. Subtract 128 to determine the true held authorization state.

Table 46. Fields Present in the DB (IMS) Record (continued)

Record Type	Line Number	Field	Contents
	12	DATABASE LEVEL TRACK= YES | NO	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.
		EEQE COUNT= *n*	Number of extended error queue elements for this database.
	13	TRACKING SUSPENDED= YES | NO	Indicates whether tracking has been suspended for this shadow database.
		RECEIVE REQUIRED COUNT= *n*	Indicates how many of the DBDSs (for this shadow database) need image copies to be received from the active site.
	14	OFR REQUIRED= YES | NO	Indicates whether online forward recovery is required for this shadow database.
	15	IC NEEDED DISABLED	For this non-recoverable DB, image copies will not be required after initial load (PROCOPT=L) or REORG of the DBDS.
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 47..

Table 47. Fields Present in the DB (HALDB) Record

Record Type	Line Number	Field	Contents
DB (HALDB Master)	1		
	2	DBD=*dbdname*	Name of the HALDB.
		DMB#=*nnn*	The value assigned when this database was registered.

Table 47. Fields Present in the DB (HALDB) Record (continued)

Record Type	Line Number	Field	Contents
		CHANGE#=*nnn*	The change version number. This is updated when the definition of the HALDB is changed.
		TYPE=HALDB	Indicates that this is the HALDB master.
	3	SHARE LEVEL=*n*	The level of data sharing for which authorized subsystems can share any partition of this HALDB.
		GSGNAME=*gsgname*	Identifies the name of the GSG to which this database belongs.
	4	PSNAME=*psname*	The name of the partition selection routine used by this HALDB, if any. Otherwise, **NULL** is printed.

		DBORG = *dbaseorg*
		DSORG = *dsetorg*
			The database and data set organization of the data DBDSs in this HALDB.

		CURRENT PARTITION ID = *nnnnn*	The partition ID of the latest partition defined for the HALDB.
	5	FLAGS: COUNTERS:	This heading line is printed for line 6, which describe the status of this HALDB.
	6	RECOVERABLE= YES | NO	The HALDB is recoverable or not.
		PARTITIONS=*n*	Number of partitions that exist for this HALDB.
|	7	ONLINE REORG CAPABLE= YES | NO	The HALDB is capable (or not) for online reorganization. The ONLINE REORG CAPABLE = YES | NO flag is included in the master HALDB and in the DBD partition record.
		Data Set Group Members	Number of data set group members
		IC NEEDED DISABLED	For this non-recoverable DB, image copies will not be required after initial load (PROCOPT=L) or REORG of the DBDS.

Fields Present in a DB (PART) 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 (PART) Record

Record Type	Line Number	Field	Contents
DB (HALDB Partition)	1		
	2	DBD=*dbdname*	Name of the HALDB partition.
		MASTER DB=*HALDB master name*	Name of the HALDB master.
		IRLMID=*xxxxx*	Identifies the IRLM name. This field is printed only when SHARE LEVEL>=2.
		CHANGE#=*nnn*	The change version number is updated when the definition of the partition is changed.
		TYPE=PART	Indicates a HALDB partition.
	3	USID=*n*	The highest update set identifier for this partition.

Table 48. Fields Present in the DB (PART) Record (continued)

Record Type	Line Number	Field	Contents
		AUTHORIZED USID=*n*	The identifier of the current update set being applied to the partition.
		HARD USID=*n*	The update set identifier of the latest changes written to the partition.
	4	RECEIVE USID=*n*	The update set identifier of the last image copy received for this partition, which applies only to shadow databases at the tracking site.
		RECEIVE NEEDED USID=*n*	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.
	5	DBRCVGRP=*rcvgrpnm*	The name of the recovery group to which this partition belongs, if any. Otherwise, `**NULL**` is printed.
	5A	DSN PREFIX=*dsname*	The prefix name used for all data sets
		PARTITION ID=*nnnnn*	The partition identifier.
	5B	PREVIOUS PARTITION= *name* NEXT PARTITION=*name*	The names of the previous and next partitions, for HALDBs using high keys.
	6	OLRIMSID=**NULL**	The OLRIMSID field reflects the name of the IMS that owns the online reorganization. OLRIMSID is always null at the tracking site.
		ACTIVE DBDS=A-J or ACTIVE DBDS=M-V	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.
	7-8	RANDOMIZER: NAME= ANCHOR= HIGH BLOCK#= BYTES=nnnnnnnn	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.
	9-10	FREE SPACE: FREE BLOCK FREQ FACTOR=nnn FREE SPACE PERCENTAGE=nn	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.
	11a	PARTITION HIGH KEY/STRING (CHAR): (LENGTH=NNN)	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).

Table 48. Fields Present in the DB (PART) Record (continued)

Record Type	Line Number	Field	Contents
	11b	PARTITION HIGH KEY/STRING (HEX):	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).
	12	OSAM BLOCK SIZE: s = nnnnn	OSAM block size is listed for each data set group member defined. s is the data set group identifier A through J.
	13	FLAGS: COUNTERS:	This heading line is printed for lines 7 through 14, which describe the status of this partition.
	14	BACKOUT NEEDED=ON|OFF	This flag indicates whether this partition needs backout by any subsystem.
		RECOVERY NEEDED COUNT=*n*	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.
	15	READ ONLY=ON	OFF	Indicates whether this partition can be authorized for read processing only, or authorized for read and update processing.
		IMAGE COPY NEEDED COUNT=*n*	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.
	16	PROHIBIT AUTHORIZATION=ON	OFF	This flag indicates whether this partition is available for authorization processing. If the partition can be authorized, the value is OFF.
		AUTHORIZED SUBSYSTEMS=*n*	The count of subsystems that have current authorization to this partition.
	17	HELD AUTHORIZATION STATE=*n*	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* > 0, see the table below for the subsystem encoded state. If *n* > 128, online image copy (OLIC) is in progress for a data set of this partition.
	18	DATABASE LEVEL TRACK=YES	NO	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.
		EEQE COUNT=*n*	Number of extended error queue elements for this partition.
	19	TRACKING SUSPENDED=YES	NO	Indicates whether tracking has been suspended for this shadow database.

Table 48. Fields Present in the DB (PART) Record (continued)

Record Type	Line Number	Field	Contents
		RECEIVE REQUIRED COUNT=*n*	Indicates how many of the DBDSs (for this shadow database) need image copies from the active site to be received.
	20	OFR REQUIRED=YES | NO	Indicates whether online forward recovery is required for this shadow database.
		OLR ACTIVE HARD COUNT=0	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.
	21	PARTITION INIT NEEDED=YES | NO	Indicates whether partition initialization is required for this HALDB partition.
		OLR INACTIVE HARD COUNT=0	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.
	22	OLREORG CURSOR ACTIVE=NO	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.
	23	PARTITION DISABLED=YES | NO	A disabled partition is known only to DBRC and the Partitition Definition Utility. A disabled partition can be enabled or deleted.
The following lines 24 through 27 are printed only when one or more subsystems are currently authorized to this partition:			
	24,25,26	ASSOCIATED SUBSYSTEM INFORMATION: -SSID- -ACCESS INTENT- ENCODED -STATE- B/O NEEDED -COUNT- -SS ROLE-	
	27-*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 partition. 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

Table 48. Fields Present in the DB (PART) Record (continued)

Record Type	Line Number	Field	Contents
		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 49.

Table 49. Fields Present in the DB (Fast Path) Record

Record Type	Line Number	Field	Contents
DB (Fast Path)	1		
	2	DBD= *dbdname*	Name of the database.
		DMB#= *nnn*	The value assigned when this database was registered.
		TYPE= FP	Indicates that the database being used is a Fast Path DEDB.
	3	SHARE LEVEL= *n*	The level of data sharing for which authorized subsystems can share this database.
	4	FLAGS: COUNTERS:	This heading line is printed for lines 4 through 6 describing status of this database.
	5	RECOVERY NEEDED COUNT= *n*	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.
	6	IMAGE COPY NEEDED COUNT= *n*	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.
	7	PROHIBIT AUTHORIZATION= ON | OFF	This flag indicates whether this database is available for authorization processing. If the database can be authorized, the value is OFF.
		AUTHORIZED AREAS= *n*	The count of areas that are currently authorized.
	8	EEQE COUNT= *n*	Number of extended error queue elements for this DEDB.

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 50.

Table 50. Fields Present in the DBDS (non-Fast Path) Record

Record Type	Line Number	Field	Contents
DBDS (non-Fast Path)	1		
	2	DSN= *dsname*	Data set name of the DBDS.

Table 50. Fields Present in the DBDS (non-Fast Path) Record (continued)

Record Type	Line Number	Field	Contents
		TYPE= IMS	Indicates that this is a DBDS rather than a FP DEDB area.
		TYPE= PART	Indicates that this is a DBDS of a HALDB partition.

	3	DBD= *dbdname*
		DDN= *ddname*
			The database name and data set ddname of the DBDS.

		DSID= *nn*	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.
		DBORG= *dbaseorg* DSORG= *dsetorg*	The database and data set organization, as defined for the DBDS in the DBDLIB data set.
	4	CAGRP= *cagrpnam*	The name of the CA group to which this DBDS belongs, if any. Otherwise, `**NULL**` is printed.
		GENMAX= *nnnn*	The maximum number of image copy data sets to be maintained for this DBDS.
		IC AVAIL= *nnnn* IC USED= *nnnn*	The number of available and in-use image copy data sets for the DBDS.
		DSSN= *nnnn*	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.
	5	REUSE | NOREUSE	REUSE is printed if you have specified in RECON that image copy data sets are to be reused for this DBDS.
		RECOVPD= *nnn*	This is the recovery period of the image copies.
		OTHER DDN=**NULL**	The partner DBDS, if one exists. Otherwise `**NULL**` is printed.
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:			
	6	DEFLTJCL= *member*	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.
		ICJCL= *member*	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.
		OICJCL= *member*	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.
		RECOVJCL= *member*	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.
	7	RECVJCL= *member*	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.

Table 50. Fields Present in the DBDS (non-Fast Path) Record (continued)

Record Type	Line Number	Field	Contents
	8	FLAGS: COUNTERS:	Heading for the following lines 9 through 11, which describe the status of this DBDS.
	9	IC NEEDED= ON ∣ OFF	Indicates whether an image copy needs to be taken for the DBDS.
	10	IC RECOMMENDED= ON	Indicates that DBRC recommends an image copy of the DBDS should be taken before using the database.
	11	RECOV NEEDED= ON ∣ OFF	Indicates whether the DBDS needs to be recovered.
	12	RECEIVE NEEDED = ON ∣ OFF	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.
		EEQE COUNT= *n*	The number of extended error queue elements for this DBDS.
The following, lines 13 through 15, are printed only if one or more extended error queue elements exist.			
	13, 14	ERROR QUEUE ELEMENTS: -EQERBA-EEQE TYPE-SSID-	Heading for the following list of extended error queue elements (lines 14-nnn).
	15-*nnn*	*eeqe rba*	The relative byte address (RBA) of the EEQE.
		eeqe type	The type of extended error queue element.
		ssid	The ID of the subsystem that created the EEQE (in doubt EEQEs only).

Fields Present in a DBDS (Fast Path) Record

A Fast Path database data set record's fields and their corresponding line numbers are described in Table 51.

Table 51. Fields Present in the DBDS (Fast Path) Record

Record Type	Line Number	Field	Contents
DBDS (Fast Path)	1		

	2	DBD= *dbdname*
		AREA= *areaname*
			The database name and area name of the Fast Path DEDB.

		IRLMID= *irlmname*	Identifies the IRLM when the share level of this DEDB is 2. This field is printed only when SHARE LEVEL=2.
		TYPE= FP	Identifies this database as a Fast Path DEDB.
	3	SHARE LEVEL= *n*	The level of data sharing for which authorized subsystems can share this area.
		DSID= *nn*	The data set ID number that appears as part of the information in the DBDLIB data set about the area data set.
		DBORG= *dbaseorg* DSORG= *dsetorg*	The database and data set organization, as defined for the area data set in the DBDLIB data set.
	4	GSGNAME= *gsgname*	Identifies the name of the GSG to which this area belongs.

Table 51. Fields Present in the DBDS (Fast Path) Record (continued)

Record Type	Line Number	Field	Contents
		USID= *n*	The highest update set identifier for this area.
	5	AUTHORIZED USID= *n*	The identifier of the current update set that is being applied to the area.
		RECEIVE USID= *n*	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.
		HARD USID= *n*	The update set identifier of the latest changes that were written to the area.
	6	RECEIVE NEEDED USID= *n*	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.
	7	CAGRP= *cagrpnam*	The name of the CA group to which this area belongs, if any. Otherwise, `**NULL**` is printed.
		GENMAX= *nnnn*	The maximum number of image copy data sets to be maintained for this area.
		IC AVAIL= *nnnn* IC USED= *nnnn*	The number of available image copy data sets, and the number of in-use image copy data sets for the area.
		DSSN= *nnnn*	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.
	8	REUSE | NOREUSE	REUSE is printed if you have specified in RECON that image copy data sets are to be reused for this area data set.
		RECOVPD= *nnn*	This is the recovery period of the image copies.
		VSO | NOVSO	Indicates whether the area resides in virtual storage.
		PREOPEN | NOPREOPEN	Indicates whether the area is opened at control region initialization or when the area is started.
		PRELOAD | NOPRELOAD	Indicates whether the VSO area is loaded into the data space the next time it is opened.
	9	CFSTR1= *cfstr_name*	The name of the first coupling facility structure for the area.
		CFSTR2= *cfstr_name*	The name of the second coupling facility structure for the area.
		LKASID | NOLKASID	Indicates whether local data caching for the specified area is used for buffer lookaside on read requests.
|		MAS | NOMAS	Indicates whether (or not) the area is to reside in a multi-area coupling facility structure.
	10	DEFLTJCL= *member*	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.

Table 51. Fields Present in the DBDS (Fast Path) Record (continued)

Record Type	Line Number	Field	Contents
		ICJCL= *member*	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.
		RECVJCL= *member*	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.
		RECOVJCL= *member*	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.
	11	DBRCVGRP=*rcvgrpnm*	The name of the recovery group to which this DBDS belongs, if any. Otherwise, `**NULL**` is printed.
	12	FLAGS: COUNTERS:	This heading line is printed for lines 11 through 19 describing the status of this area.
	13	PROHIBIT AUTHORIZATION= ON | OFF	The value of this flag is OFF if the area is available for authorization processing.
		AUTHORIZED SUBSYSTEMS= *n*	The count of subsystems that have current authorization to this area.
	14	HELD AUTHORIZATION STATE= *n*	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*> 0, see the table below for the subsystem encoded state. If *n*> 128, block-level data sharing is in effect. Subtract 128 to determine the true held authorization state.
	15	IC NEEDED= ON | OFF	Indicates whether an image copy needs to be taken for the DEDB area.
		ADS AVAIL #= *n*	Indicates the number of available ADS in this area record.
	16	IC RECOMMENDED= ON	Indicates that DBRC recommends an image copy of the area should be taken before it is used.
	17	RECOV NEEDED= ON | OFF	Indicates whether the areas associated with the DEDB should be recovered.
		REGISTERED ADS #= *n*	Indicates how many area data sets for this area are registered in RECON.
	18	DATABASE LEVEL TRACK = YES | NO	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.
		EEQE COUNT= *n*	The number of extended error queue elements for this DEDB area.
	19	RECEIVE NEEDED = ON | OFF	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.

Table 51. Fields Present in the DBDS (Fast Path) Record (continued)

Record Type	Line Number	Field	Contents
	20	OFR REQUIRED= YES I NO	Indicates whether online forward recovery is required for this shadow area.
	21	TRACKING SUSPENDED= YES I NO	Indicates whether tracking has been suspended for this shadow area.
	22	HSSP CIC IN PROGRESS= YES I NO	Indicates whether an HSSP concurrent image copy is in progress.
The following lines, 23 through 25, are printed only if one or more error queue elements exist.			
	23 24	ERROR QUEUE ELEMENTS: -EQERBA-EEQE TYPE-SSID-	Heading for the following list of error queue elements.
	25-*nnn*	*eeqe rba*	The type of extended error queue element.
		eeqe type	The type of extended error queue element.
		ssid	The ID of the subsystem that created the EEQE (in doubt EEQEs only).
If the number of registered area data sets is not zero, the following lines 26 through 28 list those area data sets that are currently registered in this area record:			
	26 27 28	ADS LIST: -ADS DDN- -ADS DSN- -STAT- CREATE -RUNNING-	These lines, 26 through 28, represent the column headings for the following line 29.
The following line (29) is repeated for each area data set that is registered for this area.			
	29-*nn*	*adsddn* *adsdsn* AVAIL I UNAVAIL YES I NO	AVAIL indicates that the area data set is available. UNAVAIL indicates that the area data set is unavailable. YES indicates that the area data set is being used. NO indicates that the area data set is not being used.
The following lines *nn+1* through *nn+3*,are printed only when one or more subsystems are currently authorized to this area:			
	nn+1 *nn*+2 *nn*+3	ASSOCIATED SUBSYSTEM INFORMATION -SSID- -ACCESS INTENT- ENCODED -STATE- -SS ROLE-	Lines *nn+1* through *nn+3* represent the column headings for the following lines *nn*+3-*mm*.
The following line *nn*+3-*mm*, is repeated for each subsystem that has authorization for this database.			
	nn+3-*mm*	*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
		ACTIVE I TRACKING	Role of the authorized subsystem.

Fields Present in an ALLOC Record

An allocation record's fields and their corresponding line numbers are described in Table 52 on page 549.

Table 52. Fields Present in the ALLOC Record

Record Type	Line Number	Field	Contents
ALLOC	1		
	2	ALLOC= *time_stamp**	The time stamp for a time that the DBDS was allocated during the run of an IMS system. An asterisk (*) indicates that it is part of the record key.
		ALLOC LRID= *n*	The log record sequence number of the begin update log record. The value for *n* is 0 for ALLOC records that are created at the active site.
	3	DSSN= *nnnn*	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.
		USID= *n*	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.
		START= *time_stamp*	The start time of the log data set that was logging change records when this DBDS was allocated.
	4	DEALLOC= *time_stamp*	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.
		DEALLOC LRID= *n*	The log record sequence number of the end-update log record. This field is only printed if a specific deallocation time exists.
	5	TRACKING SUSPENDED AT RECORD: *suspend lrid* TRACKING SUSPENDED NO RECORDS APPLIED	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.

Fields Present in an IMAGE Record

An image record's fields and their corresponding line numbers are described in Table 53.

Table 53. Fields Present in the IMAGE Record

Record Type	Line Number	Field	Contents
IMAGE	1		
	2	*	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).

Table 53. Fields Present in the IMAGE Record (continued)

Record Type	Line Number	Field	Contents
		CREATE | RUN= *time_stamp**	CREATE is printed if the image copy data set has never been used. Otherwise, RUN is printed. RUN is the time stamp of the start of processing of the database image copy utilities that used this image copy data set. The asterisk (*) indicates that the time stamp is part of the key of this record.
		RECORD COUNT= *nnnnnnnn*	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.
		USER IC	This is printed for any nonstandard image copy.
		USID= *n*	This is only listed here for a nonstandard image copy.
	3	USERDATA= *cccccccc*...	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.
		STOP= *time_stamp*	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.
		ONLINE | BATCH | CONCUR | SMSCIC | SMSNOCIC	This field indicates the type of image copy. ONLINE indicates that this image copy was produced by an online database image copy utility. BATCH indicates a batch image copy from the Database Image Copy utility. CONCUR indicates that a concurrent image copy was run. SMSCIC indicates a concurrent image copy that is produced by the Database Image Copy 2 utility. SMSNOCIC indicates a consistent image copy that is produced by Database Image Copy 2 utility. This field is not printed for available image copy data sets.
		USID= *n*	This is only listed here for a standard image copy. This is the value of the update set identifier of the database or area when the image copy was taken.
	4	IN PROGRESS	This line is listed when an HSSP concurrent image copy is currently in progress.
	5	IC1 | IC2	IC1 and accompanying lines are always printed, followed by IC2 and accompanying lines, if a duplicate image copy data set exists.
	6	DSN= *ic-dsname*	*ic-dsname* is the data set name of this image copy data set (IC1 or IC2).

Table 53. Fields Present in the IMAGE Record (continued)

Record Type	Line Number	Field	Contents
		FILE SEQ= *nnnn*	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.
	7	UNIT= *unittype*	Unit type to be used for substitution during the GENJCL process. This field is listed for non-HSSP image copies only.
		ERR | EMP	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.
		VOLS DEF= *nnnn* VOLS USED= *nnnn*	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.
	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 54.

Table 54. Fields Present in the REORG Record

Record Type	Line Number	Field	Contents
REORG	1		
	2	RUN= *time_stamp**	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.
		USID= *n*	This is the value of the update set identifier of the database or area when the reorganization of this DBDS occurred.
|	3	STOP=*time_stamp**	STOP= is only present for online reorganizations. STOP= is zero for online reorganizations that have not yet completed.
|		ONLINE RECOV= YES | NO	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.

Table 54. Fields Present in the REORG Record (continued)

Record Type	Line Number	Field	Contents
	4	OLDSTOPT = *time_stamp*	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.

Fields Present in a RECOV Record

A recovery record's fields and their corresponding line numbers are described in Table 55.

Table 55. Fields Present in the RECOV Record

Record Type	Line Number	Field	Contents
RECOV	1		
	2	RUN= *time_stamp**	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.
		RUN USID= *n*	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.
	3	RECOV TO= *time_stamp*	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.
		RECOV TO USID= *n*	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.

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 83.

Table 56 on page 553 lists the resources that can be protected.

Table 56. Resource Names for Command Authorization

Verb	Modifier	Qualifier
BACKUP	.RECON	
CHANGE	.ADS	.dbname
CHANGE	.BKOUT	.ssid
CHANGE	.CA	.grpname
CHANGE	.CAGRP	.grpname
CHANGE	.DB	.dbname
CHANGE	.DB	.ALL
CHANGE	.DBDS	.dbname
CHANGE	.DBDSGRP	.grpname
CHANGE	.IC	.dbname
CHANGE	.PART	.dbname
CHANGE	.PRILOG	.OLDS
CHANGE	.PRILOG	.RLDS
CHANGE	.PRILOG	.SLDS
CHANGE	.PRILOG	.TSLDS
CHANGE	.RECON	.CATDS
CHANGE	.RECON	.NOCATDS
CHANGE	.RECON	.CHECK17
CHANGE	.RECON	.CHECK44
CHANGE	.RECON	.NOCHECK
CHANGE	.RECON	.CMDAUTH
CHANGE	.RECON	.DASDUNIT
CHANGE	.RECON	.DUAL
CHANGE	.RECON	.REPLACE
CHANGE	.RECON	.FORCER
CHANGE	.RECON	.NOFORCER
CHANGE	.RECON	.IMSPLEX
CHANGE	.RECON	.NOPLEX
CHANGE	.RECON	.LISTDL
CHANGE	.RECON	.NOLISTDL
CHANGE	.RECON	.LOGALERT
CHANGE	.RECON	.LOGRET
CHANGE	.RECON	.MINVERS

Table 56. Resource Names for Command Authorization (continued)

Verb	Modifier	Qualifier
CHANGE	.RECON	.SIZALERT
CHANGE	.RECON	.SSID
CHANGE	.RECON	.STARTNEW
CHANGE	.RECON	.NONEW
CHANGE	.RECON	.TAPEUNIT
CHANGE	.RECON	.TIMEZONE
CHANGE	.RECON	.TIMEZIN
CHANGE	.RECON	.TIMEFMT
CHANGE	.RECON	.TRACEON
CHANGE	.RECON	.TRACEOFF
CHANGE	.RECON	.UPGRADE
CHANGE	.SG	.gsgname
CHANGE	.SUBSYS	.ssid
CHANGE	.SUBSYS	.ALL
CHANGE	.UIC	.dbname
DELETE	.ADS	.dbname
DELETE	.ALLOC	.dbname
DELETE	.BKOUT	.ssid
DELETE	.CA	.grpname
DELETE	.CAGRP	.grpname
DELETE	.DB	.dbname
DELETE	.DBDS	.dbname
DELETE	.DBDSGRP	.grpname
DELETE	.GSG	.gsgname
DELETE	.IC	.dbname
DELETE	.LOG	.INACTIVE
DELETE	.LOG	.OLDS
DELETE	.LOG	.STARTIME
DELETE	.LOG	.TOTIME
DELETE	.PART	.dbname
DELETE	.RECOV	.dbname
DELETE	.REORG	.dbname
DELETE	.SG	.gsgname
DELETE	.SUBSYS	.ssid
DELETE	.UIC	.dbname
GENJCL	.ARCHIVE	.ssid
GENJCL	.CA	.grpname
GENJCL	.CLOSE	.ssid
GENJCL	.IC	.dbname
GENJCL	.IC	.grpname

Table 56. Resource Names for Command Authorization (continued)

Verb	Modifier	Qualifier
GENJCL	.OIC	.dbname
GENJCL	.OIC	.grpname
GENJCL	.RECEIVE	.dbname
GENJCL	.RECEIVE	.grpname
GENJCL	.RECOV	.dbname
GENJCL	.RECOV	.grpname
GENJCL	.USER	.mbrname
GENJCL	.USER	.dbname
GENJCL	.USER	.grpname
INIT	.ADS	.dbname
INIT	.CA	.grpname
INIT	.CAGRP	.grpname
INIT	.DB	.dbname
INIT	.DBDS	.dbname
INIT	.DBDSGRP	.grpname
INIT	.GSG	.gsgname
INIT	.IC	.dbname
INIT	.PART	.dbname
INIT	.RECON	
INIT	.SG	.gsgname
LIST	.BKOUT	.ssid
LIST	.BKOUT	.ALL
LIST	.CAGRP	.grpname
LIST	.CAGRP	.ALL
LIST	.DB	.dbname
LIST	.DB	.ALL
LIST	.DB	.TYPEIMS
LIST	.DB	.TYPEFP
LIST	.DB	.TYPHALDB
LIST	.DBDS	.dbname
LIST	.DBDS	.grpname
LIST	.DBDSGRP	.grpname
LIST	.DBDSGRP	.ALL
LIST	.GSG	.gsgname
LIST	.GSG	.ALL
LIST	.HISTORY	.dbname
LIST	.HISTORY	.grpname
LIST	.LOG	.ALL
LIST	.LOG	.ALLOLDS
LIST	.LOG	.OLDS

Table 56. Resource Names for Command Authorization (continued)

Verb	Modifier	Qualifier
LIST	.LOG	.STARTIME
LIST	.RECON	
LIST	.RECON	.STATUS
LIST	.SUBSYS	.ONLINE
LIST	.SUBSYS	.ssid
LIST	.SUBSYS	.ALL
LIST	.SUBSYS	.BATCH
NOTIFY	.ALLOC	.dbname
NOTIFY	.BKOUT	.ssid
NOTIFY	.CA	.grpname
NOTIFY	.IC	.dbname
NOTIFY	.PRILOG	.OLDS
NOTIFY	.PRILOG	.RLDS
NOTIFY	.PRILOG	.SLDS
NOTIFY	.PRILOG	.TSLDS
NOTIFY	.PRILOG	.DPROP
NOTIFY	.RECOV	.dbname
NOTIFY	.REORG	.dbname
NOTIFY	.SECLOG	.OLDS
NOTIFY	.SECLOG	.RLDS
NOTIFY	.SECLOG	.SLDS
NOTIFY	.SECLOG	.TSLDS
NOTIFY	.SECLOG	.DPROP
NOTIFY	.SUBSYS	.ssid
NOTIFY	.UIC	.dbname
RESET	.GSG	.gsgname

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
DB2 UDB for z/OS
DFSMS/MVS
DFSMSdss
IBM
IMS
IMS/ESA
Softcopy Library Reader
OS/390
RACF
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.

Product Names

In this book, the licensed program “DB2 UDB for z/OS®” is referred to as “DB2”.

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*
- *DFSMS/MVS V1R5 MDFSMSdfp Advanced Services, SC26-4921*
- *High Level Assembler for MVS & VM & VSE Language Reference, SC26-4940*
- *High Level Assembler for MVS & VM & VSE Programmer’s Guide, SC26-4941*
- *OS/390 MVS Planning: Global Resource Serialization, GC28-1759*
- *OS/390 MVS Programming: Authorized Assembler Services Reference, GC28-1763*
- *z/OS DFSMS 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 Version 9 Library

ZES1-2330	ADB	*IMS Version 9: Administration Guide: Database Manager*
ZES1-2331	AS	*IMS Version 9: Administration Guide: System*
ZES1-2332	ATM	*IMS Version 9: Administration Guide: Transaction Manager*
ZES1-2333	APDB	*IMS Version 9: Application Programming: Database Manager*
ZES1-2334	APDG	*IMS Version 9: Application Programming: Design Guide*
ZES1-2335	APCICS	*IMS Version 9: Application Programming: EXEC DLI Commands for CICS and IMS*
ZES1-2336	APTM	*IMS Version 9: Application Programming: Transaction Manager*
ZES1-2337	BPE	*IMS Version 9: Base Primitive Environment Guide and Reference*
ZES1-2338	CR	*IMS Version 9: Command Reference*
ZES1-2339	CQS	*IMS Version 9: Common Queue Server Guide and Reference*
ZES1-2340	CSL	*IMS Version 9: Common Service Layer Guide and Reference*
ZES1-2341	CG	*IMS Version 9: Customization Guide*
ZES1-2342	DBRC	*IMS Version 9: DBRC Guide and Reference*
ZES1-2343	DGR	*IMS Version 9: Diagnosis Guide and Reference*
ZES1-2344	FAST	*IMS Version 9: Failure Analysis Structure Tables (FAST) for Dump Analysis*
ZES1-2346	OLR	*IMS Version 9: HALDB Online Reorganization Guide*
ZES1–2380	CT	*IMS Version 9: IMS Connect Guide and Reference*
ZES1-2347	JGR	*IMS Version 9: IMS Java Guide and Reference*
ZES1-2348	IIV	*IMS Version 9: Installation Volume 1: Installation Verification*
ZES1-2349	ISDT	*IMS Version 9: Installation Volume 2: System Definition and Tailoring*
ZES1-2350	INTRO	*IMS Version 9: An Introduction to IMS*
ZES1-2351	MIG	*IMS Version 9: Master Index and Glossary*
ZES1-2352	MC1	*IMS Version 9: Messages and Codes, Volume 1*
ZES1-2353	MC2	*IMS Version 9: Messages and Codes, Volume 2*
ZES1-2354	OTMA	*IMS Version 9: Open Transaction Manager Access Guide and Reference*
ZES1-2355	OG	*IMS Version 9: Operations Guide*
GC17-7831	RPG	*IMS Version 9: Release Planning Guide*
ZES1-2358	URDBTM	*IMS Version 9: Utilities Reference: Database and Transaction Manager*
ZES1-2359	URS	*IMS Version 9: Utilities Reference: System*

Supplementary Publications

GC17-7825	LPS	*IMS Version 9: Licensed Program Specifications*
ZES1-2357	SOC	*IMS Version 9: Summary of Operator Commands*

	Publication Collections	
LK3T-7213	CD	IMS Version 9 Softcopy Library
LK3T-7144	CD	IMS Favorites
LBOF-7789	Hardcopy and CD	Licensed Bill of Forms (LBOF): IMS Version 9 Hardcopy and Softcopy Library
SBOF-7790	Hardcopy	Unlicensed Bill of Forms (SBOF): IMS Version 9 Unlicensed Hardcopy Library
SK2T-6700	CD	OS/390 Collection
SK3T-4270	CD	z/OS Software Products Collection
SK3T-4271	DVD	z/OS and Software Products DVD Collection

	Accessibility Titles Cited in this Book	
SA22-7787		z/OS V1R1.0 TSO Primer
SA22-7794		z/OS V1R1.0 TSO/E User's Guide
SC34-4822		z/OS V1R1.0 ISPF User's Guide, Volume 1

Index

Program Number: 5655-J38

IBM Confidential
Printed in USA

ZES1-2342-02

Spine information:

IBM IMS DBRC Guide and Reference Version 9