IMS/ESA

DBRC Guide and Reference

Version 6

SC26-8733-04

IMS/ESA

DBRC Guide and Reference

Version 6

SC26-8733-04

Note

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

Fifth Edition (July 2000) (Softcopy only)

This edition replaces and makes obsolete the previous edition, SC26-8733-03. This edition is available in softcopy only. The technical changes for this edition are summarized under “Summary of Changes” on page xxi. The technical changes for this edition are indicated by a vertical bar to the left of a change.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

A form for readers’ comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation, BWE/H3

P.O. Box 49023

San Jose, CA, 95161-9023

U.S.A.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. 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. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter 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
500 Columbus Avenue
Thornwood, NY 10594
U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling (1) the exchange of information between independently created programs and other programs (including this one) and (2) the mutual use of the information that has been exchanged, should contact:

IBM Corporation
555 Bailey Avenue, W92/H3
P.O. Box 49023
San Jose, CA 95161-9023

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

BookManager	IBM
CICS	IMS
CICS/ESA	IMS/ESA
DB2	MVS
DFSMS	MVS/ESA

Product Names

In this book, the licensed program "DB2 for MVS/ESA" is referred to as "DB2".

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Preface

This book describes the administrative and operational tasks associated with the IMS/ESA 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 their 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, refer to the “How to Learn IMS” section of *General Information* from the Library page 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
- MVS 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. You can read Part 1 to familiarize yourself with how to use DBRC and the DBRC utilities. Then you can use Part 2 as a reference for syntax and usage of the DBRC commands. The appendixes provide additional reference material.

How to Read the Syntax Diagrams

For details see “How to Read the Syntax Diagrams” on page 98.

Change Indicators

Technical changes and additions are indicated in this publication by a vertical bar (|) to the left of the changed or added text. If a figure has changed, a vertical bar appears to the left of the figure caption.

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:
  - *IMS/ESA Master Index and Glossary*
- For more information on upgrading the RECON from previous releases of IMS:
  - *IMS/ESA Release Planning Guide*
- For installation and initialization topics:
  - *IMS/ESA Installation Volume 1: Installation and Verification*
  - *IMS/ESA Installation Volume 2: System Definition and Tailoring*

- For information on database and system administration:

  IMS/ESA Administration Guide: Database Manager

  IMS/ESA Administration Guide: System
- For information on operating procedures:

  IMS/ESA Operations Guide
- For information on system utilities:

  IMS/ESA Utilities Reference: System
- For information on the utilities used in database recovery:

  IMS/ESA Utilities Reference: Database Manager
- For information on the RECON I/O exit:

  IMS/ESA Customization Guide
- For diagnostic information and messages:

  IMS/ESA Diagnosis Guide and Reference

  IMS/ESA Messages and Codes

Summary of Changes

Changes to The Current Edition of This Book for Version 6

This edition, which is in softcopy only, includes technical and editorial changes.

Changes to This Book for Version 6

This book is an addition to the IMS Version 6 library. It contains new and pre-existing information about DBRC processes, functions, commands, and facilities.

New information on the following enhancements is also included:

- Daylight Saving Time (DST) and the Year 2000 support
- Database Image Copy 2
- The New Time Stamp Format
- Shared VSO support

This edition incorporates new technical information for Version 6 as well as editorial changes and technical corrections made to previously published information.

Library Changes for Version 6

The IMS/ESA Version 6 library differs from the IMS/ESA Version 5 library in these major respects:

- *IMS/ESA Common Queue Server Guide and Reference*

  This new book describes the IMS Common Queue Server (CQS).
- *IMS/ESA DBRC Guide and Reference*

  This new book describes all the functions of IMS Database Recovery Control (DBRC).
- The IMS Application Programming summary books (*IMS/ESA Application Programming: Database Manager Summary*, *IMS/ESA Application Programming: Transaction Manager Summary*, and *IMS/ESA Application Programming: EXEC DLI Commands for CICS and IMS Summary*) are no longer included with the IMS library.
- The Softcopy Master Index is not included.
- All information about IRLM 1.5 and data sharing using IRLM 1.5 has been removed from the IMS V6 books. If you use IRLM 1.5, and want to migrate to using IRLM 2.1 and Sysplex data sharing, see *IMS/ESA Release Planning Guide*.
- The chapter that was titled ″Database Control (DBCTL) Interface″ in the *IMS/ESA Customization Guide* has been revised for Open Database Access (ODBA) and moved to ″Appendix A, Using the Database Resource Adapter (DRA)″ in the *IMS/ESA Application Programming: Database Manager*.

Part 1. How to Use DBRC

Chapter 1. DBRC in the IMS Database Recovery Process

This chapter provides an overview of the IMS database recovery process and then describes DBRC's (IMS Database Recovery Control facility) role in that process.

In This Chapter:

DBRC helps you control log and database recovery. It also controls the data sharing environment by allowing (or preventing) access to databases by various IMS subsystems sharing those 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.
3. Back out any incomplete changes.

Figure 1 illustrates a simple database recovery.

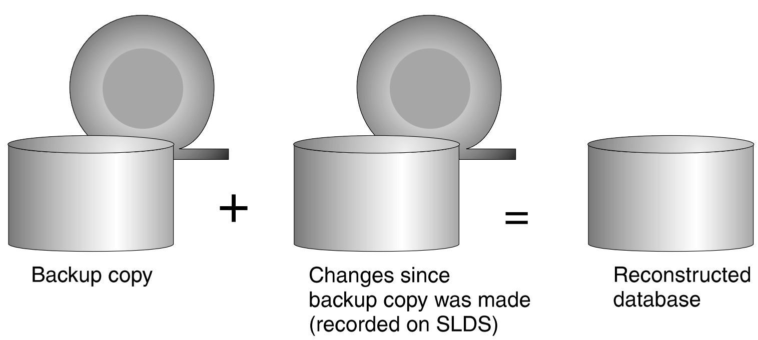


Figure 1. IMS Database Recovery

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) [1]
- Change accumulation data sets

1. system log data set (SLDS), recovery log data set (RLDS)

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

Related Reading: For more information on recovery, refer to *IMS/ESA Operations Guide*.

What is DBRC?

This section discusses tasks related to DBRC, the components of DBRC, and the IMS Recovery utilities.

DBRC Tasks

DBRC is responsible for the following tasks:

1. This first group of tasks are those performed automatically through the interaction of DBRC and IMS (including utilities):
   - Controlling logs for IMS
   - Recording recovery information in the RECON
   - Verifying that database utilities have the correct input
   - Controlling the recovery of registered databases
   - Controlling the data sharing environment by maintaining authorization information for the control and serialization of access to shared databases
2. These tasks are performed when you request them by passing commands to DBRC:
   - Recording recovery information in the RECON
   - Generating JCL for various IMS utilities and generating user-defined output (Use `GENJCL` commands to perform these operations.)
   - Listing the information in the RECONs; use `LIST`commands to list this information

Related Reading:

- For additional information about DBRC's logging support, see "Using DBRC to Record Log Information" on page 12.
- For additional information about DBRC's data sharing support, see "Using DBRC for Data Sharing" on page 19.

DBRC Components

DBRC includes the following components which are discussed in this section:

- RECONs
- Database Recovery Control utility (DSPURX00)
- Skeletal JCL
- RECON Upgrade utility (DSPURU00)

RECONs

DBRC stores recovery-related information in a set of VSAM KSDSs called the RECON (Recovery Control) data sets.

Three RECONs should be defined when you install DBRC. The first two RECONs 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 RECONs as if they were a single data set, the RECON, or simply RECON.

Related Reading: These data sets are described in detail in “Chapter 3. Initializing and Maintaining the RECON” on page 43.

Database Recovery Control Utility (DSPURX00)

This utility provides a set of commands that allow you to specify the type of information that is tracked for your IMS databases.

The DBRC commands allow you to perform all of the following tasks:
- List the information in RECON
- Update the information in RECON
- Use the information in RECON to generate jobs for the IMS utilities

Skeletal JCL

DBRC also uses partitioned data set (PDS) members that contain skeletal JCL.

DBRC uses the skeletal JCL to generate the JCL and control statements that are needed in order to run some of the recovery utilities. These PDS members are distributed with DBRC. You must make any changes to the PDS members that are necessary for your installation’s system configuration.

RECON Upgrade Utility (DSPURU00)

This utility allows you to migrate the RECON from previous releases of IMS to the current release.

Related Reading:
- “Chapter 4. RECON Upgrade Utility (DSPURU00)” on page 63 describes DSPURU00.
- “Chapter 5. Database Recovery Control Utility (DSPURX00)” on page 67 describes DSPURX00.
- “Chapter 11. GENJCL Commands” on page 175 and “Appendix B. Understanding Skeletal JCL” on page 293 contain information on generating and using skeletal JCL.

IMS Recovery Utilities

Recovery utilities is a generic term for the following IMS utilities:

Database Change Accumulation (DFSUCUM0)
Database Image Copy (DFSUDMP0)
Database Image Copy 2 (DFSUDMT0)
Online Database Image Copy (DFSUICP0)
Batch Backout (DFSBBO00)
Database Recovery (DFSURDB0)
Log Archive (DFSUARC0)
Log Recovery (DFSULTR0)

The data sets used by these utilities and recorded in the RECONs are image copy, change accumulation, and log data sets. Other recovery-related information recorded in RECON includes information about:
- use of IMS databases (for example, which IMS subsystems are using which databases at any given time)
- information about recoveries of database data sets (DBDSs)
- information about database reorganizations

Recording Recovery-Related Information

DBRC automatically records information in RECON on the status of log data sets for online subsystems. This includes the OLDS, the SLDS, and the RLDS that are produced by the Log Archive utility. If you specify that DBRC be invoked for batch jobs, DBRC also records the status of the batch SLDS in RECON.

The information recorded for each log data set includes the data set name, volume serial numbers, start and stop times, and status. DBRC records the archiving of OLDS data sets, as well as the creation of the corresponding SLDSs and RLDSs. It also records information about execution of the Log Recovery utility.

If you want DBRC to control database recovery, you must register the databases in RECON. DBRC then records information about when databases were updated 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.

Related Reading: For information about registering databases and data sets, see “Registering Databases and Database Data Sets” on page 12.

You can use the commands of the Database Recovery Control utility (DSPURX00) or their online IMS equivalents (`/RMxxxxxx`) to manually add, delete, or change information in RECON. This utility can also list the contents of RECON, generate JCL, and create a backup copy of the RECON. The utility can process these commands while running either in a batch environment or as a TSO foreground program.

You can provide your own exit routine for DBRC (named DSPCEXT1) which is to be called each time RECON records are changed or read. This exit routine, also called the RECON I/O exit routine, allows you to keep track of changes to RECON in the form of a journal.

Related Reading:

- For an overview of DBRC batch and online command syntax, see “Chapter 7. DBRC Commands” on page 91.
- For detailed information on the exit routine, see *IMS/ESA Customization Guide*.

Generating JCL

DBRC provides PDS members that contain skeletal JCL statements. These PDS members are called skeletal JCL execution members.

Figure 2 on page 9 shows the input DBRC uses to create JCL.

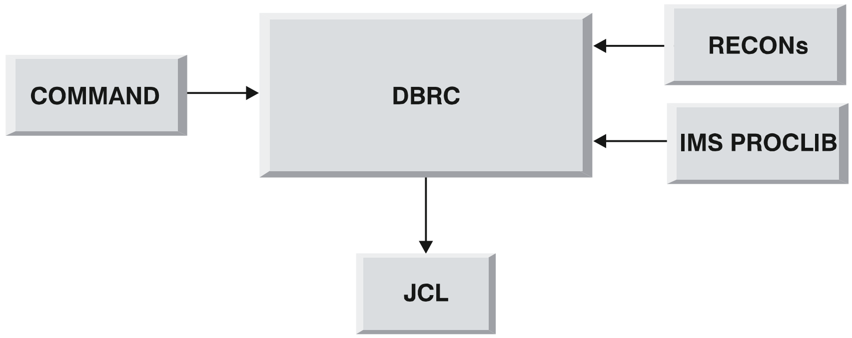


Figure 2. DBRC JCL Generation

DBRC installation procedures place these skeletal JCL execution members from an IMS distribution library (IMS.DFSISRCA) into an IMS procedure library (IMS.PROCLIB). DBRC uses these members to generate jobs (JCL and control statements) for the IMS utilities listed in Table 10 on page 293. 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.

Use the `GENJCL` command to request that DBRC generate JCL in batch or via the `/RMGENJCL` command online. When you enter this command, DBRC reads skeletal JCL and replaces symbolic parameters (performs symbolic substitution) based on the information recorded in RECON 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 information with 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 `GENJCL` process can significantly reduce the time and manual effort required to recover a database. It can also eliminate the causes of many recovery errors. You could spend much time during database recoveries determining which input data sets should be provided in what order to the Database Recovery utility. When change accumulation or RLDS data sets 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.

To receive skeletal JCL, specify PROCLIB=YES on the IMSGEN macro of your installation procedure.

Related Reading:

- For more information about the IMSGEN macro, see *IMS/ESA Installation Volume 2: System Definition and Tailoring*.
- For details about customizing your own skeletal JCL and about the contents of IMS supplied JCL, see “Appendix B. Understanding Skeletal JCL” on page 293.

When Should You Use DBRC?

Most IMS configurations require DBRC, including:

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

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

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-sharing, non-RSR batch environment), you can simplify your recovery process by using DBRC to supervise recovery and protect your databases.

Related Reading:

- *IMS/ESA Operations Guide* provides detailed descriptions of recovery procedures with and without DBRC.
- "Chapter 2. Considerations for a DBRC System" on page 23 further describes how DBRC helps with database recovery.

Initializing DBRC

You initialize DBRC during IMS system generation. This section provides a brief overview of the requirements for creating and executing a DBRC region.

Related Reading:

- For a complete description of IMS installation procedures and requirements, refer to *IMS/ESA Installation Volume 1: Installation and Verification* and *IMS/ESA Installation Volume 2: System Definition and Tailoring*.
- For information on creating and allocating the RECON, see "Chapter 3. Initializing and Maintaining the RECON" on page 43.

Specifying When DBRC is to be Used

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.

The IMSCTRL Macro

Use the DBRC= parameter to specify DBRC or no DBRC for all batch jobs. You can also use this parameter to override DBRC=NO on all batch jobs except:

- If DBRC=FORCE is specified as the first parameter for the IMSCTRL macro, DBRC is active for batch and online environments. DBRC=FORCE cannot be overridden except in these instances:
  - A batch backout execution of IMS
  - The execution of the System Log Recovery utility
  - The execution of the IMS Log Archive utility

- If DBRC=NO is specified (or defaulted) for the first parameter DBRC is active for online environments only.
- If DBRC=YES is specified for the first parameter, DBRC is active for both environments unless it is overridden on IMS procedures.

DBRC for an online IMS subsystem resides in its own address space. Use the DBRCNM= parameter to request that IMS create a cataloged DBRC procedure. The IMS control region initiates the DBRC address space with an `MVS START` command.

Copy the DBRC procedure that was created when the system was generated from IMS.PROCLIB to SYS1.PROCLIB.

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.RESLIB.

IMS.PROCLIB Execution-Parameter Members

You can override the DBRC procedure name specified at system generation 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 may be used to override the DBRC procedure name for an online IMS execution.

DBRC Procedure

IMS automatically starts the DBRC procedure with an `MVS START` command during control region initialization. This procedure specifies parameters for the DBRC region.

To include the DBRC procedure during system generation, you must copy the skeletal procedure that IMS generates in 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.

For a complete description of the DBRC procedure and its parameters see *IMS/ESA Installation Volume 2: System Definition and Tailoring*.

Initializing the RECON

Use the Access method services `DEFINE CLUSTER` command to create the RECONs and then use the `INIT.RECON` command to initialize the RECONs as RECONs 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:

- For information on creating the RECON, see “Chapter 3. Initializing and Maintaining the RECON” on page 43.
- For information on registering databases, see “Registering Databases and Database Data Sets”.
- For information on the `INIT.RECON` command, see “Chapter 12. INIT Commands” on page 211.

Registering Databases and Database Data Sets

The RECON must have a DB record for each database that DBRC is to control recovery of.

Use the `INIT.DB` command to register databases in RECON and to define them as recoverable or non-recoverable.

For each database 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).

A utility job for an `INIT.DBDS` command must include a ddname for the IMS.DBDLIB data set that contains an entry for the DBDS for which you are issuing the command (a ddname for the IMS.DBDLIB is not needed for the `INIT.DB` command).

Related Reading: For specifics about the `INIT` commands, see “Chapter 12. INIT Commands” on page 211.

Restrictions on Using DBRC

Be aware of the following general restrictions 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 RECON 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 are returned in RECON.

DBRC Functions

After you have initialized DBRC for your installation, DBRC automatically provides control for IMS log data sets. If you want DBRC to control the recovery of your DBDSs, register them with DBRC. The following sections describe how DBRC tracks and controls logs, database recovery, and data sharing.

Using DBRC to Record Log Information

DBRC automatically records information about all log data sets that are produced by the online IMS subsystem and by log archiving jobs in the RECON. IMS uses this information for database recovery jobs, if databases are registered, and also for IMS restart. DBRC also tracks the archiving requirements of the online data set (OLDS) and, if requested, generates and submits the JCL for archiving jobs.

It is not required that you use DBRC in order to control logs that are produced by batch subsystems. However, for batch jobs that use DBRC, DBRC records all log data sets that are produced by batch jobs and prevents any batch update job from executing if you specify a dummy or null log data set.

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

Changing RECON Log Records: You can use the `CHANGE.PRILOG` or `CHANGE.SECLOG` commands to change the information in the RECON about OLDS, RLDS, or SLDS records. Use these commands to indicate that errors have occurred on the data sets or that volume serial numbers have changed. You do not normally need to use these commands.

Archiving OLDS: Invoke 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 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 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, the output RLDS data sets are recorded in the PRILOG RECON record, and the SLDS data sets in the PRISLD record. If you do not request an RLDS, the same SLDS data sets are 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 secondary-log output is recorded in corresponding SECLOG and SECSLD records.

Attention: Log data sets that are output from IMS batch jobs are recorded in PRILOG / SECLOG records even though they are technically SLDSs.

Invoke the archive utility by entering the `GENJCL.ARCHIVE` command. DBRC then determines which OLDSs are full, and generates the appropriate JCL.

Related Reading: See Member DFSVSMxx in the *IMS/ESA Installation Volume 2: System Definition and Tailoring* for more information on the ARCHDEF statement and automatic archiving.

Whether you use automatic archiving or invoke archiving yourself, you should 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 OLDSs are used, 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 RECON 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 RECON as having errors.

DBRC Log Related Commands: You may use the following commands, which perform DBRC log-related functions, in your operational procedures:

BACKUP.RECON	**DELETE.LOG**	**LIST.LOG**
CHANGE.PRILOG	**GENJCL.ARCHIVE**	**NOTIFY.PRILOG**
CHANGE.RECON	**GENJCL.CLOSE**	**NOTIFY.SECLOG**
CHANGE.SECLOG	**INIT.RECON**	

Related Reading: Refer to the command chapters in Part 2 for descriptions of the commands.

Using DBRC for Database Recovery

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

DBRC also records:

- Database image copies
- Reorganizations (except DEDB online reorganizations)
- Recoveries
- Change accumulations
- Backout

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

Figure 3 shows where DBRC generally fits in utility execution.

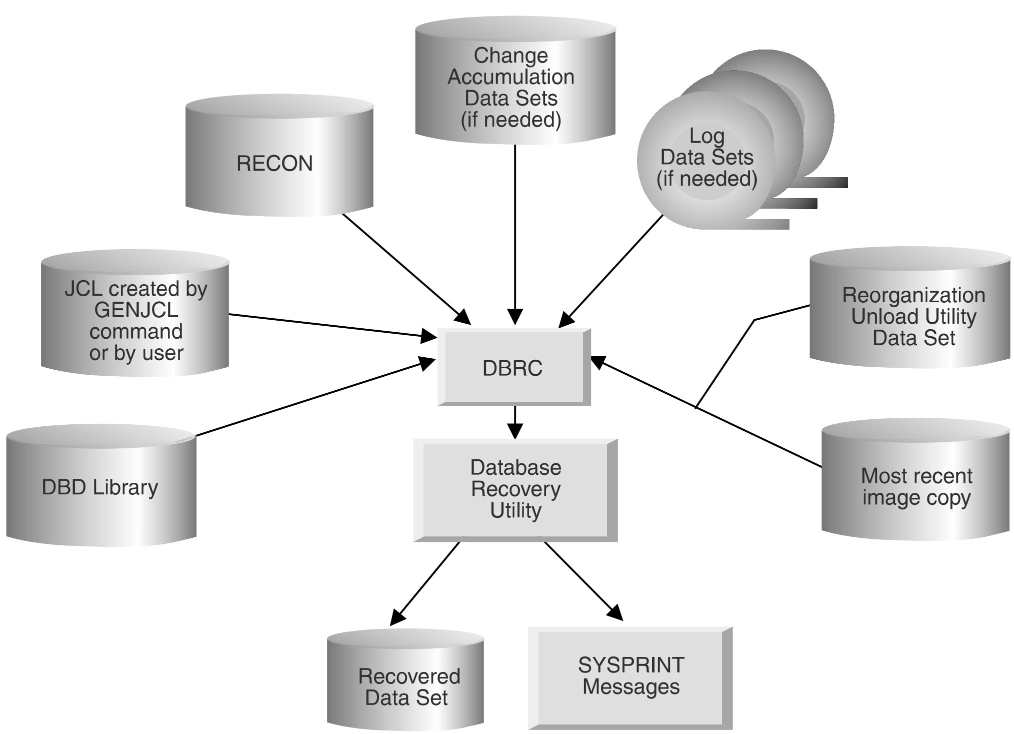


Figure 3. DBRC's Role in Utility Execution

Implement DBRC in phases, defining at first only a few recoverable databases in RECON. 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.

Using the IMS Recovery Utilities: DBRC is invoked by the following IMS utilities to validate input to the utility and record the results of the execution of the utility:

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)
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)

Figure 4 shows where DBRC specifically fits in the input scheme.

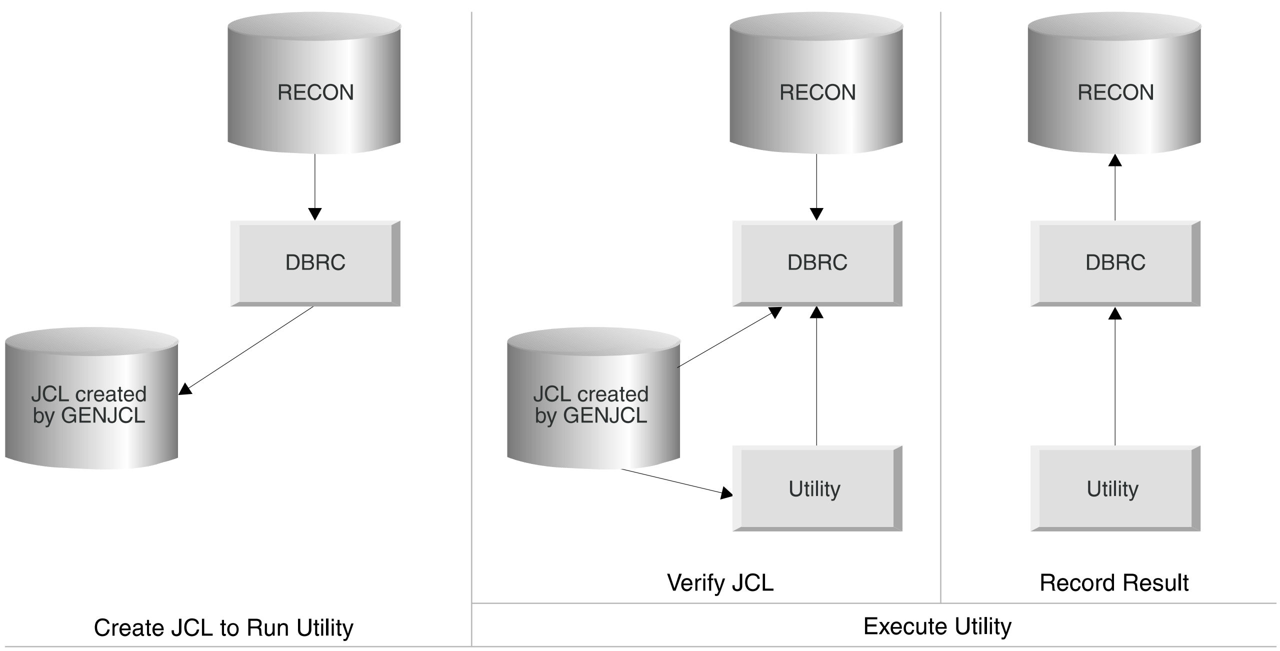


Figure 4. DBRC’s Specific Place in a Utility Input Scheme.

DBRC checks that these utilities perform the correct processing with the correct input data sets. This is true even if you do not use the `GENJCL` command. DBRC verifies the JCL for these utilities before execution.

Exception: For the HD and the HISAM reorganization utilities, DBRC only records their execution in RECON.

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.

The DEDB Area Data Set Create utility increases availability by providing additional usable copies of an online area. It does not provide backup copies for recovery. The DEDB Area Data Set Create utility uses the DBRC RECON data set as part of its input.

DBRC verifies all logs that are input to the database Batch Backout utility (DFSBBO00) by determining 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.

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.

Related Reading:

- For more information on the IMS recovery utilities, see *IMS/ESA Utilities Reference: Database Manager*.
- for information about a related command, which manually creates a backout record in RECON, see“NOTIFY.BKOUT” on page 248.

Specifying Image Copy Requirements: 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.

If you use nonstandard image copy techniques, such as a pack dump, you must create your own JCL and update RECON with a `NOTIFY.UIC` command.

When you register a DBDS in RECON, 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, you 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, you specify the optional parameter RECOVPD. This specifies 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 increases the GENMAX value.

DBRC also provides an optional data set recycling capability for standard image copies. You can, for example, pre-allocate four DASD data sets to contain the image copy, and request that DBRC reuse these data sets. The recycling capability is indicated using the REUSE keyword of the `INIT.DBDS` command. DBRC then generates JCL so that the oldest DASD data set is always used for output from 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.

Concurrent Image Copy: The concurrent image copy process requires that a database be registered with DBRC. Information about a concurrent image copy data set is recorded in the RECON data set. When concurrent image copy succeeds, a true image exists and can be used for recovery.

You cannot take concurrent image copies of a nonrecoverable databases because changes to them are not logged, and the fuzzy (incomplete) copy remains fuzzy if the database is recovered.

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.

Database Image Copy 2: The Database Image Copy 2 utility requires that databases and areas be registered with DBRC. You specify whether the created image copies are to be concurrent image copies. You cannot take concurrent image copies of nonrecoverable databases.

Specifying Change Accumulation Requirements: If you decide to use the Database Change Accumulation utility for some or all of your recoverable registered databases, specify the change accumulation groups using the `INIT.CAGRP` command. You can, for example, divide the DBDSs into the following groups:

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

As with image copies, DBRC provides an optional data set recycling capability for change accumulations. If you decide to use this capability, specify the data set information using the `INIT.CA` command.

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: When moving a member to a new CA group, it may be beneficial to do so shortly after an image copy is done. The change accumulation utility processes all logs for the member that are needed for recovery, based on the last good image copy.

Using 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.RECEIVE
GENJCL.USER
LIST.HISTORY
GENJCL.OIC
GENJCL.RECOV
LIST.DBDS

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 timestamp 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 RECON. A DBDS can belong to more than one DBDS group.

A DBDS group can be used like a DBDS group for the `GENJCL` and `LIST` commands. So there is no need to define a DBDS group consisting of the DBDSs or areas of a single database. Specify the database name and omit the DDNAME 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

Using 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`, and `/DBRECOVERY` commands instead of issuing these commands separately for each database or area. This greatly reduces the number of times these commands must be issued. Use the DATAGROUP keyword to specify the DB group name.

You can define as many DB groups as you want. Up to 2000 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 RECON.

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 RECON using the DBDS group record. The following commands affect the definition of a DB group:

INIT.DBDSGRP
CHANGE.DBDSGRP
DELETE.DBDSGRP
LIST.DBDSGRP

Opening a Database: After IMS opens a database, DBRC passes back the RECON 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.

Related Reading: For information on EEQEs, see *IMS/ESA Operations Guide*.

Recording Allocations and De-allocations: DBRC records information in RECON about changes to DBDSs and areas. DBRC subsequently uses this information to determine what log data sets might contain change records for a given DBDS or area.

When a DBDS that is registered in RECON is first updated, 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. This 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 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 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 that was created when the DBDS or area was allocated. If no de-allocation time is recorded, DBRC uses the closing time of the associated log as the de-allocation time. Thus RECON 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 them.

Using DBRC for 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: For more information on data sharing, see the *IMS/ESA Operations Guide* and the *IMS/ESA Administration Guide: System*.

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).

Sharing Information Recorded in RECON: DBRC records the following data sharing information in RECON 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 using the `INIT.DB` or `CHANGE.DB` commands.

The following sharing levels are defined using the `INIT.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 MVS. Sharing requires that IMS subsystems sharing a database use the same RECON. 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 IRLMs. Multiple IMS systems can be authorized for non-exclusive access. The IMSs can be on multiple MVSs using different IRLMs.

Data Set Naming Conventions

The use of MVS or VSAM catalog facilities for image copy and change accumulation data sets is optional, because DBRC always maintains volume serial information pertaining to these data sets in RECON. 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 RECON. DBRC assumes that data set names specified in quotation marks do not follow the name convention. Therefore, DBRC does not check data set names surrounded by quotation marks.

When you add records to RECON 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 RECON.

Naming Convention for Image Copy Data Sets

The format for image copy data sets is:

*high-level-qualifier.dbdname.ddname.*IC.*low-level-qualifier* where:

- *high-level-qualifier* is a the data character string of your choice from one to eight alphanumeric characters long. The first character must be alphabetic.
- *dbdname* is the database name of the DBDS for which the image copy data set is being recorded in RECON.
- *ddname* is the data set ddname of the DBDS for which the image copy data set is being recorded in RECON.
- IC indicates that this is the image copy data set.
- *low-level-qualifier* is a character string of your choice from one to eight 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 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*

where:

- *high-level-qualifier* is a character string of your choice that can be from one to eight alphanumeric characters long. The first character must be alphabetic.
- *cagrpname* is 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* is a character string of your choice that can be from one to eight alphanumeric characters long and must be unique for each CA group. The first character must be alphabetic.

DBRC Support for Remote Site Recovery

DBRC assists you in the installation of IMS DB and IMS TM, as well as with the definition and management of IMS components in the Remote Site Recovery (RSR) complex. In support of RSR, DBRC provides:

- Commands to define, update, and display the status of the RSR complex.

  The RECON contains the definition of an RSR complex. You define the elements of the RSR complex with DBRC commands, and you can modify and display the RSR complex definition with other DBRC commands.

  Related Reading: For more information on RSR, see *IMS/ESA Operations Guide*.
- 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. And 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 of the registered databases at the new active site to indicate that they are now the master databases.

Related Reading: See *IMS/ESA Administration Guide: System* for more information on controlling database recovery.

Chapter 2. 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

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: It is advisable to make a backup copy of the database after you initially load it. In addition, a new backup copy can be made 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 offers you several ways of making backup copies:

The Database Image Copy utility (DFSUDMP0) runs offline and uses access method services to make the copy.

The Database Image Copy 2 utility, which runs offline and invokes DFSMS Concurrent Copy to make the copy.

The Online Database Image Copy utility (DFSUICP0) runs online and uses access method 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 (dependent on installation parameters) call DBRC to update essential information in the RECON. 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 30 for a discussion of how to notify DBRC about these data sets.

Backup Considerations

Related Reading:
- See *IMS/ESA Administration Guide: Database Manager* for more information on HSSP processing of DEDB areas.
- See “Controlling the Number of Image Copies Managed” on page 27 for information on the utilities to make your backup copies.

The Image Copy Utilities (DFSUDMP0, DFSUICP0, DFSUDMT0)

IMS allows you to “take a snapshot” of your database before and after changes have been made to the database. The “snapshots” 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 the image copy utilities operate on data sets or areas, so if a database is composed of multiple data sets or areas, supply the utility with multiple specifications. You can request that one of the supported image copy utilities produce an image copy data set and a duplicate image copy data set in one run of the utility.

Recommendation: It is advisable to 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. 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 26 for more information.

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 RECON. 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), and it calls DBRC to record information in the RECON about the image copy data sets that it creates. An image copy record in RECON 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. There are two different commands to create image copy jobs:

`GENJCL.IC` for the offline utilities Database Image Copy and Database Image Copy 2 and GENJCL.ICGENJCL.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. Since the batch processing is not guaranteed to be physically repeatable, the database could be damaged by the combination.

Database Image Copy (DFSUDMP0)

This utility copies data sets for HISAM, HIDAM, and HDAM databases, and areas for DEDBs. It runs offline and supports a concurrent image copy (CIC) option that allows you to take an image copy while the database remains online.

When you run the Database Image Copy utility to take a consistent image copy (CIC option not specified), the use of 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.

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.

Using Database Image Copy, concurrent image copies of OSAM data sets and VSAM Entry Sequenced Data Sets (ESDSs) can be taken. VSAM Key Sequenced Data Set (KSDSs) are not supported for concurrent image copy by this utility.

Database Image Copy 2 (DFSUDMT0)

The Database Image Copy 2 utility (DFSUDMT0) uses DFSMS Concurrent Copy to take a image copy. It invokes the DFSMS to dump the input data set using the Concurrent Copy option. The output is recorded in the RECON 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, and HDAM 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.

The Database Image Copy 2 utility provides greater database availability when taking consistent or ’clean’ 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 than 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 ’fuzzy’ copy or 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.

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 primary and secondary copies are recorded in the RECON.

Backup Considerations

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 the Database Recovery utility. The image copies are recorded in RECON with image copy type SMSCIC or SMSNOCIC. The Database Recovery utility must use DBRC when recovering with these types of image copy data sets.

Online Database Image Copy (DFSUICP0)

This utility runs as a BMP program in the online IMS and DBCTL environments. You can use it for HISAM, HIDAM, and HDAM 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.

Related Reading: See the *IMS/ESA Utilities Reference: Database Manager* for more information about specifying the CIC option on the EXEC statement.

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 or 'clean' 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.

You can only take a concurrent image copy of a database that is registered with DBRC. Also, the Database Image Copy utility (which can take consistent image copies without using DBRC) must use DBRC when taking a fuzzy copy. You cannot take a concurrent image copy of a non-recoverable database. Fuzzy copying of non-recoverable databases is not allowed because there is no log data to 'complete' the image copy.

Creating Image Copy Data Sets for Future Use and Reuse

You can use the `INIT.DBDS` and the `CHANGE.DBDS` commands to define a ″pool″ of data sets to receive output from the image copy utilities for the DBDS or area.

Use the REUSE parameter to inform DBRC that you want to be able to define image copy data sets and record them in RECON 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 RECON 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 RECON 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:

- For more information about the RECOVPD parameter, see “Recovery Period of Image Copy Data Sets and GENMAX”.
- For more information on the REUSE, NOREUSE, GENMAX, and RECOVPD see “INIT.DBDS” on page 216.

Recovery Period of Image Copy Data Sets and GENMAX

A `recovery period` is the minimum amount of time that image copy information is maintained in RECON. For example, if the recovery period of a DBDS or DEDB area is 14 days, image copies are maintained for at least 14 days.

Backup Considerations

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. GENMAX increased (DSP0065I)
=GENMAX	=GENMAX	>RECOVPD	Oldest IC DS reused

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
=GENMAX	<RECOVPD	No delete (DSP0064I). GENMAX increased
<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 RECON. 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.

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, `GENJCL.IC` for a DBDS defined as REUSE generates JCL only one or two output copies (because you can define image copy data sets for only one or two copies). If you use `GENJCL.IC` for the Database Image Copy 2 utility to process a DBDS defined as REUSE and you want more than two output copies, you have to modify the generated JCL before you execute the job.

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), as shown in Figure 5. 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.

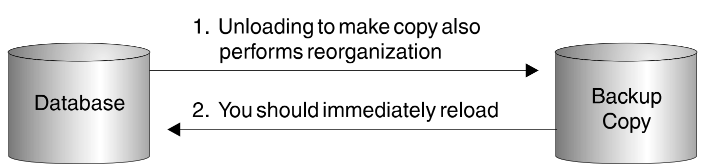


Figure 5. Making a Backup Copy with HISAM Unload

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 RECON 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.

Backup Considerations

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 RECON 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 RECON 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 RECON automatically; use a `NOTIFY.UIC` command to inform RECON of these data sets. If this information is not recorded in RECON, 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 RECON (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).

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

Frequency and Retention

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 timestamp recovery, make a backup copy for use in subsequent recoveries.

Log Record Change Accumulation

This section describes the use of the Database Change Accumulation utility.

Using Log Data Sets for Recovery

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.

DBRC 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 DBRC 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 6 on page 32 illustrates how change accumulation merges data from divergent data streams.

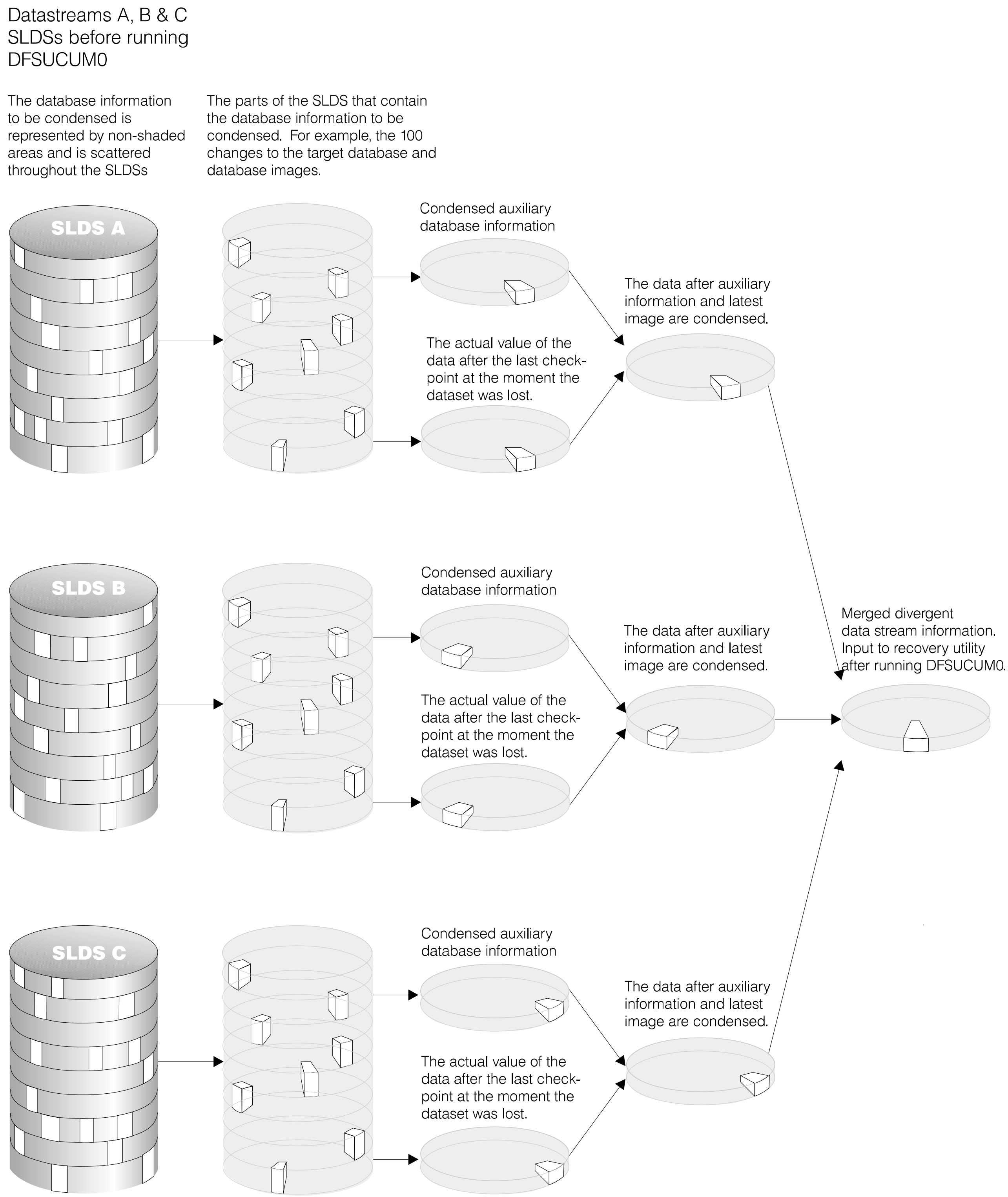


Figure 6. What Change Accumulation Does with Data from Divergent Data Streams

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 DBRC).

Figure 7 illustrates why 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 DBRC.

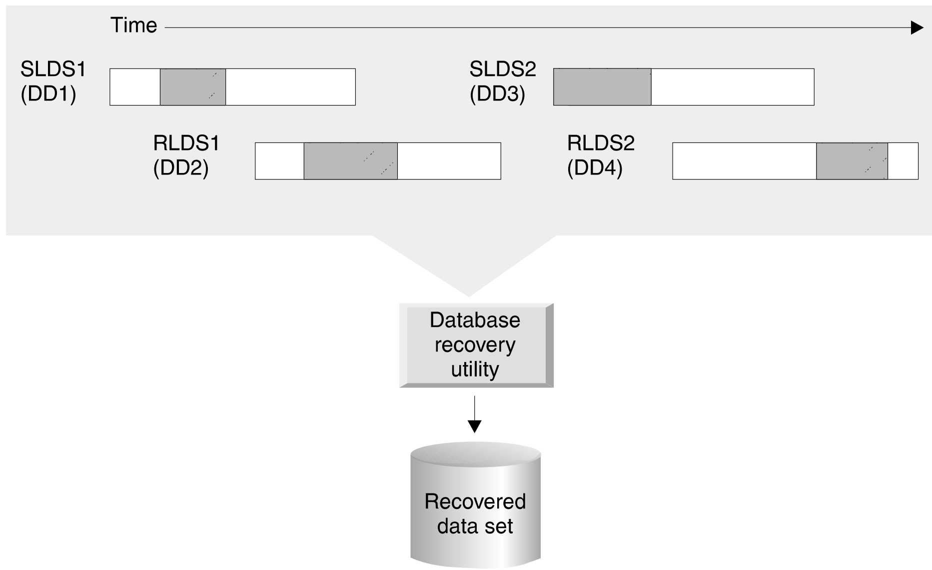


Figure 7. Non-Concurrent Data Set Update Information in Logs: Change Accumulation not required

Figure 8 on page 34 illustrates why change accumulation can be required when concurrent data set update information exists in various logs. The logs cannot be input to DBRC in a way that the change records are seen in the correct order.

Log Record Change Accumulation

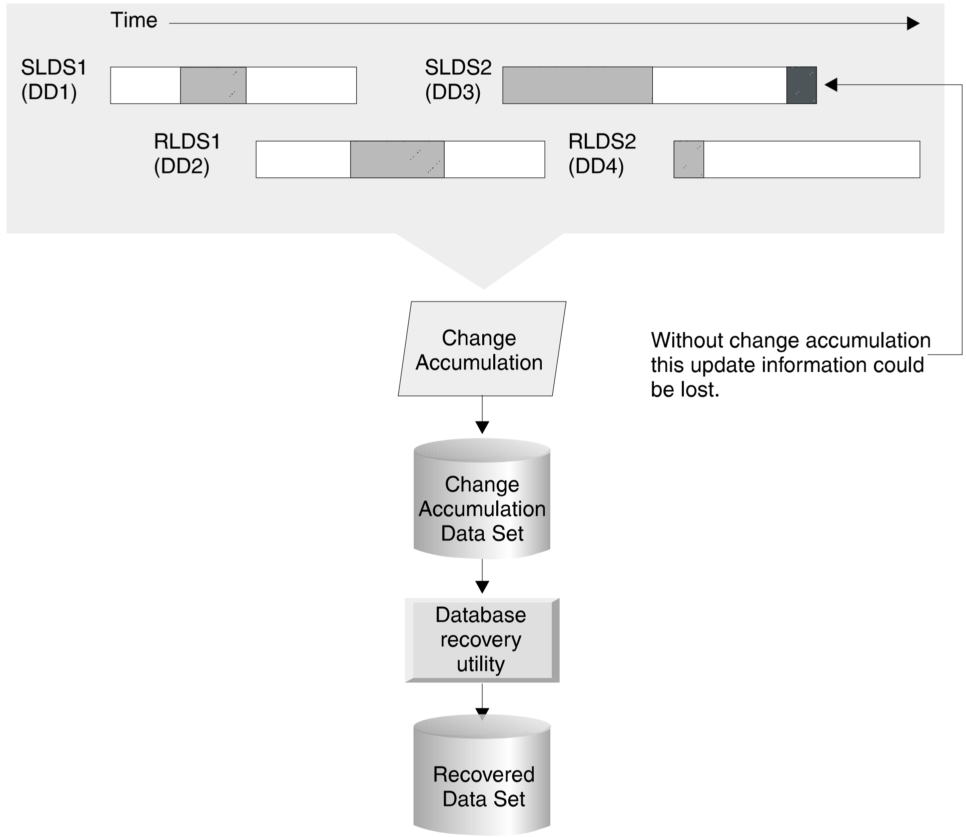


Figure 8. Concurrent or Overlapping Data Set Update Information in Logs: Change Accumulation Required

Related Reading: For detailed instructions on running the Database Change Accumulation utility, see the *IMS/ESA Utilities Reference: Database Manager*.

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 DBRC 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 RECON 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 timestamp 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 during change accumulation to reduce data to a minimum.

Related Reading:

- Sample operating procedure 162 in *IMS/ESA Sample Operating Procedures*. describes the task of running the Database Change Accumulation utility.
- For more information on the change accumulation process in general, see the *IMS/ESA Operations Guide*.

Change Accumulation Groups

A CA group can contain from one to 2000 members, which must be registered DBDSs or areas. Use `INIT.CAGRP` to create your CA group. To modify your CA group, use `CHANGE.CAGRP`. A DBDS or area can belong to only one CA group.

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

You can run the CA utility for DBDSs that do not belong to a CA 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 CA group with the REUSE parameter and also use a `GENJCL.CA` command to generate the job for the Database Change Accumulation utility, reuse can occur. When all these available change accumulation data sets for this CA 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.

Log Record Change Accumulation

If you define the CA 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 CA utility and the number of data sets specified by GRPMAX has been reached, DBRC deletes the RECON record of the change accumulation 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: For details on the Change Accumulation JCL specifications, see "Database Change Accumulation Utility (CAJCL)" on page 317 and "Database Change Accumulation Utility JCL (CAJCL)" on page 329.

Recovery

This section describes various aspects of database recovery, including methods of recovering databases with DBRC.

The Database Recovery Process

Recovery of a database data set by DBRC, DFSURDB0, consists of two steps:

1. Loading (usually) the most recent image copy.

   If the image copy is a UIC, the utility does not load the most recent image copy, so you must perform this step yourself.
2. Applying all changes logged since the ″purge time″, which is the point-in-time selected to start the Change Accumulation utility.

The changes can be recorded in log data sets, in change accumulation data sets, or in a combination of the two.

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.

If the database was accessed for update at the time the copy was made, the image copy is said to be ″fuzzy″; that is, changes already made to the database by active applications might be missing from the copy because they had not yet been physically written to the data set. They had, however, been written to the log. In this case, to ensure that all changes are applied it is necessary to go back to some earlier point in the logs, how far depends on the type of database and which image copy utility was used.

Note that the input to the Change Accumulation utility, when the most recent image copy is ″fuzzy″, is subject to the same conditions. The point-in-time selected to start the Change Accumulation utility is called the ″purge time.″

You can omit all logged changes after a certain time from the input. This is called a ″timestamp recovery″, and is equivalent to backing-out the omitted changes from the database. However, if you want to omit changes, it is impossible to do so correctly without DBRC.

How DBRC Helps in Recovery

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 RECON, DBRC selects the image copy data set to use for step 1, and selects the CA and log data sets that are to be input to Step 2. 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 command fails.

You can request a timestamp recovery, and DBRC selects the correct logs and, at execution time, communicate to the utility where to stop processing the input to correctly perform your request.

Validating Utility JCL

When DBRC runs against a registered database data set, it presents its inputs to DBRC for validation. Whether the recovery JCL was created by you or by DBRC, DBRC verifies that the input for both Step 1 and Step 2 are correct according to the current RECON information. (It is possible, even if you created the JCL with the `GENJCL` command, that intervening events could invalidate it before it runs.)

Recording the Result

Upon successful conclusion of the recovery, DBRC records it in RECON, including the range of omitted changes if you did a timestamp recovery. To ensure that those changes stay backed-out, you must ″lock in″ the range of omitted changes by either updating the database data set or by taking an immediate image copy. DBRC then prevents the changes from being reapplied on any subsequent recovery.

Planning Recovery Procedures

Recovery in a data sharing environment is similar to standard IMS recovery. It involves you in these primary tasks:

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

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.

Database reconstruction (forward recovery), program-level (dynamic) backout, database backout, and restart (with embedded recovery) in a data sharing environment, however, are different in a nonsharing environment.

Related Reading: For detailed information on recovery procedures in sharing and nonsharing environments, see the *IMS/ESA Operations Guide*.

Recovery Facilities

The following recovery facilities behave differently in data sharing and nonsharing environments:

- Dynamic backout

- Batch backout
- Forward recovery

Dynamic Backout

An online IMS subsystem in a data sharing environment 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.

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

Related Reading: For detailed information about dynamic backout, see the *IMS/ESA Operations Guide*.

Batch 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 (BKOUT) 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 executing an emergency restart with the COLDSYS parameter (`/ERESTART` COLDSYS), run batch backout with either the COLDSTART or ACTIVE parameter. This creates a backout record (BKOUT) for all in-flight and indoubt UORs as candidates for backout. The next IMS restart promotes candidate UORs to backout-needed status. Backout-needed flags are set in database records in RECON.

When dynamic backout fails, a backout record is created with a UOR indicating dynamic backout failure and the database record in RECON is flagged as needing backout.

When the databases are backed out successfully, the flags in database records in RECON are reset appropriately and the backout records are updated. When backouts for all of the UORs have been completed, the backout record is deleted from RECON.

For DB control if you choose not to back out an unresolved indoubt UOR, use the `DELETE.BKOUT` or `CHANGE.BKOUT` command to remove it from the RECON backout record.

DBRC commands are also available to update backout records if needed. The need to make manual changes to the backout record should be minimal. If timestamp recoveries are run instead of batch backout, then the backout record in RECON need to be updated if one exists, using DBRC commands.

Recommendation: Use the `DELETE.BKOUT` or `CHANGE.BKOUT` commands with extreme caution.

DBRC provides batch backout input log verification. Log volumes input to the batch backout utility are checked to ensure they are in the correct sequence, all logs are provided, and properly closed. When ACTIVE or COLDSTART options are included in the utility SYSIN statement, then an additional check is done to ensure all volumes related to restart are included. For DL/I batch logs, a check is also done 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 input log verification and notification of UORs is not done. Existing UORs are updated on successful completion of backouts.

Related Reading: For more information on UOR and backout, see *IMS/ESA Operations Guide* and *IMS/ESA Utilities Reference: Database Manager*.

Limitations of DBRC Backout Support

Although this support protects databases from damage caused by the most common errors in using backout, some possible sources of damage to databases remain.

- If an IMS subsystem abends and an ERE COLDSYS is done 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 statement (or the ACTIVE statement).
- Including logs from multiple runs of a batch job in the same log data set (by specifying DISP=MOD) makes log verification unreliable.
- DBRC commands are provided for the modification of the RECON. Carelessness in using these commands could lead to errors which would allow backouts that should not be done or allow other subsystems access to databases that are in need of back out.
- The Batch Backout control statement BYPASS LOGVER is provided to allow backouts to be done when RECON information indicates that the input log is invalid or the backouts are not needed. Careless use of this control statement can cause backouts to be performed that should not be.
- 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 timestamp recovery was performed (to recover the database to a point in time prior to this log) makes log verification unreliable. Since the log being backed out is the last log for this SSID, it passes DBRC verification.

Forward Recovery

The process of recovering a database in a data sharing environment has certain similarities to recovering a database in a nonsharing environment. In both environments, use DBRC (DFSURDB0) and provide it the most recent image copy of the lost database, and all pertinent system log data sets used since that copy was made.

Block-level data sharing, however, might require the additional step of Change Accumulation.

Related Reading: For further discussion of forward recovery, see the *IMS/ESA Operations Guide*.

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: For detailed information about recovery without DBRC and its related commands, see the *IMS/ESA Operations Guide*.

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 IMS Failure

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

- If you can shut down the subsystem normally (a checkpoint shutdown /CHECKPOINT PURGE, /CHECKPOINT DUMPQ, or /CHECKPOINT FREEZE), you can restart it normally, using the /NRESTART command.
- If the subsystem failed, an emergency restart must be done using the /ERESTART command.

In a data sharing environment, however, consider that if the associated IRLM also stops or fails, you must first start IRLM before you start IMS in a block-level environment.

Related Reading: For more information on restart after IMS failure, see the *IMS/ESA Operations Guide*.

Restart after DBRC Failure

Because DBRC runs under the control of IMS, IMS knows whether its DBRC fails. 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 using the /ERESTART command.

Recovery Involving IRLM Configurations

DBRC records IRLM status information in the subsystem record.

Related Reading: For information on the IRLM status information that DBRC records in the subsystem record, see “Appendix C. Sample Listings of RECONs” on page 347 and *IMS/ESA Operations Guide*.

Batch Backout

Prior to executing an emergency restart with the COLDSYS parameter (/ERESTART COLDSYS), run batch backout 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 RECON backout record.

Recommendation: Use the DELETE.BKOUT command with extreme caution. It deletes all backout information for a subsystem from the RECONs`.

Archiving Log Records

Related Reading:

- For more information on automatic, manual and custom archiving of log records, see *IMS/ESA Operations Guide*.
- For more information about specifying entry points and running the Log Archive utility, see *IMS/ESA Utilities Reference: System*.
- For more information about writing exit routines, refer to *IMS/ESA Customization Guide* .

Chapter 3. Initializing and Maintaining the RECON

In This Chapter:

DBRC records recovery-related information in a pair of key-sequenced data sets (KSDSs) called RECON (recovery control). DBRC uses dual recovery control (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 RECONs and an error occurs on one of them during operation, the current jobs continue using the remaining one. New jobs start only if your RECON setting allows a start with only one active RECON. If you want to continue operations in dual mode, you can define a third RECON (RECON3). DBRC does not use this spare data set unless an error occurs on one of the two active RECONs. 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 normal three RECON operating configuration.

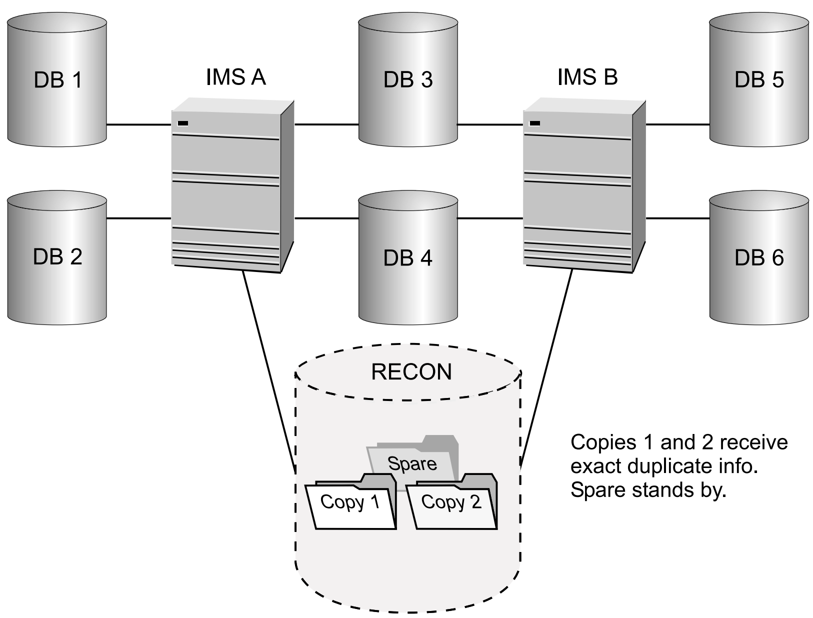


Figure 9. RECON Three-Data-Set Scheme

Planning Considerations for the RECON

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

Attention: The key size was changed for Version 6 from 24-bytes to 32-bytes.

Maintaining the RECON

Dynamically allocate your RECONs specifying VSAM SHAREOPTION(3,3). Ideally, each data set should be on a different device, channel, control unit, and catalog. Data sets should also be of different sizes (see "Avoiding RECON Space Problems" on page 48).

When defining the RECONs using VSAM Access Method Services (AMS), you must use the same index control interval (CI) size and data CI size for all the RECONs. Unequal index or data CI sizes cause DBRC to use virtual storage inefficiently.

Recommendation:

- Ensure that the smallest data CI size specified exceeds the largest specified index CI size by at least 2048 bytes. Failure to do so can seriously degrade your DBRC performance.
- The maximum record size for all RECONs must be specified when defining the RECONs using AMS. DBRC initialization fails if the maximum record size of the two active RECONs do not match.

First-Time Users

After allocating the RECONs, use the `INIT.RECON` command to initialize the RECON header and control records. This command is only valid for an empty, uninitialized RECON. If the initialization job fails before it has completed successfully, delete and redefine the data sets before rerunning it.

Avoiding RECON Contention Problems

Maximum RECON availability is dependent on two things:

- Eliminating deadlock
- Minimizing situations in which more than one RECON becomes simultaneously unavailable

For maximum availability, each RECON must:

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

For additional information on avoiding space problems, see "Avoiding RECON Space Problems" on page 48.

Recommendation:

To eliminate contention and facilitate recoverability it is recommended that RECONs be the only type of data set on their respective devices.

To eliminate deadlocks, the RECONs must:

- Be the only objects cataloged in their respective catalogs.
- Be on the same device as their catalogs. When RECONs are accessed, an enqueue on the RECONs can result, followed by an enqueue on the catalog. When the RECONs and catalog are on the same device, the possibility of conflicts with another job enqueueing the devices in reverse order is eliminated.

Avoiding RECON Deadlock

To avoid deadlock situations, give special consideration to the placement of RECONs 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 RECONs. So, during the use of DBRC, two RECONs are reserved most of the time. DBRC always reserves both RECONs 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.

Related Reading: The *MVS/ESA System Programming Library: Application Development Guide* explains deadlock situations and volume assignments further; refer to it when placing your RECONs.

RECON Serialization

Access to the RECONs must be controlled to avoid deadlock. The following macros are used to control access to the RECONs: RESERVE, GRS, OBTAIN, DEQ. DFP Record Management Services are also discussed and RECON serialization strategies are presented.

RESERVE: DBRC issues the MVS RESERVE macro to serialize access to each RECON data set. DBRC keeps the RECONs reserved until it completes its processing; the more RECON records it must examine or change, the longer it holds the RECONs. 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.

GRS: Global Resource Serialization (GRS) 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.

For more information about GRS and the MVS RESERVE, DEQ, and ENQ macros, see the following publications:
- OS/390 MVS Programming: Authorized Assembler Services Reference
- OS/390 MVS Planning: Global Resource Serialization

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 reserved in a multi-system environment. For more information about the OBTAIN DADSM macro, see MVS/DFP V3R3 System Program Reference Manual.

DEQ: DBRC releases the RECONs by using the MVS DEQ macro.

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

RECON Serialization Strategies: The following items discuss several serialization implementations and their effects.
- GRS SYSTEMS EXCLUSION RNL

  If you implement GRS SYSTEMS EXCLUSION RNL, then GRS does not do global serialization, and the RESERVE is issued. This is the recommended way to implement IMS with DBRC and it works well provided that the RECON data set is located on a DASD volume which does not contain other data sets which are needed by other MVS systems.

  To implement this method follow these steps:
  1. Add the RECON QNAME to the SYSTEMS EXCLUSION RNL
     – RNLDEF RNL (EXCL) TYPE (GENERIC) QNAME (DSPURIO1)
  2. Carefully consider the placement of the following VSAM QNAMEs
     – SYSZVVDS
     – SYSIGGV2

  The performance (in terms of least CPU time used, least storage used, and least elapsed time) is best for this option. It does require careful control of the contents of the DASD volume holding the RECON (ideally a dedicated volume exists for the RECON).
- RESERVE CONVERSION RNL

  If you implement GRS RNL CONVERSION (by adding the QNAME for the RECON, DSPURI01, to the conversion list), the hardware reserve is eliminated and replaced by a GRS enqueue that is propagated to all sharing MVS systems (they must all be in the same GRS ring). This locks out access to the RECON data set by IMS code that does the enqueue on the RECON). It does not lock out access by non-IMS code that does not issue an ENQUEUE for DSPURI01.

Thus, there is a potential integrity exposure, but only for non-IMS products, that should NOT access the RECON in any case).

Recommendation: An example of a non-IMS job that may access the RECON (invalidly) is an IDCAMS REPRO, that takes a ″backup″ of the RECON data set while it is active and in use by IMS. For case #1 (exclusion), the REPRO waits on the RESERVE until the RECON is idle (and then locks DBRC out from the RECON until the REPRO completes). In case #2 the REPRO runs ″successfully″ because it does not ENQUEUE on the DSPURI01 RNL (and therefore is not aware that the RECON is in use). This is one reason that GRS reserve conversion is not recommended without careful control of the operating environment.

Other data sets on the same DASD volume can be used while the RECON is ″reserved″; this is the benefit of performing the RNL conversion. GRS RNL conversion uses CPU and storage and can affect system performance. Conversion of a production IMS system is not recommended. It may be reasonable to do for a development IMS system, but the performance analysis is a decision for each specific environment.

- Do nothing.

  If you do nothing, you get the worst of both worlds (case #1 and #2 above). If neither GRS RNL exclusion or conversion is specified, the system issues the RESERVE and also propagates the ENQUEUE. You get the performance cost of #2, and the reserve of #1. This should be avoided.

Related Reading: There are a number of possible solutions to contention or performance problems associated with the RECON data sets. For information on RECON contention and performance problems see the Informational APAR II10915.

For information on performance and tuning diagnostics, see the Informational APAR II10735.

Allocating the RECONs

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, 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.RESLIB. When multiple processors access the same RECONs, keep the IMS.RESLIB information pertaining to dynamic allocation parameters in synchronization on all processors.

Related Reading: For information about the DFSMDA macro, see *IMS/ESA Utilities Reference: System*.

DBRC always allocates RECONs with DISP=SHR.

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

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

The principal advantages of dynamic RECON allocation 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 RECONs, in case of error, without having to shut down online IMS systems.

Avoiding RECON Space Problems

Allocate the RECONs with different amounts of space so that if one becomes full, the system can continue using the other RECON while you provide a replacement.

If one RECON becomes full during online operation, IMS deallocates it. DBRC responds by copying and reorganizing the good RECON to a spare RECON, if one is available. If no spare RECON 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 VSAM AMS. If you are in single mode, and a spare RECON is available, the next time DBRC accesses the RECON, 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

The following list gives the DEFINE CLUSTER keywords that are recommended when defining a RECON VSAM KSDS:

CONTROLINTERVALSIZE

The values used with this keyword affect the total amount of storage used by DBRC for VSAM buffers. DBRC uses the Local Shared Resources (LSR) option of VSAM to process the RECONs. If the number of index and data buffers created by DBRC is allowed to default, the amount of storage used for RECON buffers is (6·index_ci_size + 12·data_ci_size). This occurs when the index and data CI sizes are the same for all RECONs. The amount of storage used by DBRC for buffer space can be adjusted by changing the index or data control interval (CI) size on this keyword.

Recommendations: Initially set your minimum CI size to a minimum of eight KB. Also, ensure that the smallest data CI size specified exceeds the largest index CI size specified by at least 2048 bytes. Failure to do so can seriously degrade your DBRC performance. Alternatively, you can change the default number of index or data buffers used by DBRC in an online or batch environment.

Related Reading: See *IMS/ESA Customization Guide*for further details about using the Buffer Size Specification Facility.

CYLINDERS

FREESPACE

The default values of FREESPACE(0 0) must not be used. While you are entering initial information in the RECON, 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

KEYS

KEYS(32 0) is required.

NAME

NOERASE

NOREUSE

RECORDSIZE
A minimum of RECORDSIZE(4086 32600) is recommended. This is the default value when SPANNED is specified. 32600 is also the largest record size supported by the AMS REPRO and EXPORT functions with sequential output files.

Depending on the configuration of your IMS environment, the number of databases a subsystem might access, and the elapsed time between IMS startup and shutdown, you may want to increase the maximum record size beyond 32600. If this is done, the DBRC `BACKUP.RECON` command still functions correctly; however, the AMS REPRO and EXPORT functions cannot be used to copy the RECON to a sequential file.

If a new release of DBRC changes the length of RECON records, you may need to adjust the RECORDSIZE value so that jobs that ran in the lower release can continue to run in the new release.

For more information on record size changes, see *IMS/ESA Release Planning Guide*.

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

SPANNED

SPEED

UNORDERED

UNIQUE
If ICF catalogs are used, this keyword is not necessary.

VOLUMES

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

Avoid the use of WRITECHECK; it can degrade RECON I/O performance. The use of dual RECONs eliminates the need for WRITECHECK.

Security Considerations for RECON

All jobs that access the RECONs must have control-level authority to the RECONs. 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.

Initial RECON Access

When a job needs to read the RECON, the RECON must be opened. When three RECONs exist, DBRC determines which two are the active RECONs and which is the spare data set. Table 3 on page 50 shows how this determination is made.

- RECONA, RECONB, and RECONC represent the RECON1, RECON2, and RECON3 DD statements in no particular sequence.
- Data Set Status indicates the status of the data set during open time.

- Data Set Use indicates how DBRC assigns the data set.

Table 3. Determining Which RECONs Are Accessed

Case	DD Statement	Data Set Status	Data Set Use	DBRC Selection Criteria
1	RECONA RECONB RECONC	Create Mode Create Mode Create Mode	Copy1 Copy2 Spare	Specify INIT.RECON command Specify INIT.RECON command
2	RECONA RECONB RECONC	RECON Create Mode Create Mode	Copy1 Copy2 Spare	Current copy of RECON Produced from Copy1
3	RECONA RECONB RECONC	RECON RECON Create Mode	Copy1 Copy2 Spare	Current copy of RECON Current copy of RECON
4	RECONA RECONB RECONC	RECON RECON Create Mode	Copy1 Unused Copy2	Current copy of RECON Older copy than RECONA Produced from Copy1
5	RECONA RECONB RECONC	RECON RECON RECON	Copy1 Unused Unused	Current copy of RECON Older copy than RECONA Older copy than RECONA
6	RECONA RECONB RECONC	RECON RECON RECON	Copy1 Copy2 Unused	Current copy of RECON Current copy of RECON Older copy than RECONA
7	RECONA RECONB	RECON Create Mode	Copy1 Copy2	Current copy of RECON Produced from Copy1
8	RECONA RECONB	Create Mode Create Mode	Copy1 Copy2	Specify INIT.RECON command Specify INIT.RECON command
9	RECONA RECONB	RECON RECON	Copy1 Copy2	Current copy of RECON Current copy of RECON
10	RECONA RECONB	RECON RECON	Copy1 Unused	Current copy of RECON 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 RECONs are available, but one is now out of date. DBRC does not use the out-of-date RECON. Instead, it copies the up-to-date RECON to the spare data set.

Only one RECON 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. Otherwise, processing ends.

Records in the RECON

The RECON contains many types of records. Some records, such as header records, exist primarily to control RECON processing. 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 51 shows the major relationships of RECON record types.

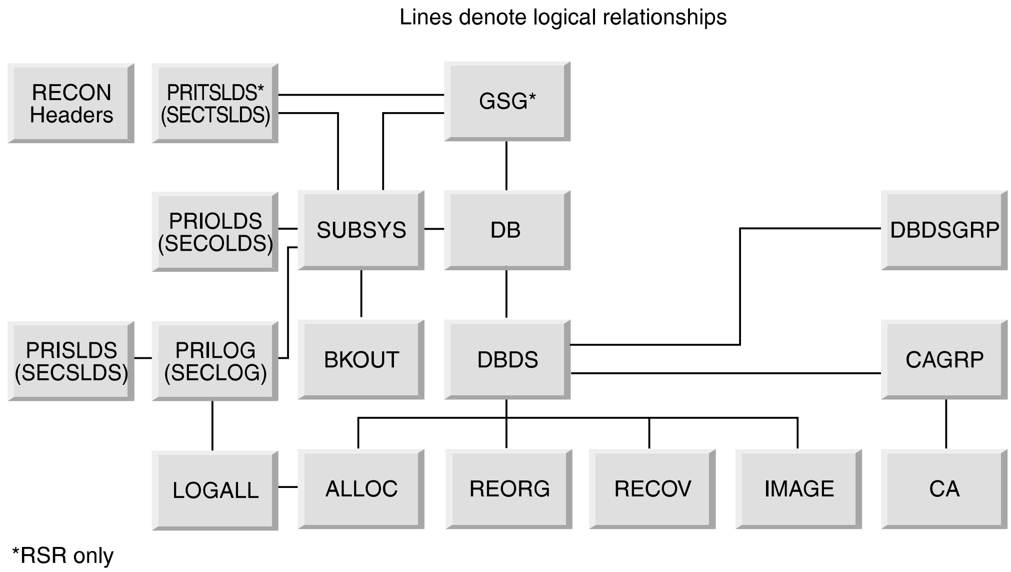


Figure 10. RECON Record Types

Related Reading:

See "Appendix C. Sample Listings of RECONs" on page 347 for details of record contents.

RECON Header Records

The RECON has three 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 configuration and state data that DBRC uses to process the RECONs.
- RECON Time History Table (THT) Extension

  The RECON THT extension contains a history of time-zone changes (such as when the system time was changed to and from Daylight Savings Time) during the period covered by the RECON. The THT is required for RECON migration and must be maintained as long as pre-IMS/ESA Version 6 systems share the RECON.

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:

For more information about DBRC and the archiving of online logs, see “Archiving Log Records” on page 41.

Log data sets output from IMS batch jobs are recorded in PROLOG / 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: For more information about interim primary and interim secondary-log data sets, see *IMS/ESA Operations Guide*.

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 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 2000 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 this in the `INIT.CAGRP` command that you use to create a CAGRP record in the RECON.

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:

- the data set name
- volume serial numbers
- the creation time stamp of the change accumulation data set that was created
- the stop time of the last log volume that the Database Change Accumulation utility processed
- or 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.

Database Data Set Group Record (DBDSGRP): You can use a DBDSGRP record to defines either:

a named group of databases, or

a group of DBDSs and DEDB areas.

You must keep the two in separate groups.

Although either type of group can be named in the DATAGROUP parameter for the `/STA`, `/STO`, and`/DBR` commands, the database group is preferred for those commands because it is more efficient.

You can tell the difference between a database group and a DBDS group in a RECON listing because the ″DDN or area″ is null for all members of a database group.

Related Reading: For more information on DBDS groups, see “Using DBDS Groups” on page 17.

Database Records (DB): DBRC treats DL/I and Fast Path database records differently. These types of records, their treatment, and contents are explained in this section.

DL/I Database Record: 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)

DL/I Database Data Set Record: 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 has a logical structure of records, as shown in Figure 11.

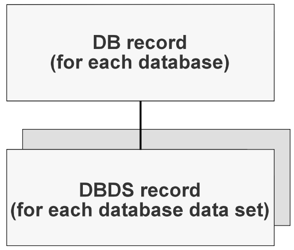


Figure 11. DBRC DL/I Records

Fast Path Database Records: To describe DEDBs, AREAs, and AREA Data Sets (ADSs), DBRC has a logical structure of records, as shown in Figure 12.

DBRC uses the DB and DBDS records to describe both DL/I databases and DEDBs; however, DBRC adds an ADS (area data set) 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 C. Sample Listings of RECONs” on page 347.

When an Area is registered in RECON, 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

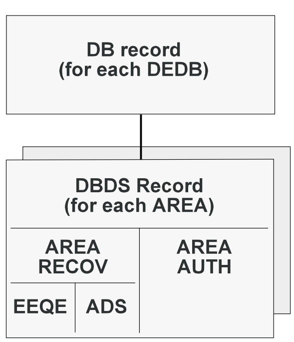


Figure 12. DBRC Fast Path Database Records

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.

Related Command:

- For more information on the REORG record, see the `NOTIFY.UIC` command.

Reorganization Record (REORG): DBRC creates a REORG record each time you run the HISAM Reorganization Reload utility or HD Reorganization Reload utility for a registered DBDS. DBRC creates a REORG record in the RECON for each DBDS that you reorganize.

Related Command:

For more information on the REORG record, see the `NOTIFY.REORG` command.

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 RECON.

Related Command:

For more information on the ALLOC record, see the `DELETE.ALLOC` and the `NOTIFY.ALLOC` commands.

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; in which case the RECOV record contains the time stamp of the recovery.
- A timestamp recovery; in which case 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:

For more information on the RECOV record, see the `NOTIFY.RECOV` command.

Subsystem Records (SSYS): RECON 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:

For more information on the SSYS record, see the `NOTIFY.SUBSYS` command.

Maintaining the RECONs

You should perform periodic maintenance on the RECONs to maintain data integrity and an acceptable level of performance. Maintenance of the RECONs 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
- Replacing a discarded RECON
- Recovering the RECON

Backing Up RECON

Back up the RECONs frequently. They are a critical resource. Always make backup copies of RECON 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. For additional information, see “Chapter 8. BACKUP Command” on page 103. 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. Its parameters determine whether it makes one or two copies.

Related Reading:

For additional information on the `BACKUP.RECON` command, see “Chapter 8. BACKUP Command” on page 103

Deleting Unnecessary RECON Log Records

You can delete unnecessary RECON records using the following methods:

- Automatic deletion by RECON
- 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 RECON 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 RECON. This is the image copy cleanup process.

At the same time that extraneous IMAGE records are deleted from RECON, 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) on that log
- Earliest restart checkpoint for the online IMS

PRILOG record compression deletes inactive data set entries up to the first 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 or after a CICS log EOV call is used. 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 RECON.

If the size of the PRILOG record exceeds a threshold percentage of the maximum RECON defined record size, then the PRILOG record is compressed. This 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.

Automatic PRILOG record compression occurs only if the size of the log exceeds the threshold value. Two threshold values are used: 50% and 75% of the maximum RECON defined record size. If the log cannot be compressed to the current threshold value, the threshold value is increased to the 75% value. Likewise, if the log can be compressed below the 75% threshold, PRILOG compression is attempted at every OLDS switch.

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, regardless of whether the PRILOG record exceeds the current threshold size.

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 RECONs from filling up because the space freed by deletions may not be reused by VSAM. However, if you delete the log records before backing up or reorganizing RECON, you are able to reclaim the space during backup or reorganization.

Related Reading: For more information on `DELETE.LOG`, see "DELETE.LOG (for OLDS)" on page 167. For more information on `LIST.LOG`, see "LIST.LOG (for a PRILOG family)" on page 239.

Reorganizing RECON

You need to reorganize the RECONs periodically. Many of the record keys in RECON 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 may not prevent RECON from filling up, because VSAM does not always reuse the space freed.

You can reorganize a RECON online if you are using dynamic allocation for the RECON. A spare RECON 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 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 (the one that needed reorganization). However, before you can delete and redefine this data set, 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. 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 is not available, you must wait for the online subsystem and all other subsystems that access RECON to deallocate it before you can reorganize.

Recommendation: Back up RECONs before and after reorganizing, using the following procedure:

1. Copy the RECONs 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.

RECON Reorganization Procedure

To clean up your logs and reorganize your RECONs, following 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 reorganization.
3. Issue a `CHANGE.RECON` with the REPLACE option for RECON1.
4. If your RECONs are shared across multiple CPUs, issue either an `/RMLIST DBRC='RECON STATUS'` or `/DISPLAY OLDS` command now on the other CPU. This is done so that your other CPU can detect that a RECON reconfiguration has occurred and can discard RECON1.
5. Run IDCAMS `DELETE` and `DEFINE CLUSTER` procedures for RECON1.
6. Repeat steps 3, 4, and 5 for RECON2.

7. Then repeat steps 3, 4, and 5 for the spare RECON.

Your log clean up has been performed and your RECONs have now been reorganized.

Replacing Damaged RECONs

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

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

After the spare RECON replaces the RECON that has the error, redefine the discarded RECON 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 RECONs.

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

Recovering the RECON

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. As soon as other subsystems that are still using the failed RECON have completed their processing, you can delete and redefine the RECON with the error so that it can be used as a spare. If, however, you want to analyze the RECON error, you should allocate new space for the RECON, rather than deleting and redefining it.

If a spare RECON is not available, all currently executing jobs continue processing using the RECON 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. This is not recommended as it jeopardizes the integrity of the system.

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

In an RSR environment, the isolated log sender (ILS) starts its own copy of DBRC. 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 RECONs so you can replace the unusable RECON. Issue `STOP ILS(gsg)` for each started ILS instance. Then issue `START ILS(gsg)` to bring up ILS and DBRC again.

What to Do If Both RECONs Are Unusable:

It is unlikely that both RECONs would be unusable; however, if both RECONs ever become unusable, follow this procedure:

1. Stop all jobs that require access to the RECON.
2. If you can access both RECONs, use the AMS `REPRO` command to back them up. This step is optional. You should do it, though, so that regardless of what happens, you are no worse off than you were at the start of this procedure.
3. Use the AMS utility to delete and redefine your RECONs.

4. Use the AMS REPRO command to restore one of the RECONs.
5. Use the AMS REPRO command to restore the other RECON from the first.
6. Use the LIST.RECON command to list one of the RECONs. Evaluate the list and determine which DBDSs have been updated since you made the backup in step 2 on page 61. 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 RECONs.

The RECONs are now restored and resynchronized with the databases.

If you do not control an excessive number of databases, it may be easier to follow this procedure:

1. Stop all jobs that require access to the RECON.
2. Define new RECONs.
3. Initialize these RECONs.
4. Register the environment (always keep a backup copy of the most recently initialized, but not-yet-used, RECON available).
5. Take image copies of all databases.

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

Replacing a Discarded RECON

DBRC detects that a RECON is discarded only when some activity occurs that causes DBRC to access the RECONs. You can have multiple instances of DBRC whenever you have multiple IMS subsystems, online or batch. You cannot delete and redefine a discarded RECON until all instances of DBRC detect that a change has occurred and they deallocate the discarded data set.

To redefine a RECON 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. This causes the online subsystems to obtain the same Copy1 and Copy2 RECONs and to deallocate the discarded RECON.
3. Use the AMS DELETE command to delete the discarded RECON.
4. Use the AMS DEFINE command to recreate the RECON as an empty VSAM KSDS data set. Use the same procedure that you used originally to create the RECON. See “Creating a RECON” on page 48.

Chapter 4. RECON Upgrade Utility (DSPURU00)

In This Chapter:

The RECON Upgrade utility converts a RECON from IMS/ESAVersion 4 Release 1 or Version 5 Release 1 to the format required for Version 6.

Before You Start

The steps listed below must be performed while no other subsystems, batch jobs, or utility jobs are accessing the RECON. All such jobs must be completed or terminated before the upgrade process is initiated.

Requirement: In order for pre-Version 6 subsystems and jobs to access the upgraded RECON, the coexistence SPE must be applied.

Related Reading: For more information on the coexistence SPE, refer to the chapters on migration and coexistence in theVersion 6 *IMS/ESA Release Planning Guide*.

The RECON Upgrade utility does not upgrade a RECON if any of the following conditions exist:

- The RECOVCTL option is in effect. This option must be changed to SHARECTL before the RECON can be upgraded.
- Any **local** subsystem record exists in the RECON. In this case, the utility assumes that the RECON is in use and terminates with a message. (At a tracking site, the RECON might contain SSYS records representing active subsystems at another site; it can still be upgraded.)
- A multiple-update operation failed leaving the RECON in a non-updated condition. The condition must be corrected before the RECON Upgrade utility can be run again. Issuing a `LIST.RECON` command normally corrects it.

  Attention: The operation being attempted at the time of failure might not have completed. It is necessary to examine the `LIST.RECON` output to determine if the command must be repeated.

Upgrade Procedure

Follow this procedure to convert a RECON to Version 6 format:

1. Quiesce all current jobs and subsystems accessing the RECON.
2. Use your **current** release of IMS to issue a `LIST.SSYS` command. If any SSYS records are listed, determine why they exist and take the appropriate action to remove them before proceeding.
3. Make a backup copy of the RECON; using your **current** release of IMS, issue the `BACKUP.RECON BOTH` command, ensuring that the data set is KSDS format. This command creates a copy of the RECON. This copy is input to the Upgrade utility, and can be used to restore the original RECON in case of a failure in the following steps.

Attention: If you have DBICF installed and have ″registered the RECON-ID in the RECON″ (which adds a special PRILOG / SECLOG in the RECON used as a report header by DBICF); your Upgrade may fail with a DSP0343I time conversion failure message. Delete the special PRILOG / SECLOG and repeat the Upgrade.

4. Install the new version of IMS.
5. Delete and redefine the RECON1, RECON2, and RECON3 data sets as empty data sets, each with a 32-byte key, using AMS.
6. Issue the following DBRC command to initialize the RECONs (defined in the preceding step) and create the Time History Table (THT).

   Attention: The THT is required for RECON migration and must be maintained as long as pre-IMS Version 6 systems share the RECON.

   The `INIT.RECON` command requires the two empty RECONs to run successfully.

   ```
   INIT.RECON UPGRADE
   ```

   The UPGRADE parameter allows the new RECON to be used only by the RECON Upgrade utility until the upgrade has successfully completed.

   Related Reading: For more information on THT migration recommendations, see the Migrating DBRC section in the Version 6 *IMS/ESA Release Planning Guide*.
7. Run the RECON Upgrade utility to convert the RECON. If the utility runs successfully, continue to the next step. If the utility fails, see the return code.

Code	Meaning
8	The upgrade process could not be started; the new RECON has not yet been changed. • Correct the problem. • Repeat this step.
12	The upgrade process completed but one or more records had timestamp fields containing invalid data. To identify the invalid data see the text of messages DSP0343I and DSP0350I. • Determine the necessary corrective action and whether to use the new RECON or correct the old one. If you choose the old RECON, repeat step 5.
16	The new RECON has been partially written. • Correct the problem. • Repeat the process beginning at Step 5.

   Attention: RECON records have increased in length. Consequently, the RECON Upgrade utility might fail with a “RECON RECORD TOO LONG” message. If so, return to Step 5, specifying a larger RECORDSIZE value for RECON1.

   Related Reading: For more information on RECORDSIZE recommendations, see section “Creating a RECON” on page 48.
8. Delete and redefine RECON2 and RECON3 data sets as empty data sets, each with a 32-byte key, using AMS. The next time DBRC accesses the upgraded RECON1 data set, DBRC copies its contents to one of these empty data sets, identifying the new copy as COPY2. The other defined data set is available as a spare. A batch job that issues the `LIST.RECON STATUS` command can be used to perform the copy operation.

Input and Output

Figure 13 shows input to and output from the RECON Upgrade utility.

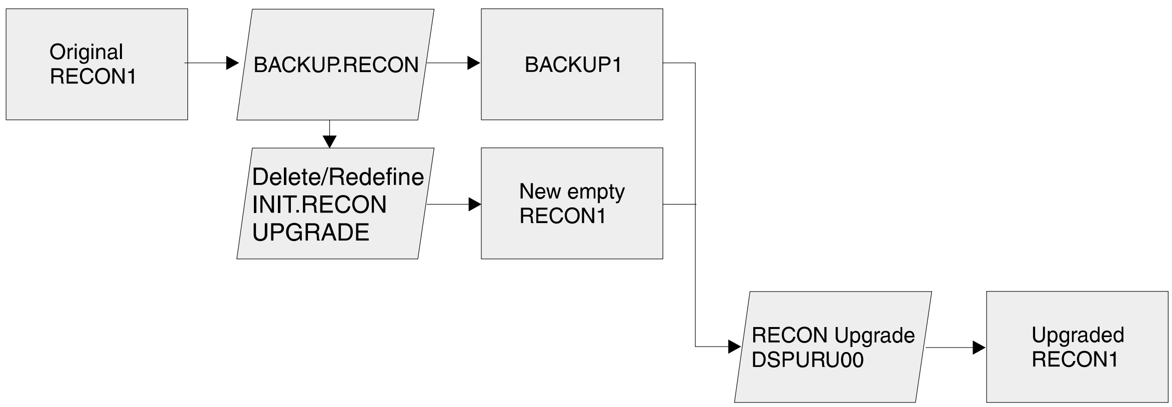


Figure 13. RECON Upgrade Utility Input and Output

JCL Requirements

The RECON Upgrade utility runs as a standard MVS job. A job statement, which you define, an EXEC statement, and DD statements.

EXEC
Specifies the utility DSPURU00 and a region size of at least 4096 KB.

The following DD statements are used to define inputs and outputs that are required:

SYSPRINT DD
Specifies a sequential message data set. The data set can reside on a tape, DASD volume, or printer, or it can be routed through the output stream.

BACKUP1 DD
Specifies the copy of the old RECON created at Step 3 on page 63.

RECON1 DD
Specifies the redefined RECON (from Step 5 on page 64) which is to be updated. Neither a RECON2 nor RECON3 DD statement can be specified.

Upgrade Utility Return Codes and Message Information

The RECON Upgrade utility issues the following return codes:

Code	Meaning
0	The operation completed successfully. The RECON is upgraded.
8	The operation was not performed. Possible reasons: • The input BACKUP1 data set could not be upgraded, so the output data set has not been changed. Refer to the message log that was written to the SYSPRINT data set for information on the cause of the error. • The output RECON1 data set was not suitable for upgrade. The data set may have been left in an unacceptable state following a previous upgrade attempt. Refer to the message log that was written to the SYSPRINT data set for information on the cause of the error.

12 The upgrade process completed, but one or more records had timestamp fields containing invalid data that is identified by messages DSP0343I and DSP0350I.

16 The RECON1 data set was left in a partially upgraded state; an error occurred while upgrading it. Refer to the message log that was written to the SYSPRINT data set for information on the cause of the error.

Related Reading: See the *IMS/ESA Messages and Codes* for more information on RECON Upgrade utility messages.

Example of RECON Upgrade Utility JCL

The RECON Upgrade utility processes the RECON IMS.RECON1. An example of a job to execute the RECON Upgrade utility with a JOBLIB statement is shown.

```
//UPGRADE      JOB (place your job parameters here)
//JOBLIB       DD      DSN=IMS61.RESLIB,DISP=SHR
//DOIT         EXEC    PGM=DSPURU00,REGION=4096K
//SYSPRINT     DD      SYSOUT=A
//BACKUP1      DD      DSN=IMS.RECONX,DISP=OLD
//RECON1       DD      DSN=IMS.RECON1,DISP=OLD
```

The RECON Upgrade utility processes the RECON IMS.RECON1. The following example is for a job which can be used to execute the RECON Upgrade utility with a STEPLIB statement.

```
//UPGRADE      JOB (place your job parameters here)
//DOIT         EXEC    PGM=DSPURU00,REGION=4096K
//STEPLIB      DD      DSN=IMS61.RESLIB,DISP=SHR
//SYSPRINT     DD      SYSOUT=A
//BACKUP1      DD      DSN=IMS.RECONX,DISP=OLD
//RECON1       DD      DSN=IMS.RECON1,DISP=OLD
```

Chapter 5. Database Recovery Control Utility (DSPURX00)

In This Chapter:

The DBRC utility supports commands that build and maintain the RECON, add information to the RECON, and generate jobs for utilities.

Commands submitted to the DBRC utility have the same general format. Each command is composed of a verb and a modifier, separated by a period and followed by parameters.

Related Reading: For information about command syntax, see "Chapter 7. DBRC Commands" on page 91.

Input and Output

Figure 14 shows the I/O requirements for DBRC. Notes on the figure follow.

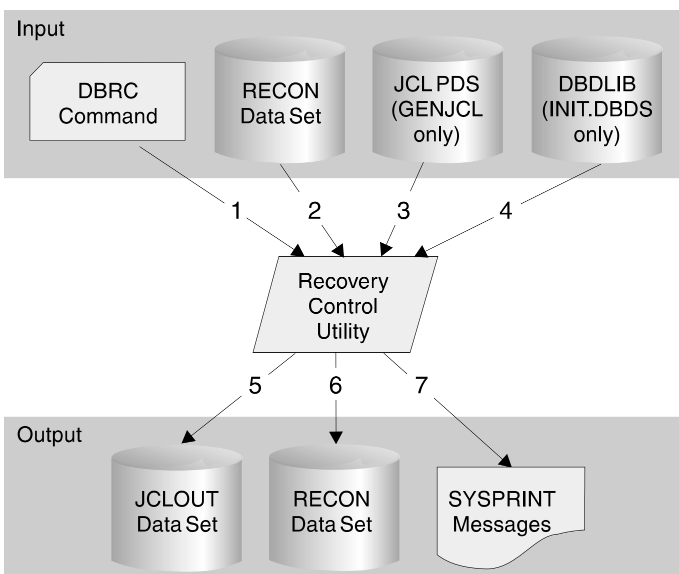


Figure 14. Database Requirements for the DBRC

Notes to Figure 14:

The input sources to DBRC are:

1. The DBRC command.
2. The RECON, which is the source of the name of the JCL PDS member that is to be used as the source of skeletal JCL for the utility.
3. The PDS which contains the JCL and control statements for the utility that DBRC uses to generate a job.
4. The data set that contains the database descriptions for the databases that are to be placed under the control of DBRC.

The output from the DBRC are:

5. Jobs created by GENJCL commands
6. The RECON, which may have been updated by the utility.
7. One or more of the following things:
   - A listing of the input commands.
   - Informational messages associated with their execution or diagnostic messages explaining any failures and return codes.
   - In the case of GENJCL commands, a listing of each job that was created.

Example of DBRC Utility JCL

This sample job initializes the RECON and registers one database with two DBDSs.

```
   //INITRCON JOB
 1 //INIT04   EXEC PGM=DSPURX00
 2 //STEPLIB  DD   DSN=IMS.RESLIB
 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 15. Inputs and outputs of the DBRC Utility

JCL Requirements

The DBRC utility runs as a standard MVS job. The numbers below refer to the JCL statements illustrated in the previous example.

1. EXEC Indicates the program to be executed.

2. STEPLIB Points to IMS.RESLIB, 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 RECON is allocated dynamically.

5. IMS Defines the IMS DBDLIB data set; it is required only for the NOTIFY.REORG and INIT.DBDS commands, and for the CHANGE.DBDS command if you change a DBDS ddname or area name.

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.)

Following are the input command statements:

9. INIT.RECON
Initializes the RECON 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 20. 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 20.

Chapter 6. Hints and Tips for DBRC

This chapter provides task-oriented instructions for frequently-used procedures, including the following topics:

In This Chapter:

Locating the Last SLDS Stop Time in RECON

Here is a short procedure describing how to find the last SLDS stop time in RECON using the GENJCL.USER command. This can be done when the IMS is still running and also works if the PRISLD is already closed.

1. First create an execution member. 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. Receive and interpret the GENJCL.USER command results and output. Below is the SLDS listing that results from issuing the previous sample GENJCL.USER command.

```
 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.USERcommand.

```
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 are true:

1. Situation - 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.
2. Situation - GENMAX Value is Too High

   No empty image copy data sets remain for the identified database data set or area data set. The maximum number of recovery generations to be maintained for this database data set or area data set has not yet been reached. (This would allow the reuse of existing image copy data sets.) The current execution of the image copy utility was successful, and the utility used the last remaining predefined image copy data set.

Message DSP0123I was displayed , stating that no predefined image copy data sets remain for the specified DBDNAME, DDNAME.

A subsequent execution of the Image Copy utility fails unless one of the following is done:

- Additional image copy data sets for the identified database data set or area data set are predefined, or
- The GENMAX value for the identified database data set or area data set is changed to permit the reuse of image copy data sets during the next execution of the image copy utility.

Solution for 1. Situation

Issue the `INIT.IC` command to define an available image copy of the required type, or issue the `CHANGE.DBDS` command to change the recovery period. The `INIT.IC` and `CHANGE.DBDS` commands can be issued only for DBDSs or DEDB areas registered to DBRC.

Issue the `INIT.IC` or `CHANGE.DBDS` commands with the GENMAX parameter increased to your new value.

1. Create a JCL job like one of the following job. These examples assume that your DBD name is THISDB, that your area name is AREA1, and that your previous GENMAX value was 2.

   Use this example for NOREUSE data sets.

   ```
   //CHNGDBDS JOB

   //SYSIN DD *
        CHANGE.DBDS DBD(THISDB) AREA(AREA1) -
                    GENMAX(4)
   /*
   ```

   Use this example for REUSE data sets.

   ```
   //INITIC JOB

   //SYSIN DD *
        INIT.IC DBD(THISDB) DDN(DDN1) -
                ICDSN(IMS.*.NEWICDSN)
   /*
   ```

2. Run the job.
3. Check the JES log. Ensure that the job ran successfully.
4. You can run a `LIST` command to see the new GENMAX value.
5. Run your image copy job.

Solution for 2. Situation

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.

Getting PRILOG Compression to Work

If your PRILOG is not getting compressed, do the following things:

- Check the EARLIEST CHECKPOINT time stamp listed in the PRILOG.

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.

- Review the allocation records for all databases listed in the LOGALL record that is associated with this PRILOG.

  If the oldest allocation time stamp is within the first DSN entry of the PRILOG, then it is preventing compression.

  Concurrent image copies (CICs) update the allocation time to reflect the earliest log that is needed for recovery. 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.

Sizing a PRILOG Record

DBRC abends when the PRILOG record exceeds the maximum record size defined for the RECONs.

The record size for the PRILOG record can be calculated using the following formula:

```
lrecl = (PRILOG_HEADER_LENGTH) +
        (NUMBER_OF_DSNs × DSN_LENGTH) +
        (NUMBER_OF_VOLSERS × LENGTH_OF_VOLINFO)

lrecl = 112 bytes +
        (NUMBER_OF_DSNs × 120 bytes) +
        (NUMBER_OF_VOLSERS × 32 bytes)
```

DBRC issues warning messages indicating that the PRILOG is approaching maximum size.

For each unarchived OLDS, DBRC assumes 16 VOLSERs are needed and calculates the size of each DSN entry according to this formula:

```
DSN ENTRY = 120 + (16 × 32) = 632 bytes
```

How many DSN entries fit in a 32,000 byte PRILOG record before DBRC abends if each entry has one volume and each OLDS is archived as used?

```
Answer:  202 DSNs

    (2 × 632) +
    112 + (202 × 120) + (202 × 32) = 32,080 bytes

Note:  When the OLDS switch occurs with PRILOG filling up,
       DBRC takes into account the space needed for the
       unarchived OLDS just closed and also allows space
       for the new OLDS to be used.
```

All numbers are in decimal.

The macros of the RECON record DSECTs are included in GENLIBB.

Using NOTIFY.PRILOG to Close an Open Online PRILOG

During timestamp recoveries, when DBRC reports that there are open allocations that are not needed for the recovery, it may be easier to close the open PRILOG rather than deleting the individual allocations.

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 command may be used to create a dummy DSN entry in the PRILOG:

```
 NOTIFY.PRILOG DSN(DUMMY) STARTIME(960011314544) FIRSTREC(011) -
  VOLSER(SC3390) SSID(IMSA)
 NOTIFY.PRILOG STARTIME(960011314544) LASTREC(015) -
   SSID(IMSA) RUNTIME(960021315001)
 CHANGE.PRILOG  STARTIME(960011314544) ERROR -
  DSSTART(960021315000)
```

Below is the PRILOG as it appears in RECON prior to issuing the `NOTIFY.PRILOG` command to close it:

```
PRILOG
 START = 96.001 13:14:54.4             *    SSID=IMSA      VERSION=6.1
 STOP  = 00.000 00:00:00.0                  #DSN=2
 GSGNAME=**NULL**
 FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 0
 EARLIEST CHECKPOINT = 96.001 13:14:54.9

 DSN=LOG1                                                UNIT=3400
 START = 96.001 13:14:54.4               FIRST DS LSN= 0000000000000001
 STOP  = 96.001 13:15:00.0               LAST  DS LSN= 0000000000000005
 FILE SEQ=0001    #VOLUMES=0001

   VOLSER=SC3390 STOPTIME = 96.001 13:15:00.0
     CKPTCT=0    CHKPT ID = 00.000 00:00:00.0

 DSN=LOG2                                                UNIT=3400
 START = 96.001 13:15:00.0               FIRST DS LSN= 0000000000000006
 STOP  = 96.002 13:15:00.0               LAST  DS LSN= 000000000000000A
 FILE SEQ=0001    #VOLUMES=0001

   VOLSER=SC3390 STOPTIME = 96.002 13:15:00.0
     CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
```

Below is the PRILOG as it appears in RECON after issuing the `NOTIFY.PRILOG` command to close it (differences are highlighted):

```
PRILOG
 START = 96.001 13:14:54.4             *    SSID=IMSA      VERSION=6.1
 STOP  = 96.002 13:15:00.1                  #DSN=3
 GSGNAME=**NULL**
 FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 0
 EARLIEST CHECKPOINT = 96.001 13:14:54.9

 DSN=LOG1                                                UNIT=3400
 START = 96.001 13:14:54.4               FIRST DS LSN= 0000000000000001
 STOP  = 96.001 13:15:00.0               LAST  DS LSN= 0000000000000005
 FILE SEQ=0001    #VOLUMES=0001

   VOLSER=SC3390 STOPTIME = 96.001 13:15:00.0
```

```
    CKPTCT=0    CHKPT ID = 00.000 00:00:00.0

DSN=LOG2                                                 UNIT=3400
START = 96.001 13:15:00.0               FIRST DS LSN= 0000000000000006
STOP  = 96.002 13:15:00.0               LAST  DS LSN= 000000000000000A
FILE SEQ=0001    #VOLUMES=0001

  VOLSER=SC3390 STOPTIME = 96.002 13:15:00.0
    CKPTCT=0    CHKPT ID = 00.000 00:00:00.0

DSN=DUMMY                                        ERROR  UNIT=3400
START = 96.002 13:15:00.0               FIRST DS LSN= 000000000000000B
STOP  = 96.002 13:15:00.1               LAST  DS LSN= 000000000000000F
FILE SEQ=0001    #VOLUMES=0001

  VOLSER=SC3390 STOPTIME = 96.002 13:15:00.1
    CKPTCT=0    CHKPT ID = 00.000 00:00:00.0
```

Deleting Log Records

As time passes, the RECON 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 167 and “DELETE.LOG (for RLDS and SLDS)” on page 168 for descriptions of the available DELETE.LOG commands and their parameters.

Use the DELETE.LOG INACTIVE INACTIVE command periodically to keep the RECON 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 RECON? If authorization for a database fails and you list the database and all the subsystem records in the RECON, you may wonder how the subsystem names (SSIDs) were created. The next section explains the naming conventions for the subsystem records.

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 conventions:

```
ONLINE IMAGE COPY - XXXTZZZZ where

          XXXT is the DMB number of DB + Trans DCB digit 0-9,A-Z
          ZZZZ is the control region IMSID
```

Online image copy is the only BMP that creates its own subsystem record in RECON.

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 RECON, 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, so maximum LRECL in the VSAM define cluster should be large enough to hold the maximum SSYS size. If the maximum LRECL is too small to hold the maximum SSYS size, IMS abends.

The record size for the subsystem record can be calculated using the following formula:

```
  72 + (number_of_db_authorized × 32)

Note:  72 = fixed portion of record
       32 = information for each database authorized
       (All numbers in decimal)
```

Removing an SSYS from DB Authorization

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; the following command can be used to remove the authorization.

```
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 MVS 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 RECON listing. 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 an IMS log in the RECON. 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. In rare cases, you might find the NOTIFY.PRILOG or NOTIFY.SECLOG command useful.

Related Reading:

See the CHANGE.PRILOG commands starting with "CHANGE.PRILOG (for OLDS)" on page 127, the CHANGE.SECLOG commands starting with "CHANGE.SECLOG (for OLDS)" on page 145, or the appropriate NOTIFY.PRILOG, or NOTIFY.SECLOG for details.

Reorganizing RECON to Increase Maximum RECORDSIZE

The following actions may result in the need to increase the maximum RECORDSIZE in your AMS (Access Method Services) format RECONs:

- Registering additional databases with DBRC or a change in workload that may cause the number of databases authorized to a subsystem to increase. This may cause the subsystem to grow beyond the current maximum RECORDSIZE.
- PRILOG compression is unable to compress the online PRILOG to keep it within the current maximum RECORDSIZE.

A new instance of DBRC cannot be started if the RECORDSIZE for two RECONs do not match. All batch activity must be quiesced. The online IMS subsystems can remain active if the RECONs are dynamically allocated, otherwise they must be shutdown.

Determine the status of the RECONs using one of the following methods:

- From online IMS, enter:

```
/RML DBRC='RECON STATUS'
```

- In batch, execute program DSPURX00 to do a LIST.RECON STATUS.

```
Example of current status:  RECON1  Copy1
                            RECON2  Copy2
                            RECON3  Spare
```

If the online IMS subsystems are shut down, follow this procedure:

- Use the DBRC BACKUP.RECON command or Access Method Services (IDCAMS) REPROcommand (or both) to backup RECON1. If you are reorganizing the

RECON because message DSP0029I was received, you must use the IDCAMS REPRO command to take the backup. The DFSMSdss (DFDSS) utility can be used in place of REPRO.

- Delete and redefine RECON1, RECON2, and RECON3 with the larger maximum RECORDSIZE.
- Using IDCAMS REPRO, copy the backup RECON into RECON1 and RECON2.

If the online IMS subsystems are active, follow this procedure:

- Delete and redefine the spare RECON using IDCAMS with a larger maximum RECORDSIZE.

```
Example:   RECON1 = Copy1   (RECORDSIZE 32K)
           RECON2 = Copy2   (RECORDSIZE 32K)
           RECON3 = Spare   (RECORDSIZE 64K)
```

- Issue the online command to force a copy of the RECON2.

```
/RMC DBRC='RECON REPLACE(RECON1)'
```

  RECON1 is discarded and deallocated.
- For ALL other online IMS subsystems that are active, issue the following command:

```
/RML DBRC='RECON STATUS'
```

```
Status of RECONs   RECON1 = discarded (RECORDSIZE 32K)
                   RECON2 = Copy1     (RECORDSIZE 32K)
                   RECON3 = Copy2     (RECORDSIZE 64K)
```

  This forces all the online subsystems to deallocate RECON1.
- Delete and redefine RECON1 with the larger RECORDSIZE.
- Issue the following online command to force a copy of RECON3:

```
/RMC DBRC='RECON REPLACE(RECON2)'
```

  RECON2 is discarded and deallocated.
- For ALL other online IMS subsystems that are active, issue the following command:

```
/RML DBRC='RECON STATUS'
```

```
Status of RECONs:  RECON1 = Copy2     (RECORDSIZE 64K)
                   RECON2 = discarded (RECORDSIZE 32K)
                   RECON3 = Copy1     (RECORDSIZE 64K)
```

  This forces all the online subsystems to deallocate RECON2.
- Delete and redefine RECON2 with the larger RECORDSIZE.
- Check the status of the RECONs by issuing the following command:

```
/RML DBRC='RECON STATUS'
```

```
Status of RECONs:  RECON1 = Copy2     (RECORDSIZE 64K)
                   RECON2 = spare     (RECORDSIZE 64K)
                   RECON3 = Copy1     (RECORDSIZE 64K)
```

Your reorganization is complete.

If you choose to reset the status of the RECONs so that RECON1 is Copy1, RECON2 is Copy2 and RECON3 is the spare; do the following:

- Issue this online command to force a copy of RECON1:

  ```
  /RMC DBRC='RECON REPLACE(RECON3)'
  ```

  RECON3 is discarded and deallocated.
- For ALL other online IMS subsystems that are active, issue this command:

  ```
     /RML DBRC='RECON STATUS'

  Status of RECONs:  RECON1 = Copy1     (RECORDSIZE 64K)
                     RECON2 = Copy2     (RECORDSIZE 64K)
                     RECON3 = discarded (RECORDSIZE 64K)
  ```

  This forces all the online subsystems to deallocate RECON3.
- Delete and redefine RECON3 with the RECORDSIZE (64 KB).
- Check the status of the RECONs again by issuing the following command:

  ```
     /RML DBRC='RECON STATUS'

  Status of RECONs:  RECON1 = Copy1     (RECORDSIZE 64K)
                     RECON2 = Copy2     (RECORDSIZE 64K)
                     RECON3 = spare     (RECORDSIZE 64K)
  ```

Recommendation: If your RECON RECORDSIZE is greater than 32 KB, 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.

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.

To specify that these data sets are cataloged specify:

```
//CHGRECON JOB
 ⋮
//SYSIN   DD  *
     CHANGE.RECON  CATDS
/*
```

Hints and Tips

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: For more information, see "INIT.RECON" on page 224 and "CHANGE.RECON" on page 138.

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 RECON 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 RECON). If you also need to delete the OLDS from RECON, 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 Enqueue Problems

The following section describes some reasons that you may experience RECON enqueue problems and things to watch out for to avoid them.

Related Reading: For more information, see “RECON Serialization” on page 45.

In a Shared DASD Environment

Operating in a shared DASD environment, the most common cause of RECON enqueue problems is failing to follow the recommendation to catalog each RECON in its own ICF catalog on the same volume as the RECON.

If you catalog each RECON 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 may 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 caused not by deadlock, but rather by contention and slow performance. There are a few things to look at in this situation.

1. Minimize the amount of time any job holds the RECON.

   One way to minimize that time is to tune the LSR buffer pool DBRC uses when accessing the RECONs. There is a CSECT, DSPBUFFS, that contains the values used for online, batch, and a procedure in the *IMS/ESA Customization Guide* on how to zap it to change the values. The defaults may be low for your usage.

2. Analyze the other contents of the RECON volumes. Consider isolating the RECON volumes, to prevent interference from other I/O activity (and vice versa). Consider placing the RECONs on high performance cached DASD, or perhaps solid state DASD.

Part 2. Command Reference

Chapter 7. DBRC Commands

This section 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:

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 create records in the 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 execute these commands online with the /RMxxxxxx command.

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/ESA Operator's Reference* for online command syntax.
- See *CICS/ESA-CICS Supplied Transactions* for a description of CDBM, and *CICS/ESA-CICS IMS Database Control Guide* for details of the DBCTL and DBRC commands you can use with the CDBM transaction.
- See "Using the Commands in this Book" on page 98 for information the syntax diagrams and notation used in this book.
- See "Data Set Naming Conventions" on page 20 for information on naming conventions.

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.

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.

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

There are no positional 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)
```

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 MVS JCL manuals.

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 255 characters.

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.

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 365)

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: For more information on the TIMEZIN parameter, see "CHANGE.RECON" on page 138.

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 95.

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'
```

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.

DBRC Commands Affected by the Time Stamp Format

These DBRC commands have time stamps as part of their syntax.

Related Reading: For more information on the standard time stamp format, see "Standard Time Stamp Format" on page 93.

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**

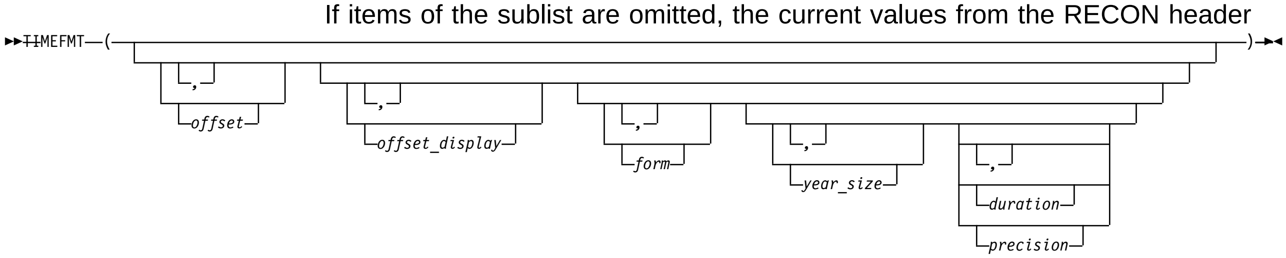
TIMEFMT Parameter Sublist

TIMEFMT(sublist)

An 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.

The format of the TIMEFMT parameter sublist is as follows:

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 assumed.

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: For more examples of `%SET` output specifications, see "Using the %SET TIMEFMT Keyword" on page 302.

precision

`precision` can only be coded on a `%SET` statement in skeletal JCL. It can be 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.

(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 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)**.

The defaults used by `GENJCL` commands are **TIMEFMT(O,O,C,2,1)**.

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.

DBRC Online Command Syntax

Related Reading: For details on DBRC commands that can be entered online, see *IMS/ESA Operator's Reference*.

Using the Commands in this Book

The following sections describe how the commands are presented in this book.

Notational Conventions

DBRC commands generally have both required and optional parameters.

The following list of symbols is used to define the format of the DBRC commands. The symbols, however, should not be used in an actual command unless indicated:

- A vertical bar (|) separates a pair of mutually exclusive parameters in a command; the last one is used by DBRC.
- An ellipsis (...) indicates that multiple entries of the type immediately preceding the ellipsis are allowed.
- Other punctuation (parentheses, commas, and spaces) must be entered exactly as shown.
- **Boldface** type indicates the exact characters to be entered. Such items must be entered exactly as shown.
- Lowercase words (for example, name) indicate parameters for which you supply the value.
- A default value is indicated in one of two ways. In a syntax diagram, the default appears above the horizontal line. In text, the default parameter is indicated by **underscored** text and may be noted in a separate paragraph. If the parameter is omitted, the default is assumed.

How to Read the Syntax Diagrams

The following rules apply to the syntax diagrams used in this book:

Arrow symbols

Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

►►── Indicates the beginning of a statement.

──► Indicates that the statement syntax is continued on the next line.

►── Indicates that a statement is continued from the previous line.

──►◄ Indicates the end of a statement.

Diagrams of syntactical units other than complete statements start with the ►── symbol and end with the ──► symbol.

Conventions

- Keywords, their allowable synonyms, and reserved parameters, appear in uppercase for MVS and OS/2 operating systems, and lowercase for UNIX operating systems. These items must be entered exactly as shown.
- Variables appear in lowercase italics (for example, *column-name*). They represent user-defined parameters or suboptions.
- When entering commands, separate parameters and keywords by at least one blank if there is no intervening punctuation.
- Enter punctuation marks (slashes, commas, periods, parentheses, quotation marks, equal signs) and numbers exactly as given.
- Footnotes are shown by a number in parentheses, for example, (1).
- A ƀ symbol indicates one blank position.

Required items

Required items appear on the horizontal line (the main path).

```
►►──REQUIRED_ITEM──────────────────────────────────────────────►◄
```

Optional Items

Optional items appear below the main path.

```
►►──REQUIRED_ITEM──┬───────────────┬───────────────────────────►◄
                   └─optional_item─┘
```

If an optional item appears above the main path, that item has no effect on the execution of the statement and is used only for readability.

```
                   ┌─optional_item─┐
►►──REQUIRED_ITEM──┴───────────────┴───────────────────────────►◄
```

Multiple required or optional items

If you can choose from two or more items, they appear vertically in a stack. If you *must* choose one of the items, one item of the stack appears on the main path.

```
►►──REQUIRED_ITEM──┬─required_choice1─┬────────────────────────►◄
                   └─required_choice2─┘
```

If choosing one of the items is optional, the entire stack appears below the main path.

```
►►──REQUIRED_ITEM──┬──────────────────┬────────────────────────►◄
                   ├─optional_choice1─┤
                   └─optional_choice2─┘
```

Repeatable items

An arrow returning to the left above the main line indicates that an item can be repeated.

```
                   ┌─────────────────┐
                   ▼                 │
►►──REQUIRED_ITEM────repeatable_item─┴─────────────────────────►◄
```

If the repeat arrow contains a comma, you must separate repeated items with a comma.

```
                   ┌─,───────────────┐
                   ▼                 │
►►──REQUIRED_ITEM────repeatable_item─┴─────────────────────────►◄
```

Command Syntax

A repeat arrow above a stack indicates that you can specify more than one of the choices in the stack.

Default keywords
IBM-supplied default keywords appear above the main path, and the remaining choices are shown below the main path. In the parameter list following the syntax diagram, the default choices are underlined.

```
                 ┌default_choice──┐
►►──REQUIRED_ITEM┼────────────────┼──────────────────────────────►◄
                 ├optional_choice─┤
                 └optional_choice─┘
```

IMS-specific syntax information

Fragments
Sometimes a diagram must be split into fragments. The fragments are represented by a letter or fragment name, set off like this: | A |. The fragment follows the end of the main diagram. The following example shows the use of a fragment.

```
►►──STATEMENT──item 1──item 2──┤ A ├──────────────────────────►◄
```

A:

```
├──┬item 3─┬──KEYWORD──┬─────────────┬───────────────────────┤
   └item 4─┘           │  └─item 5─┘ │
                       └─────────────┘
                          └─item 6─┘
```

Substitution-block
Sometimes a set of several parameters is represented by a substitution-block such as **<A>**. For example, in the imaginary /VERB command you could enter `/VERB LINE 1`, `/VERB EITHER LINE 1`, or `/VERB OR LINE 1`.

```
►►──/VERB──┬─────┬──LINE──line#──────────────────────────────►◄
           └─<A>─┘
```

where <A> is:

```
►►──┬EITHER─┬────────────────────────────────────────────────►◄
    └OR─────┘
```

Parameter endings
Parameters with number values end with the symbol '#', parameters that are names end with 'name', and parameters that can be generic end with '*'.

```
►►──/MSVERIFY──┬─MSNAME──msname──┬──────────────────────────►◄
               └─SYSID──sysid#───┘
```

The MSNAME keyword in the example supports a name value and the SYSID keyword supports a number value.

Chapter 8. BACKUP Command

The `BACKUP` command creates backup copies of the RECON.

BACKUP.RECON

```
             ┌─RECON1──┐
►►──BACKUP.RECON─┼─────────┼──────────────────────────────────────────►◄
             ├─RECON2──┤
             └─BOTH────┘
```

Use the `BACKUP.RECON` command to create backup copies of the RECON from the Copy1 RECON. The command first invokes a process to clean up the RECON. The command then invokes the MVS AMS `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 a backup of the RECON is required because message DSP0029I was received, you cannot use the `BACKUP.RECON` command. The `BACKUP.RECON` command will also fail with message DSP0029I. Instead, use Access Method Services (IDCAMS) REPRO or DFSMSdss (DFDSS). The process to clean up the RECON will occur automatically during later processing.

Parameters

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 change information in the RECON.

CHANGE.ADS

```
►►──CHANGE.ADS──ADDN(name)──AREA(name)──DBD(name)──┬─────────────────┬──────►
                                                   └─ADDNNEW(name)───┘

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

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 see "INIT.ADS" on page 211.

Parameters

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.

CHANGE.ADS

```
//CHGADS   JOB
⋮
//SYSIN    DD   *
     CHANGE.ADS DBD(DBD001) AREA(AREA002)  -
                ADSN(ADSN004) ADDN(ADDN004)
/*
```

CHANGE.BKOUT

```
►►──CHANGE.BKOUT──SSID(name)──UOR(uor)──UORTIME(time stamp)──────────────────►

►──┬───────────────────────────────┬──┬──────────────┬──┬──────────────────────┬──►
   └─DELETE(──┬─UOR────────┬─)─────┘  └─PSB(name)────┘  │       ┌─,────────┐  │
              ├─ALL────────┤                            └─DBD(──▼──name────┴─)┘
              │  ┌─,────┐  │
              └──▼─name─┴──┘

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

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.

Parameters

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.

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.

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.

Example of Using the CHANGE.BKOUT Command

```
//CHGBKOUT JOB
 :
//SYSIN    DD  *
     CHANGE.BKOUT  SSID(SYS3)-
                   UOR(E2E8E2F340404000000000600000003)-
                   UORTIME(930931345027) DELETE(ALL)-
                   DBD(DATA1,DATA2,DATA3C)-
                   BKO(DATA4,DATA5,DATA3A)
/*
```

CHANGE.CA

```
►►──CHANGE.CA──GRPNAME(name)──RECTIME(time stamp)──┬────────────┬──────────────►
                                                   └─CADSN(name)─┘

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

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

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.

Parameters

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 93).

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 DBRC.

INVALID
Specifies that the change accumulation data set is not to be used as input for a subsequent run of change accumulation or DBRC. 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 213 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 DBRC. 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

```
>>--CHANGE.CAGRP--GRPNAME(name)--+-----------------------------------+---------->
                                 |       .-,-----------------.        |
                                 |       V                   |        |
                                 +-ADD(----(dbname,ddname)---+--)-----+
                                 |          .-,-----------------.     |
                                 |          V                   |     |
                                 '-DELETE(----(dbname,ddname)---+--)--'

>--+----------------+--+---------------------+--+---------------+--+-----------+------><
   '-CAJCL(member)--'  +-DEFLTJCL(member)-+  '-GRPMAX(value)--'  +-NOREUSE-+
                       '-NODEFLT----------'                      '-REUSE---'
```

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.

Parameters

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 2000 members.

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. *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 179.

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 111.

For additional information about reusing change accumulation data sets, see the explanation of the REUSE parameter of the `INIT.CAGRP` command 214.

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

```
►►──CHANGE.DB──┬─DBD(name)─┬──┬────────┬──┬─────────────────────────────┬──────────────►
               └─ALL───────┘  ├─AUTH───┤  ├─NOBACK──────────────────────┤
                              └─NOAUTH─┘  └─BACKOUT(value)SSID(name)────┘

►─┬─────────┬──┬──────────────────────┬──┬──────────┬──┬─────────┬──────────────────────►
  ├─READON──┤  └─SHARELVL(──┬─0─┬──)─┘  ├─NONRECOV─┤  ├─TYPEFP──┤
  └─READOFF─┘               ├─1─┤       └─RECOVABL─┘  └─TYPEIMS─┘
                            ├─2─┤
                            └─3─┘

►─┬──────────────────┬──┬──────────┬─────────────────────────────────────────────────►◄
  ├─GSGNAME(gsgname)─┤  ├─RCVTRACK─┤
  └─NOTCOVER─────────┘  └─DBTRACK──┘
```

Or:

```
                                                                ┌─ACTIVE───┐
►►──CHANGE.DB──UNAUTH──DBD(name)──┬────────────┬──SSID(name)──┤          ├────────►◄
                                  └─AREA(name)─┘               └─TRACKING─┘
```

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 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 data base or area (DB/AREA) record. This inconsistency occurs when either the SSYS record still has an entry for the

DB/AREA in its authorized data bases/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 UNAUTH parameter with the syntax shown above.

Restrictions: The following restrictions apply when you use the UNAUTH parameter:

- Only the DBD, AREA, SSID, and ACTIVE | TRACKING parameters are valid with UNAUTH. If any other parameters are specified, 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, then the command fails.
- The ACTIVE or TRACKING parameter must match the SS ROLE field in the associated subsystem information entry of the specified database or area. If these do not match, the command fails.

Parameters

DBD(*name*) | ALL
Mutually exclusive, required 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.

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.

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.

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. Message DSP1044I is issued if you attempt to change the covered state of the database with this command. In order to change a database from not covered to covered, issue the following two commands for the database:

- DELETE.DB
- INIT.DB

Neither GSGNAME nor NOTCOVER can be specified for Fast Path DEDBs. For DL/I databases, neither GSGNAME nor NOTCOVER can be specified while the database is in use. A non-recoverable database cannot be assigned to a GSG.

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 dta bases/areas list.

SSID is required with either the BACKOUT or UNAUTH parameters If coded without either BACKOUT or UNAUTH, the command fails.

NONRECOV | RECOVABL

Mutually exclusive, optional parameters used to specify whether update logs of the specified database are recorded in the RECON. These parameters work with a specified DBD(name) and do not work if ALL is specified.

If the database is registered as RECOVABL, 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. These parameters are meaningful for TYPEIMS databases only, and are rejected for TYPEFP databases.

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.

RCVTRACK | DBTRACK
Mutually exclusive, optional parameters you use to specify the type of RSR tracking (shadowing) for a database that is 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.

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. If any subsystem has already authorized the database, this parameter does not modify the database record. 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/ESA 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/ESA 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 and resets the LKASID parameter to NOLKASID. See "CHANGE.DBDS" on page 117 for explanations of the CFSTR1, LKASID, and NOLKASID 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.

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

```
>>--CHANGE.DBDS--DBD(name)--+-DDN(name)--+--+------------------------+-->
                            '-AREA(name)-'  |           .-,-----.    |
                                            |           V       |    |
                                            +-ADDEQE(---value---+-)--+
                                            |           .-,-----.    |
                                            |           V       |    |
                                            '-DELEQE(---value---+-)--'

>--+--------------+--+--------------+--+--------+--+----------------+-->
   '-CFSTR1(name)-'  +-CFSTR2(name)-+  +-AUTH---+  +-DDNNEW(name)---+
                     '-NOCFSTR2-----'  '-NOAUTH-'  '-AREANEW(name)--'

>--+------------------+--+-----------+--+---------------+--+------------------+-->
   +-DEFLTJCL(member)-+  '-DSN(name)-'  '-GENMAX(value)-'  +-GSGNAME(gsgname)-+
   '-NODEFLT----------'                                    '-NOTCOVER---------'

>--+----------------+--+-------+--+---------+--+-----------------+--+----------+-->
   '-ICJCL(member)--'  +-ICON--+  +-NOREUSE-+  '-OICJCL(member)--'  +-LKASID---+
                       '-ICOFF-'  '-REUSE---'                       '-NOLKASID-'

>--+---------+-------------------------------------------------------------->
   +-PRELOAD-+
   '-NOPREL--'

>--+---------+--+----------+--+---------+--+------------------+------------->
   +-PREOPEN-+  +-RCVTRACK-+  +-RECOV---+  '-RECOVJCL(member)-'
   '-NOPREO--'  '-DBTRACK--'  '-NORECOV-'

>--+---------------+--+-----------------+--+-------+----------------------><
   '-RECOVPD(value)-'  '-RECVJCL(member)-'  +-VSO---+
                                            '-NOVSO-'
```

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.

Parameters

DBD(*name*)
Required parameter you use to identify by its database name the DBDS or DEDB area whose record is to be changed.

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 MVS 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 MVS coupling facility structure naming conventions.

Related Reading: For details on coupling facility structure naming conventions, see *IMS/ESA Administration Guide: Database Manager*.

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, you should use the `INIT.IC` command to create additional image copy records in RECON. If the number of data sets does not equal GENMAX, reuse 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. In the associated database record, a counter increases to indicate how many DBDSs need an image copy.

ICOFF
Specifies that a DBDS does not need an image copy. In the associated database record, a counter decreases to indicate 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, you should use the `INIT.IC` command to add additional data sets; otherwise reuse does not take place. See the explanation for 119.

For additional information about reusing image copy data sets, see “INIT.DBDS” on page 216 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.

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 the next time 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 the next time 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 MVS 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.

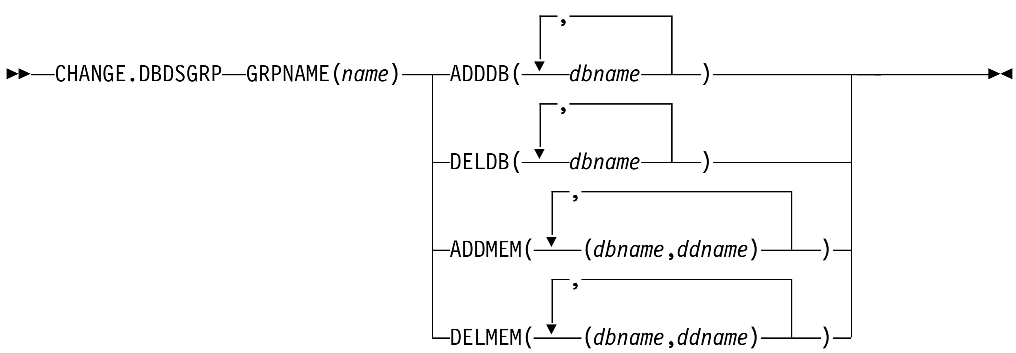
If an area was previously specified as SHARELVL(2 | 3), changing the area to NOVSO clears the coupling facility structure names and resets the LKASID setting to NOLKASID. NOVSO cannot be specified if the area is in use.

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.

Parameters

GRPNAME(*name***)**
Required parameter you use to identify the DBDSGRP to be changed. A record with that name must already exist.

ADDDB(*dbname***)**

DELDB(*dbname***)**

ADDMEM(*dbname,ddname***)**

DELMEM(*dbname,ddname***)**
Mutually exclusive, optional parameters you use to identify the member or members to be added to or deleted from the group. Note that a group can contain a maximum of 2000 members, and that any member can belong to more than one group.

ADDDB

DELDB
Identify one or more databases or DEDB areas by name. The group must have been defined as a database group.

ADDMEM

DELMEM
Identify 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 name. The group must have been defined as a DBDS.

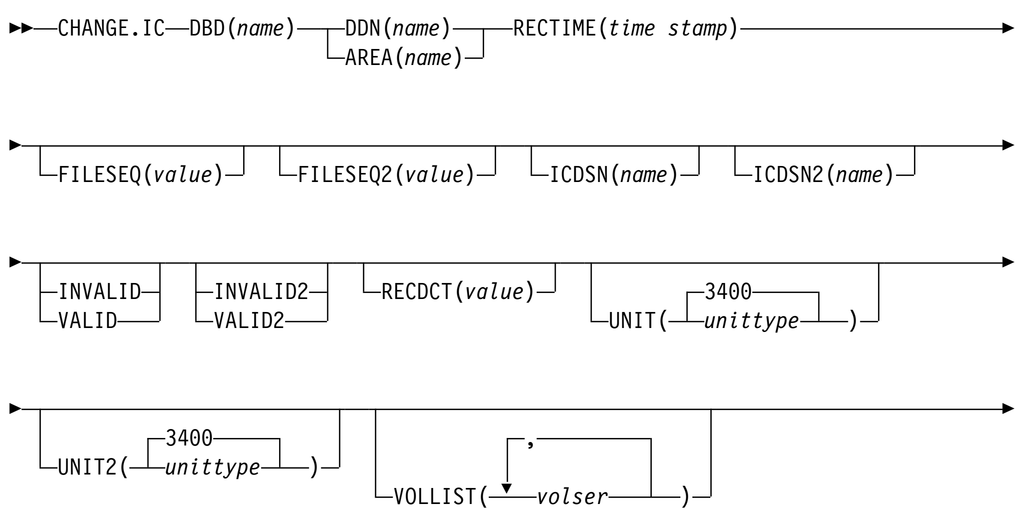
If you delete all the members of a group, the record of that group is deleted from RECON.

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



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

Use a `CHANGE.IC` command to modify information contained in an image copy record in RECON.

Parameters

DBD(*name***)**
: Required parameter you use to identify the database name of the DBDS whose image copy record is to be modified.

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 93

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 you use to prevent or permit the use of an image copy data set as input to a subsequent run of an image copy job or DBRC.

  INVALID
  : Prevents the use of the specified image copy data set as input to a subsequent run of DBRC or an image copy job. If the invalidated image copy data set is reused, it is automatically marked as valid. (See “INIT.DBDS” on page 216 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 DBRC or an image copy job.

INVALID2 | VALID2
Mutually exclusive, optional parameters you use to prevent or permit the use of a duplicate image copy data set as input to a subsequent run of the Database Image Copy utility or DBRC.

INVALID2
Prevents the use of the specified, duplicate image copy data set as input to a subsequent run of the Database Image Copy utility or DBRC. If the invalidated, duplicate image copy data set is subsequently reused, it is automatically marked as valid. (See "INIT.DBDS" on page 216 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 Image Copy utility or DBRC.

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 93) 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.PRILOG (for OLDS)

```
►►──CHANGE.PRILOG──OLDS(ddname)──┬───────────┬──┬─────────┬──┬───────────┬──────────►
                                 ├─ARNEEDED──┤  ├─AVAIL───┤  └─DSN(name)─┘
                                 ├─ARSCHED───┤  └─UNAVAIL─┘
                                 ├─ARSTART───┤
                                 └─ARCHIVED──┘

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

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.

Parameters

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)

```
>>--CHANGE.PRILOG--RLDS--STARTIME(time stamp)--+--------------+-------->
                                               '-DSN(name)----'

>--+----------------------------+--+------------------------------+---->
   |            .-,-----.       |  |             .-,-------.      |
   |            V       |       |  |             V         |      |
   '-CHKPTCT(------value-+---)--'  '-CHKPTID(-------chkptid-+---)-'

>--+----------------------+--+--------+--+----------------+--+----------------+-->
   '-DSSTART(time stamp)--'  +-ERROR--+  '-FILESEQ(value)-'  '-GAP(--+-ON--+--)-'
                             '-NORMAL-'                              '-OFF-'

>--+---------------+--+-----------------------------------+------------->
   '-GSG(gsgname)--'  |            .-,------------.        |
                      |            V              |        |
                      '-NEWTIME(------time stamp--+---)----'

>--+--------------------------+--+--------------------------+----------->
   |           .-,------.     |  |           .-,------.     |
   |           V        |     |  |           V        |     |
   '-NEWVOL(------volser-+--)-'  '-OLDVOL(------volser-+--)-'

>--+----------------------+--+-------------------------------------+---->
   '-PREVGAP(--+-ON--+--)-'  |             .-,------------.        |
               '-OFF-'       |             V              |        |
                             '-RUNTIMES(------time stamp--+---)----'

>--+----------------------------+--+---------------------------+------><
   |        .-3400------.       |  |            .-,------.     |
   '-UNIT(--+-unittype--+---)---'  |            V        |     |
                                   '-VOLLIST(------volser-+--)-'
```

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.

Parameters

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 93).

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 93). 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 93).

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 258 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. If you are changing the first volume of a data set, the first time stamp in NEWTIME must be greater than the STARTIME parameter. Each time stamp must be in standard form (see “Standard Time Stamp Format” on page 93).

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 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. 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 93).

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 258.

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” on page 130 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:

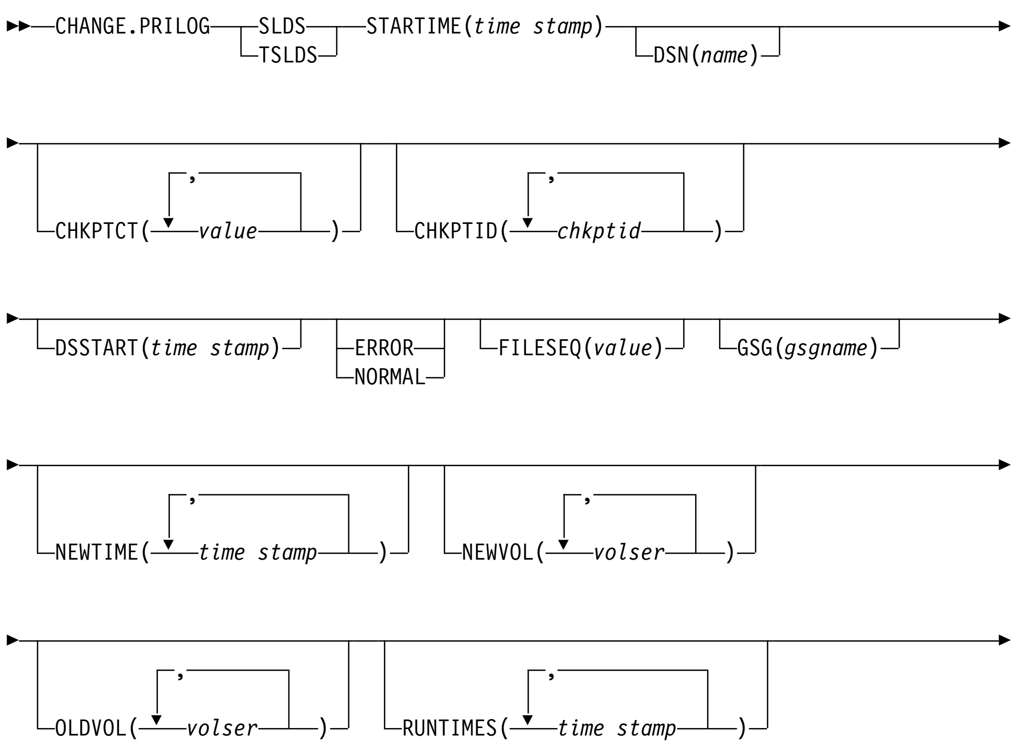
```
//CHGPRILG JOB
 .
 .
 .
//SYSIN    DD   *
     CHANGE.PRILOG  RLDS STARTIME(842331243299)   -
                    OLDVOL(VOL003,VOL004)         -
                    NEWVOL(VOL007,VOL008,VOL009)
/*
```

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)



CHANGE.PRILOG (for SLDS and TSLDS)

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

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.

Parameters

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 93).

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 93). 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 93).

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 263 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. If you are changing the first volume of a data set, the first time stamp in NEWTIME must be greater than the STARTIME parameter. Each time stamp must be in standard form (see “Standard Time Stamp Format” on page 93).

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. 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 93).

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 263.

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. 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.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 STARTIME(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 STARTIME(820541212120) -
                    DSSTART(820541212120) NORMAL
/*
```

CHANGE.RECON

```
►►──CHANGE.RECON──┬─────────┬──┬──────────────────────────────┬──────────────►
                  ├─CATDS───┤  │            ┌─3400─────┐      │
                  └─NOCATDS─┘  └─DASDUNIT(──┴─unittype─┴──)──┘

►──┬─────────────────┬──┬──────────┬──┬─────────┬──┬──────────┬──────────────►
   ├─DUAL────────────┤  ├─FORCER───┤  ├─NOCHECK─┤  ├─LISTDL───┤
   └─REPLACE(RECONn)─┘  └─NOFORCER─┘  ├─CHECK17─┤  └─NOLISTDL─┘
                                      └─CHECK44─┘

►──┬─────────────────────────┬──┬─────────┬──┬────────────┬──┬──────────┬────►
   └─LOGRET('time_interval')─┘  ├─NOCOEX──┤  └─SSID(name)─┘  ├─STARTNEW─┤
                                └─COEX────┘                  └─NONEW────┘

►──┬────────────────────┬──┬──────────┬───────────────────────────────────────►
   └─TAPEUNIT(unittype)─┘  ├─TRACEON──┤
                           └─TRACEOFF─┘

►──┬───────────────────────────────────────────────┬──────────────────────────►
   │           ┌─,──────────────────────────────┐  │
   │           ▼                                │  │
   └─TIMEZONE(───(label──┬─────────┬──┬───┬─────┴─)┘
                         └─,offset─┘  └─)─┘

►──┬───────────────────────────────────┬──┬────────────────┬─────────────────►◄
   └─TIMEZIN(offset_rep ,duration)─────┘  └─TIMEFMT(sublist)┘
```

Notes:

1 The *offset* subparameter of the TIMEZONE parameter must be omitted in order to delete an entry.

Use a `CHANGE.RECON` command to update the RECON header record.

Parameters

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 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. 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.

DASDUNIT(3400 | *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. These parameters are intended to be used in an IMS online environment.

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. REPLACE is recommended only for online use.

Related Reading: For more information about using REPLACE, see “Chapter 3. Initializing and Maintaining the RECON” on page 43.

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 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.

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.

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 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.

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 93 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')  = 10 days, 12 hours, 30 seconds
```

Related Reading:

- See “DELETE.LOG (for RLDS and SLDS)” on page 168 for a more information on deleting inactive logs.

NOCOEX | COEX

Mutually exclusive, optional parameters you use to specify whether pre-version 6 and version 6 RECONs may coexist.

NOCOEX

Use this keyword if you no longer intend to run any pre-version 6 jobs against the RECON. It disables coexistence, removing limitations on version 6 processing, when the local clock is changed (such as for daylight saving time). You should be sure that the decision to use the NOCOEX keyword is appropriate. It is possible to reverse this decision but, see the Attention description below.

COEX

Allows you to reverse the effect of the NOCOEX keyword, again enabling coexistence with pre-version 6 RECONs.

Attention: Use this keyword with extreme caution. If the local clock has been changed since coexistence was disabled, or if you have used the NOCOEX keyword to recover from an ABEND U2475, in version 6 (not recommended), endless loops and other unpredictable malfunctions in your pre-version 6 systems may result.

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.

STARTNEW | NONEW
Mutually exclusive, optional parameters you use to specify whether new jobs are to be started when fewer than two RECONs are 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.

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: For more information about the TIMEZONE parameter, see “Suggestions for Time Zone Label Table Management” on page 143.

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 93 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".

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 MVS 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***)**
An 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 93 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 Sublist" on page 95.

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 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.RECON (for THT or REPTHT)

```
►►──CHANGE.RECON──┬────────┬──────────────────────────────────────────►◄
                  ├─THT────┤
                  └─REPTHT─┘
```

Use a `CHANGE.RECON` (for THT or REPTHT) command to specify whether an additional new time history table (THT) is created or the existing entry is replaced.

These parameters are mutually exclusive with one another and all other `CHANGE.RECON` parameters and subparameters. So, to affect the time history table with `CHANGE.RECON`, enter **only** THT **or** REPTHT.

Parameters

THT | REPTHT
Mutually exclusive, optional parameters you use to specify whether an additional new Time history table is created or the existing entry is replaced, when the current THT entry differs from the current MVS offset.

THT
Generates a new THT entry if the current MVS offset from UTC differs from the current THT entry.

REPTHT
Replaces the current THT entry. Use this keyword only if you have previously changed the clock incorrectly and generated a new THT entry, then corrected the clock. This keyword can only be used if `neither` of the following are true:

- The RECON has been updated by Version 6 DBRC (except by your command)
- The RECON has been accessed by a pre-Version 6 version since you generated the current THT entry

Otherwise, you must use the THT keyword to create a new THT entry.

Related Reading: For information on the time history table (THT), see "RECON Header Records" on page 51.

Example of Specifying a Replacement THT Entry

In this example, you replace the THT entry that you created after an incorrect clock change.

```
//CHGRECON JOB
⋮
//SYSIN    DD   *
     CHANGE.RECON  REPTHT
/*
```

CHANGE.SECLOG (for OLDS)

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

►──┬────────────┬────────────────────────────────────────────────────────────►◄
   └─SSID(name)─┘
```

Use a CHANGE.SECLOG (for OLDS) command to change information in RECON about a secondary OLDS.

Parameters

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.

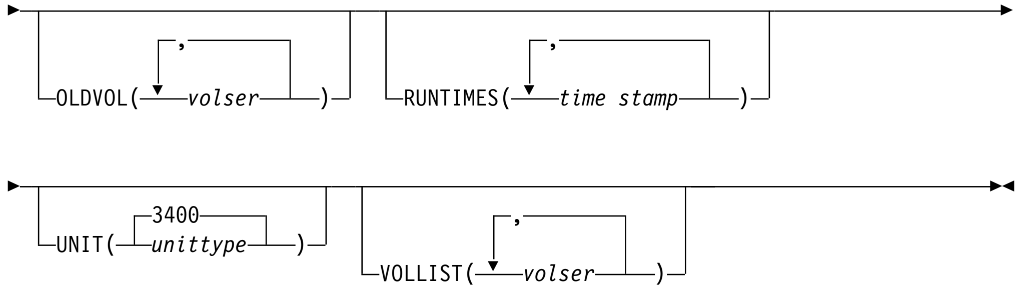
Example Showing a SECOLDS Error

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)

```
                 ┌RLDS┐
►►──CHANGE.SECLOG─┴────┴─STARTTIME(time stamp)─┬───────────┬──────────►
                                               └─DSN(name)─┘

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

►─┬────────────────────┬─┬────────┬─┬───────────────┬─┬─────────────┬─►
  └─DSSTART(time stamp)┘ ├─ERROR──┤ └─FILESEQ(value)┘ └─GSG(gsgname)┘
                         └─NORMAL─┘

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



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.

Parameters

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 93).

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 93). 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 93).

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 258 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. If you are changing the first volume of a data set, the first time stamp in NEWTIME must be greater than the STARTIME parameter. Each time stamp must be in standard form (see “Standard Time Stamp Format” on page 93).

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 93).

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 258.

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” on page 130 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)
/*
```

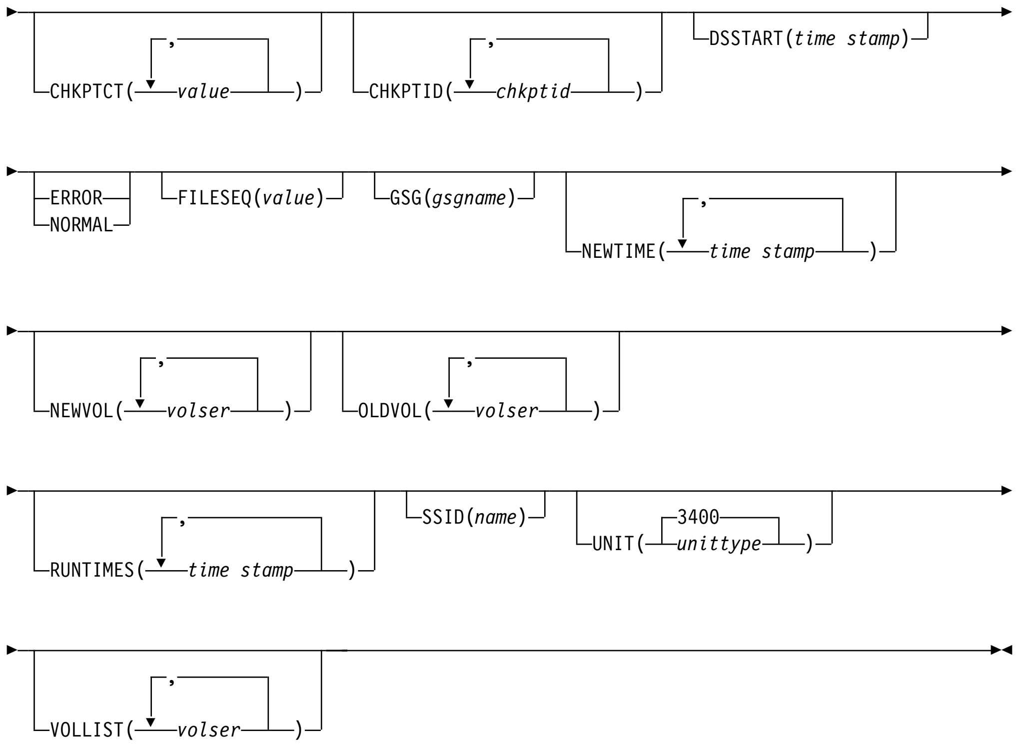
Example of changing volume stop times

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)

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



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.

Parameters

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 93).

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 93). 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 93).

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 263 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. If you are changing the first volume of a data set, the first time stamp in NEWTIME must be greater than the STARTIME parameter. Each time stamp must be in standard form (see “Standard Time Stamp Format” on page 93).

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 93).

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 263.

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

```
►►──CHANGE.SG──SGNAME(sgname)──GSGNAME(gsgname)──┬──────────┬──┬──────────┬──┬───────┬──────►◄
                                                 ├─ACTIVE───┤  ├─LOCAL────┤  └─NORTA─┘
                                                 └─TRACKING─┘  └─NONLOCAL─┘
```

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.

Example:

This command changes SG1 to a tracking role:

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

In this case, SG2 is automatically changed to an active role.

Parameters

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
/*
```

CHANGE.SUBSYS

```
              ┌─ALL────────┐
►►──CHANGE.SUBSYS──┴─SSID(name)─┴──┬──────────────────┬──┬──────────────┬──┬──────────┬──►
                                   ├─BACKIRLM(name)─┤  └─IRLMID(name)─┘  ├─NORMAL───┤
                                   └─NOBACKUP───────┘                    └─ABNORMAL─┘

►──┬───────────┬──┬────────────┬──────────────────────────────────────────────────►◄
   ├─STARTRCV──┤  └─TRACKING───┘
   └─ENDRECOV──┘
```

Use a `CHANGE.SUBSYS` command to modify information that is contained in a subsystem record in RECON.

Parameters

SSID(*name*) | ALL
Required parameter 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 MVS or IMS subsystem identification name. If you specify ALL, the requested processing is done 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

```
►►──CHANGE.UIC──DBD(name)──┬─DDN(name)──┬──RECTIME(time stamp)──UDATA('string')──────────►◄
                           └─AREA(name)─┘
```

Use a CHANGE.UIC command to modify information in the image copy record in RECON that corresponds to a nonstandard image copy data set.

Parameters

DBD(*name*)
Required parameter you use to identify the database name of the DBDS for which a nonstandard image copy data set exists.

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 93).

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

DELETE.ADS

```
►►──DELETE.ADS──ADDN(name)──AREA(name)──DBD(name)──────────────────────────────►◄
```

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.

Parameters

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

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

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 only be deleted 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.

Parameters

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.

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(*timestamp*)
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 93).

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

```
►►──DELETE.BKOUT──SSID(name)──────────────────────────────────────►◄
```

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.

Parameters

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

```
►►──DELETE.CA──GRPNAME(name)──RECTIME(timestamp)──────────────────────────►◄
```

Use a `DELETE.CA` command to delete from RECON a change accumulation run record for a specified CA group.

Parameters

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 timestamp must be in standard form (see "Standard Time Stamp Format" on page 93).

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

```
►►──DELETE.CAGRP──GRPNAME(name)──────────────────────────────────────────►◄
```

DELETE.CAGRP

Use a `DELETE.CAGRP` command to delete a CA group record and all associated CA run records from RECON.

Parameters

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

```
►►──DELETE.DB──DBD(name)──────────────────────────────────────────►◄
```

Use a `DELETE.DB` command to delete from RECON a database record and all its associated DBDS records. If any subsystem is authorized to use the specified database, the command fails and none of the RECON records are deleted.

Parameters

DBD(*name*)
Required parameter you use to identify the name of the database whose records are 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.

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

```
►►──DELETE.DBDS──DBD(name)──┬─DDN(name)──┬──────────────────────────────►◄
                            └─AREA(name)─┘
```

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.

Parameters

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

```
►►──DELETE.DBDSGRP──GRPNAME(name)─────────────────────────────────────►◄
```

Use a DELETE.DBDSGRP command to delete the record of a specified DBDS group from RECON.

Parameters

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
 ⋮
```

DELETE.DBDSGRP

```
//SYSIN      DD   *
     DELETE.DBDSGRP   GRPNAME(DBDSGRP1)
/*
```

DELETE.GSG

```
►►──DELETE.GSG──GSGNAME(gsgname)──────────────────────────────────────►◄
```

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.

Parameters

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

```
►►──DELETE.IC──DBD(name)──┬─DDN(name)──┬──RECTIME(timestamp)──ICDSN2(name)──────►◄
                          └─AREA(name)─┘
```

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.

Parameters

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.

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(*timestamp*)
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 93).

ICDSN2(*name*)
Optional parameter you use to specify the name of a duplicate image copy data set for which information is to be deleted from an image copy record. (The record of the first image copy data set remains in RECON.)

Example of Deleting Information from an Image Copy Record

In this example, information about a duplicate image copy data set is deleted from an image copy record in RECON. The parameters DBD, AREA, ICDSN2, and RECTIME identify the information to be deleted. The asterisk (*) in the ICDSN2 parameter is to be expanded by DBRC according to the default-naming convention for image copy data sets.

```
//DELIC    JOB
 .
 .
 .
//SYSIN    DD   *
     DELETE.IC   DBD(DBDKSDS1)     AREA(AREA006)  -
       RECTIME(841231223221)  ICDSN2(IMS.*.ICDSN5)
/*
```

DELETE.LOG (for OLDS)

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

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

Parameters

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.

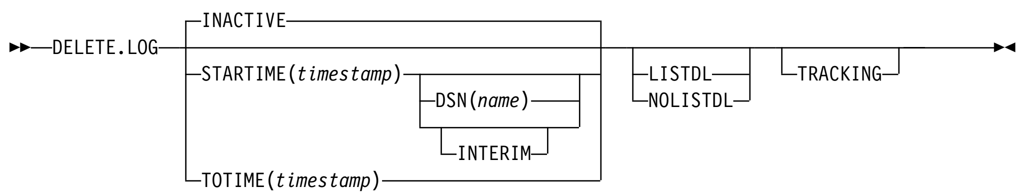
  The SSID is an eight-character string of any alphanumeric characters that comprise a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. Use the `INIT.RECON` or `CHANGE.RECON` command to set the default subsystem 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)



With this command you can 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)

Use this command to prevent the PRILOG records from exceeding the maximum RECON record size.

You can also delete records of SLDSs created by RSR tracking IMS subsystems. In addition, you can delete single data set entries from RLDS and SLDS records.

Parameters

INACTIVE | STARTIME(*timestamp*) | TOTIME(*timestamp*)
: 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 exceeds the log retention period 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 does not delete PRILOG and SECLOG records for logs that do not satisfy all three conditions. Most logs satisfy the conditions. Use STARTIME(*timestamp*) for those logs that do not satisfy all three conditions.

Active PRILOG records are also examined to determine whether they should be compressed, and to delete any inactive data set entries in the records. 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 specified in the `INIT.RECON` or `CHANGE.RECON` command.
- It is older than the earliest log volume required for recovery of any DB that is registered in the RECON.
- It is older than the earliest checkpoint that is required for system restart.

See *Deleting Log Records* and *Getting PRILOG Compression to Work* for more information about deleting log records and compression.

STARTIME
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 you use to specify the data set name of a particular log data set 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.

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.

STARTIME is required with DSN. If DSN is not specified, the entire RLDS or SLDS record is deleted. You cannot specify TOTIME, INTERIM, or TRACKING with DSN.

Only log data at the tracking site is eligible for deletion.

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 deleted. You must specify a time that is less than the current time, minus the log retention period. The time stamp must be in standard form (see “Standard Time Stamp Format” on page 93). If the PRILOG is older than TOTIME but it is still active, it will be examined to delete all of the inactive data set entries in the record.

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, which is set 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.

The default is the value specified by the `INIT.RECON` or `CHANGE.RECON` 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. This parameter must be specified with SSID, STARTIME, and TOTIME.

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.RECOV

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

Use a `DELETE.RECOV` command to delete a specified recovery run record from RECON. Specifying `DELETE.RECOV` for the recovery run record of a timestamp 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 timestamp 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 timestamp recovery occurred.

Parameters

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.

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(*timestamp***)**
: 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 93).

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

```
►►──DELETE.REORG──DBD(name)──DDN(name)──RECTIME(timestamp)──────────────────────────►◄
```

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.

Parameters

DBD(*name***)**
: Required parameter you use to identify the reorganization record to be deleted; *name* is the database name of the related DBDS.

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(*timestamp*)
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 93).

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

```
►►──DELETE.SG──GSGNAME(gsgname)──SGNAME(sgname)──────────────────────────►◄
```

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.

Parameters

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

```
►►—DELETE.SUBSYS—SSID(name)——————————————————————————————————►◄
```

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 156.

To close the subsystem log, issue the NOTIFY.PRILOG command, and then issue the DELETE.SUBSYS command.

Parameters

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

```
►►—DELETE.UIC—DBD(name)——┬─DDN(name)──┬——RECTIME(timestamp)——————————————►◄
                         └─AREA(name)─┘
```

Use a DELETE.UIC command to delete the record of a nonstandard image copy data set from RECON.

Parameters

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.

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(*timestamp***)**
Required parameter you use to specify the timestamp of the nonstandard image copy record to be deleted. Use the timestamp with an adjacent asterisk (*) in a list of the IMAGE record. The timestamp must be in standard form (see “Standard Time Stamp Format” on page 93).

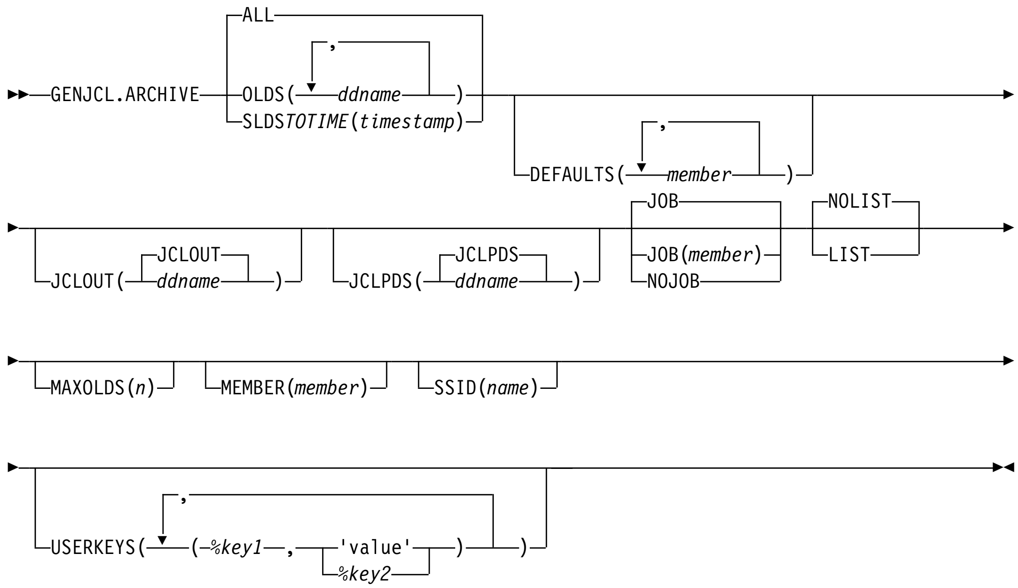
Example of Deleting a Nonstandard Image Copy Data Set Record

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

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 "Using the Commands to Generate JCL and User-Defined Output" on page 293.

Parameters

ALL | OLDS(*ddname*) | SLDS

Mutually exclusive, required parameters 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(*timestamp*)
Specifies that only tracking log data sets with start times older than or equal to *timestamp* 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 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 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 “Using the Commands to Generate JCL and User-Defined Output” on page 293.

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 “Understanding Simple Keywords” on page 295.

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

```
►►──GENJCL.CA──GRPNAME(name)──┬─────────────────────────┬─┬──────────────────────────┬──►
                              │            ┌─,──────┐   │ │          ┌─JCLOUT─┐      │
                              └─DEFAULTS(──▼─member─┴─)─┘ └─JCLOUT(──┴─ddname─┴──)──┘

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

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

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

►──┬──────────────┬──────────────────────────────────────────────────────────────────►◄
   ├─VOLNUM(value)┤
   └─CATIME(time)─┘
```

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 “Using the Commands to Generate JCL and User-Defined Output” on page 293.

Parameters

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 (<u>JCLOUT</u> | *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 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 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 “Using the Commands to Generate JCL and User-Defined Output” on page 293.

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 “Understanding Simple Keywords” on page 295.

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 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 93).

Examples of Running the Change Accumulation Utility

Here are some examples of using the GENJCL.CAcommand.

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

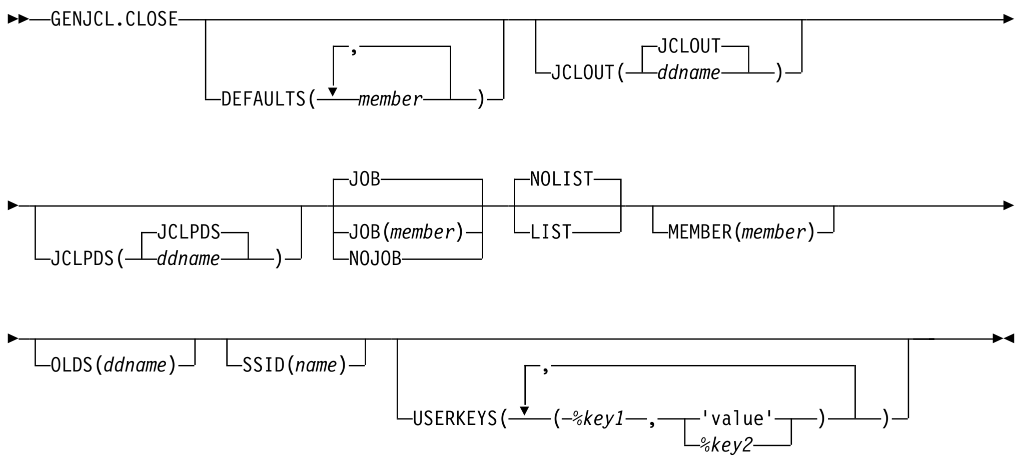
In this example, the GENJCL.CA command generates JCL and control statements to run the Change Accumulation utility for CAGRP3, which is defined as REUSE. JCL execution member CAJCLB is taken from the JCLPDS data set identified by the JCLPDS DD statement.

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 "Using the Commands to Generate JCL and User-Defined Output" on page 293.

Parameters

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(<u>JCLOUT</u> | *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(<u>JCLPDS</u> | *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.

<u>JOB</u> | JOB(*member***)|NOJOB**
Mutually exclusive, optional parameters 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 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 "Using the Commands to Generate JCL and User-Defined Output" on page 293.

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 “Understanding Simple Keywords” on page 295.

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)
/*
```

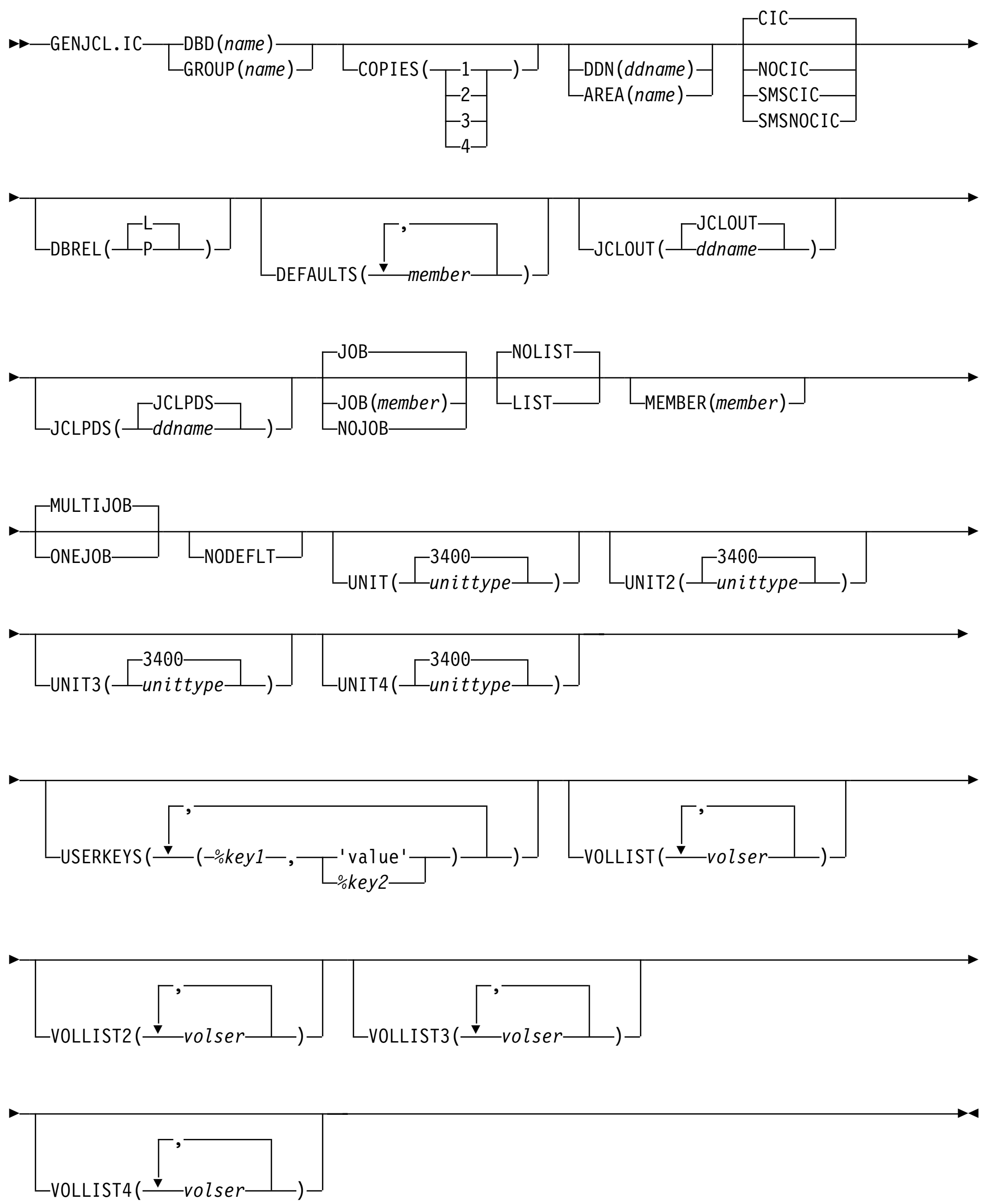
After the close job runs, the PRIOLDS record in RECON that corresponds to the OLDS is updated to indicate the successful close.

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 “Using the Commands to Generate JCL and User-Defined Output” on page 293.

Parameters

DBD(*name*) | GROUP(*name*)
Mutually exclusive, required parameters you use to specify the DBDS or DEDB area to be copied.

DBD
Specifies the name of the database that contains the DBDS or area to be copied.

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.

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.

If the specified DBDS is identified in RECON with the REUSE attribute, you must specify the COPIES parameter if you want two or more image copy data sets; otherwise only 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.

DDN(*name***) | AREA(***name***)**
Mutually exclusive, optional parameters 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.

CIC | NOCIC | SMSCIC | SMSNOCIC
Optional, mutually exclusive, parameters you use to indicate how the image copy is to be taken.

CIC
Specifies that the Database Image Copy (DFSUDMP0) utility is to be used to take an image copy concurrent with update processing.

NOCIC
Specifies that the Database Image Copy (DFSUDMP0) utility is to be used to take an image copy while the database is unavailable for update processing.

SMSCIC
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 DFSMS concurrent copy.

SMSNOCIC
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 DFSMS concurrent copy.

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 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 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 "Using the Commands to Generate JCL and User-Defined Output" on page 293. If this parameter is not specified, the default specified on the `INIT.DBDS` command is used.

MULTIJOB | ONEJOB
Mutually exclusive, optional parameters you use to control how many JOB 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 multi-step job for the whole group.

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 “Understanding Simple Keywords” on page 295.

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 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='

⋮
//SYSIN   DD *
D1 DBDKSDS1 DDNKSDS1 DATAOUT1
/*
```

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='
⋮
//SYSIN   DD *
D1 DBD1GRP1 DDN1GRP1 DATAOUT1
/*
//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      -
              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 DATAOUT1 DATAOUT2 DATAOUT3 DATAOUT4 S
/*
```

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.

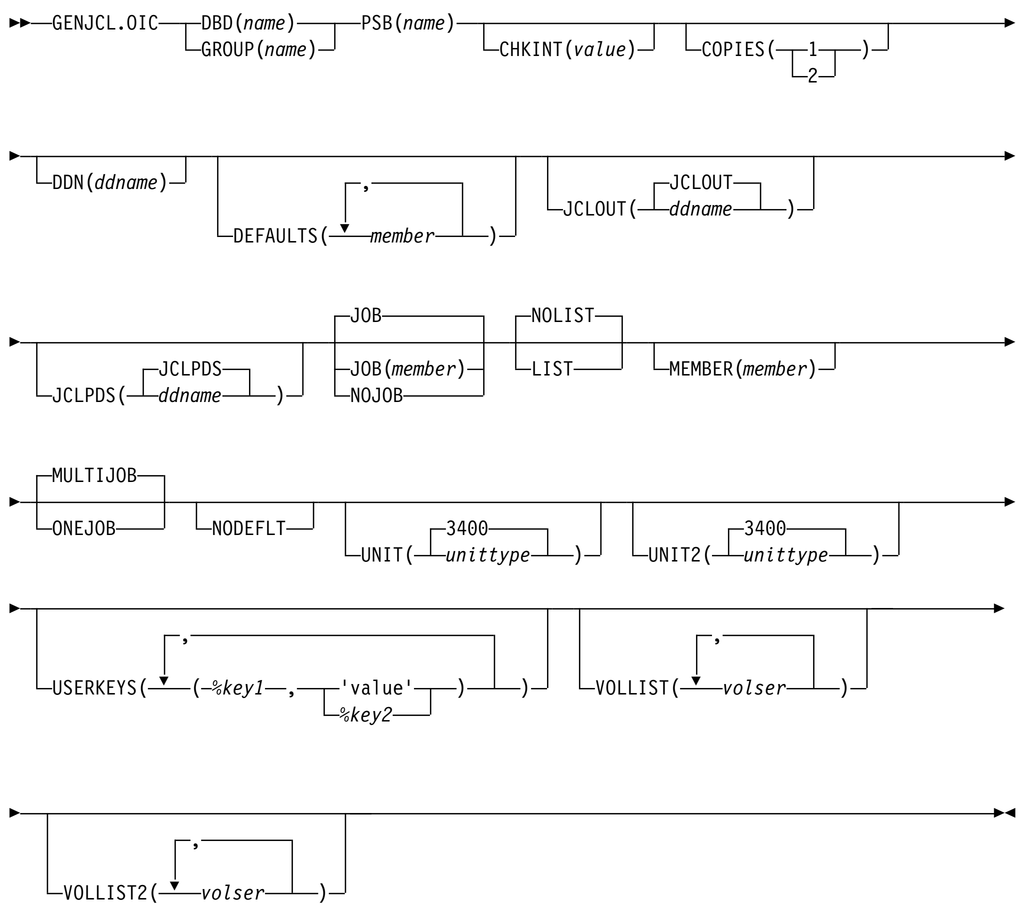
```
//GENJIC4  JOB
//JCLOUT   DD   . . .
//JCLPDS   DD   . . .
```

```
⋮
//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
/*
```

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 "Using the Commands to Generate JCL and User-Defined Output" on page 293.

The GENJCL.OIC command and online image copy cannot be used on databases at an RSR-tracking site.

Parameters

DBD(*name*) | GROUP(*name*)
Mutually exclusive, required parameters 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.

GROUP
Specifies that all DBDSs of a DBDS group are to be copied. If GROUP is specified, the `GENJCL.IC` 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.

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 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 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 "Using the Commands to Generate JCL and User-Defined Output" on page 293. If this parameter is not specified, the default specified on the `INIT.DBDS` command is used.

MULTIJOB | ONEJOB
Mutually exclusive, optional parameters 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 “Understanding Simple Keywords” on page 295.

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.

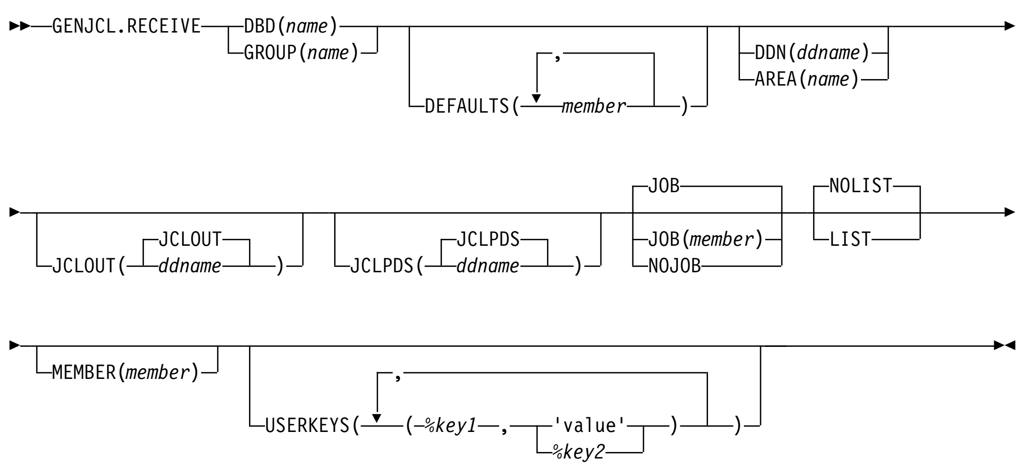
```
//GENJOIC  JOB
//JCLPDS   DD   . . .
//JCLOUT   DD   . . .

⋮
//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
- 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.

For information about the IBM-supplied skeletal JCL execution member that is used by `GENJCL.RECEIVE`, see "Using the Commands to Generate JCL and User-Defined Output" on page 293.

Parameters

DBD(*name*) | GROUP(*name*)
Mutually exclusive, required parameters 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.

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 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.

If DDN or AREA is not specified, the `GENJCL.RECEIVE` command executes repeatedly, once for each DBDS or area of the specified database.

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 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 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 “Using the Commands to Generate JCL and User-Defined Output” on page 293. 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 “Understanding Simple Keywords” on page 295.

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

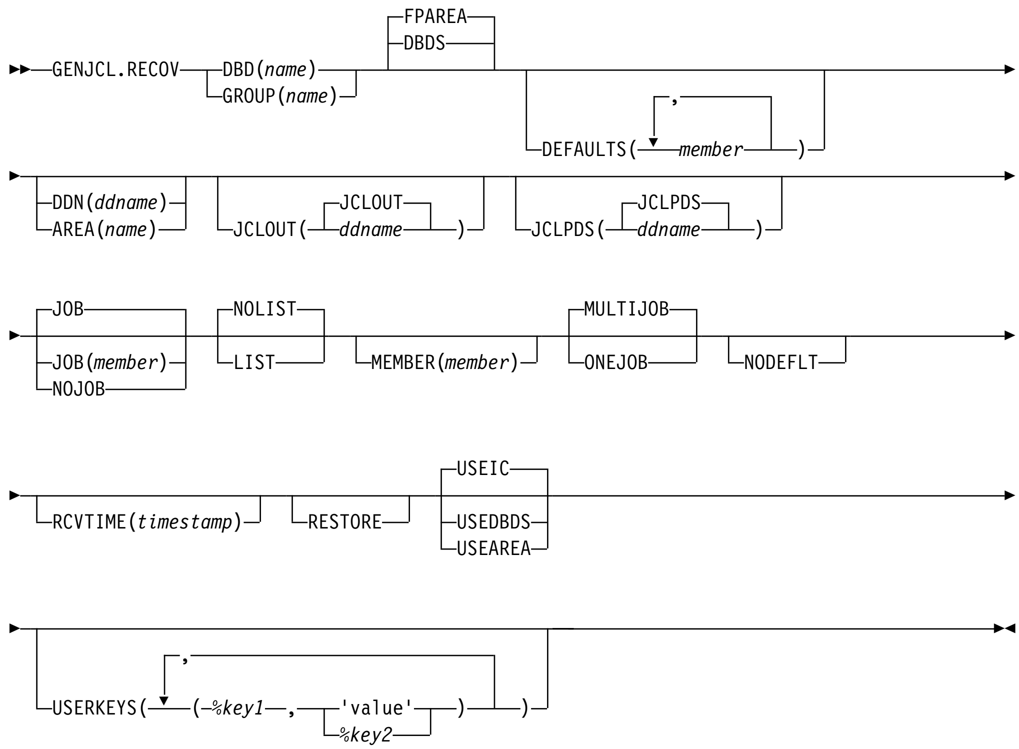
In this example, the `GENJCL.RECEIVE` command generates JCL and control statements to run the database recovery utility to receive image copies for all DBDSs of GROUP1.

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 timestamp recovery of a specified DBDS or area. All log data must be archived; otherwise the GENJCL.RECOV command fails.

Restriction: 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.

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 “Using the Commands to Generate JCL and User-Defined Output” on page 293.

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

Parameters

DBD(*name***) | GROUP(***name***)**
Mutually exclusive, required parameters you use to specify the DBDSs that are to be recovered.

DBD
Specifies the database name of the DBDSs to be recovered.

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.

DBDS | FPAREA
Mutually exclusive, optional parameters you use to specify the type of recovery to be done for the specified DBDS or area.

DBDS
Generates JCL for either a full recovery or a timestamp recovery of a DBDS.

If you have specified the DDN parameter, DBDS is the default.

FPAREA
Generates JCL for a full recovery or a timestamp recovery of an area.

If you have specified the AREA parameter, FPAREA is the default.

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 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.

If DDN or AREA is not specified, the `GENJCL.RECOV` command executes repeatedly for each DBDS or area of the specified database.

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 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 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 “Using the Commands to Generate JCL and User-Defined Output” on page 293. If this parameter is not specified, the default specified on the `INIT.DBDS` command is used.

MULTIJOB | ONEJOB
Mutually exclusive, optional parameters 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(*timestamp***)**
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 302 for more information on full recovery.
- See *IMS/ESA Operations Guide* for more information on time stamp recovery.

RESTORE
Optional parameter used to generate JCL for a database data set that is designated as nonrecoverable. If the last image copy was taken **before** the DBDS was designated as nonrecoverable, normal recovery JCL is generated to recover the DBDS up to the point of the recovery-status change. If the last image copy was taken **after** the DBDS was designated as nonrecoverable, the generated JCL uses only the image copy for recovery.

If you attempt an implicit or explicit group execution with recoverable and nonrecoverable DBDSs (and RESTORE is specified), JCL is generated only for the nonrecoverable DBDSs.

Do not specify RESTORE for a recoverable DBDS.

USEIC | USEDBDS | USEAREA
Mutually exclusive, optional parameters 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 timestamp 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 timestamp recovery in which an image copy data set is used as input.

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 "Understanding Simple Keywords" on page 295.

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)
/*
```

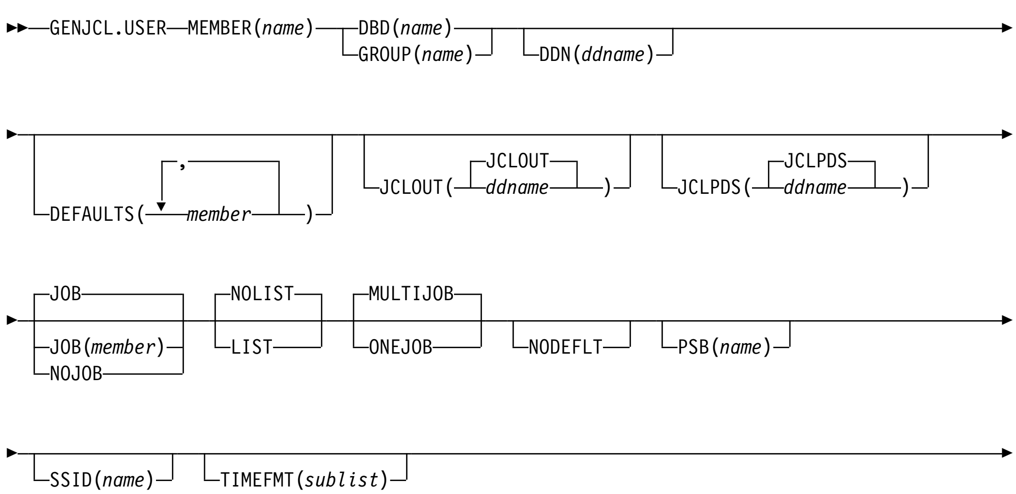
Example for all DBDSs in a Group

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



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

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 “Using the Commands to Generate JCL and User-Defined Output” on page 293.

Parameters

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 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.

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.

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(<u>JCLOUT</u> | *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 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 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 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.

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 Sublist” on page 95.
- For information on the precedence of the subparameters, see “TIMEFMT Subparameter Order of Precedence” on page 97.
- See “Standard Time Stamp Format” on page 93 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 “Understanding Simple Keywords” on page 295.

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

INIT.ADS

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

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 `INIT.DBDS` 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).

Parameters

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 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

```
                                                      ┌─,──────┐
►►──INIT.CA──CADSN(name)──GRPNAME(name)──VOLLIST(─────▼─volser─┴─)──────────────►

►──┬──────────────────────────┬──┬──────────────────────────┬──────────────────►◄
   │          ┌─1─────┐       │  │       ┌─3400─────┐       │
   └─FILESEQ(─┴─value─┴─)─────┘  └─UNIT(─┴─unittype─┴─)─────┘
```

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.

Parameters

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 MVS 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.

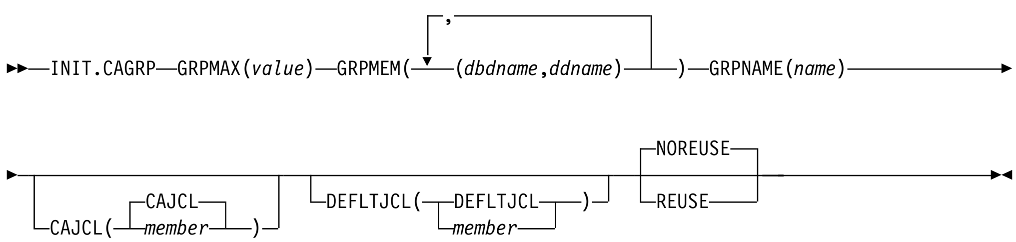
UNIT(<u>3400</u> | *unittype*)
Optional parameter you use to specify the unit type of the volumes on which the change accumulation data sets are to reside. The unit type can be up to eight alphanumeric characters.

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.

Parameters

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 2000 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.

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 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

```
                              ┌─RECOVABL──┐                          ┌─TYPEIMS─┐
►►──INIT.DB──DBD(name)────────┼───────────┼──┬──────────────────────┬─┼─────────┼──────────►
                              └─NONRECOV──┘  │           ┌─0─┐      │ └─TYPEFP──┘
                                             └─SHARELVL(─┼─1─┼─)────┘
                                                         ├─2─┤
                                                         └─3─┘

                               ┌─DBTRACK──┐
►──┬─────────────────────┬─────┼──────────┼────────────────────────────────────────────►◄
   └─GSGNAME(gsgname)────┘     └─RCVTRACK─┘
```

Use an `INIT.DB` command to create a database record in RECON and register the database with DBRC. The database must be registered in RECON before you can initialize a new DBDS or DEDB area with the `INIT.DBDS` command. You can also use the `INIT.DB` command to specify the level of database sharing.

Parameters

DBD(*name*)
Required parameter you use to specify the database name of the database to be registered in RECON.

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.

NONRECOV | RECOVABL
Mutually exclusive, optional parameters used to specify whether DBRC can record the updates in the RECON for the data sets of the specified databases. These parameters are meaningful for TYPEIMS databases only, and they are rejected for TYPEFP databases.

If the database is registered as RECOVABL, 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 data sets, are deleted at job termination, so they are not usable for recovery.

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.

Restrictions:
- NONRECOV cannot be specified if GSGNAME is specified.
- Fast Path DEDBs are never registered as nonrecoverable.
- You cannot make concurrent image copies of nonrecoverable 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 20.

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/ESA Administration Guide: System*.

TYPEFP | TYPEIMS
Mutually exclusive, optional parameters you use to specify whether the database is a Fast Path DEDB or a DL/I database.

TYPEFP
Specifies that the database is a Fast Path DEDB.

TYPEIMS
Specifies that the database is a DL/I database.

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

```
                                                                  ┌─DBTRACK─┐
►►──INIT.DBDS──DBD(name)──┬─DDN(name)──┬──DSN(name)──GENMAX(value)──┼─────────┼──┬──────────────────┬──►
                          └─AREA(name)─┘                              └─RCVTRACK─┘  └─DEFLTJCL(member)─┘

                                                     ┌─NOREUSE─┐                                      ┌─NOPREO──┐
►──┬──────────────────┬──┬──────────────────────┬──┼─────────┼──┬────────────────────────┬──┼─────────┼──►
   └─GSGNAME(gsgname)─┘  │        ┌─ICJCL──┐     │  └─REUSE───┘  │          ┌─OICJCL─┐     │  └─PREOPEN─┘
                         └─ICJCL(─┴─member─┴──)──┘               └─OICJCL(─┴─member─┴──)─┘

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

   ┌─NOVSO──────────────────────────────────────────────────────────────────┐
►──┼──────────────────────────────────────────────────────────────────────────┼──────────────►◄
   │                                                ┌─NOLKASID─┐  ┌─NOPREL──┐ │
   └─VSO──┬─────────────────┬──┬─────────────────┬──┼──────────┼──┼─────────┼─┘
          └─CFSTR1(name)────┘  └─CFSTR2(name)────┘  └─LKASID───┘  └─PRELOAD─┘
```

Use an `INIT.DBDS` command to initiate DBRC's control of the recovery of a DBDS or DEDB area. The `INIT.DBDS` command causes a DBDS record to be written in RECON. The DBDS record must exist for any of the other commands to work for a given DBDS or DEDB area. Before creating the DBDS record, 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 DBDS or DEDB area is in use.

Parameters

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 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 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 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 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:

For more information about the recovery period of image copy data sets, see *IMS/ESA Operations Guide*.

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 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 MVS 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.

CFSTR1(*name*)
Optional parameter you use to specify the name of the first coupling facility structure for the identified area. MVS 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. MVS 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, and the name cannot be the area name if the CFSTR1 keyword is not specified.

Related Reading: For details on CFSTR (coupling facility structure) naming conventions, see *IMS/ESA Administration Guide: Database Manager*.

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 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.

PRELOAD | NOPREL
Mutually exclusive, optional parameters 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 an MVS data space the next time it is opened. Selecting this option also causes the area to be pre-opened.

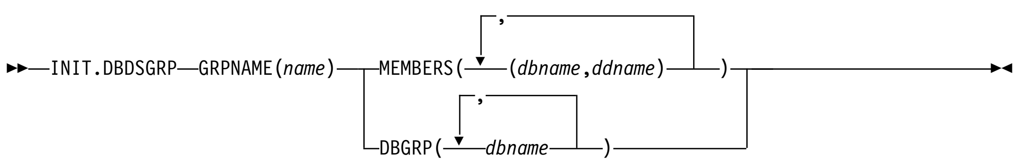
NOPREL
Indicates that the area is not to be loaded into an MVS data space the next time it is opened. For VSO areas, CIs are copied into a data space when they are read for the first time.

Example of Identifying the DBDS to Initiate DBRC's Control Over Recovery

In this example, a DBDS is identified to DBRC to initiate DBRC's control over the recovery of this DBDS. 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, its image copy data sets can be reused, and the maximum number of images that DBRC is to maintain for it 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 whose members are either:

- DBDSs or DEDB areas (DBDS group)
- databases or DEDB areas (DB group)

A single group cannot contain both types of members.

A DBDS group can be used anywhere a DB group can be used, such as for the /DBR command, but this usage is inefficient. It is preferable to define a separate DB group for such use.

Parameters

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***)**
Mutually exclusive, required parameters that identify the members to be included in the new group. A group can contain up to 2000 members. Any member can belong to more than one group.

MEMBERS(*dbname,ddname***)**
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.

DBGRP(*dbname***)**
Identifies one or more databases or area names.

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

```
►►—INIT.GSG—GSGNAME(gsgname)—┬────────────────┬——————————————————►◄
                             └─SEQNUM(number)─┘
```

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.

Parameters

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.

```
>>--INIT.IC--DBD(name)--+-DDN(name)--+--ICDSN(name)--+-----------------------+--+------------------------+-->
                        '-AREA(name)-'               |             .-1-----. |  |              .-1-----. |
                                                     '-FILESEQ(----+-value-+-)-'  '-FILESEQ2(----+-value-+-)-'

>--+---------------+--+----------------------+--+-----------------------+--+------------------------+-->
   '-ICDSN2(name)--'  |        .-3400-----.  |  |         .-3400-----.  |  |           .-,------.   |
                      '-UNIT(--+-unittype-+-)-'  '-UNIT2(--+-unittype-+-)-'  |           V        |   |
                                                                            '-VOLLIST(----volser-+-)-'

>--+-------------------------+--------------------------------------------------------------------><
   |            .-,------.   |
   |            V        |   |
   '-VOLLIST2(----volser-+-)-'
```

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.

Parameters

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.

DDN(*name*) | AREA(*name*)
Mutually exclusive, required parameters 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.

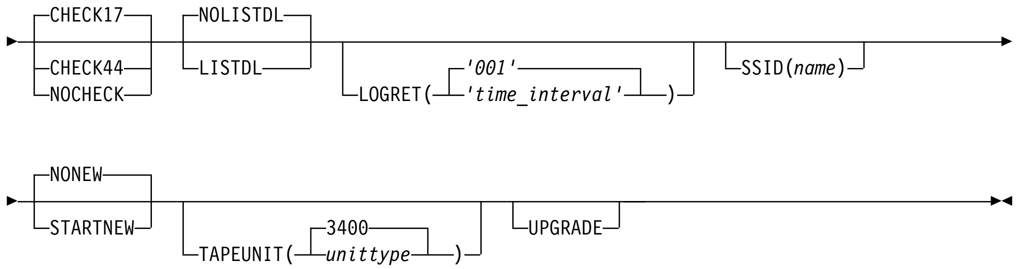
Example of Creating a Record that Defines the ICDSN

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.RECON

```
                    ┌─NOCATDS─┐  ┌─COEX───┐                                    ┌─NOFORCER─┐
►►──INIT.RECON──────┼─────────┼──┼────────┼──┬────────────────────────────┬──┼──────────┼──────►
                    └─CATDS───┘  └─NOCOEX─┘  │            ┌─3400─────┐    │  └─FORCER───┘
                                             └─DASDUNIT(──┴─unittype─┴─)──┘
```



Use an INIT.RECON command to cause the RECON header records that are to be written in both RECONs. The RECON header records must be the first records that you cause to be written in the RECONs, because these records identify the RECONs to DBRC's programs. A time history table (THT) is created each time `INIT.RECON` is issued. You also use the `INIT.RECON` command to define the default subsystem ID to be used with the following commands:

- `CHANGE.PRILOG`
- `CHANGE.SECLOG`
- `DELETE.LOG`
- `GENJCL.ARCHIVE`
- `GENJCL.CLOSE`
- `NOTIFY.PRILOG`
- `NOTIFY.SECLOG`

Related Reading: For information about the THT, see "RECON Header Records" on page 51.

Before writing the RECON header records with the `INIT.RECON` command, you must build the RECONs with the AMS `DEFINE` command. The RECONs must be empty.

Parameters

CATDS | NOCATDS

Mutually exclusive, optional parameters 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 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 that 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 of 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.

COEX | NOCOEX
Mutually exclusive, optional parameters you use to specify whether pre-version 6 and version 6 RECONs may coexist.

COEX
Enables the coexistence of pre-version 6 and version 6 RECONs, and imposes the same processing limitations on version 6 as on pre-version 6 when the local clock is changed (for example, as for daylight saving time).

Attention: `CHANGE.RECON COEX` allows you to reverse the setting of NOCOEX. There is no harm done as long as the local time has not changed since coexistence was disabled. Otherwise, serious malfunctions could occur in your pre-version 6 systems. IMS makes no attempt to protect you from this.

NOCOEX
Used if you never intend to run any pre-version 6 jobs against the RECON (including fall back to pre-version 6 after encountering a problem with version 6). NOCOEX disables coexistence so that there are no limitations on version 6 processing when the local clock is changed. You should be sure of the appropriateness of your decision to use the NOCOEX keyword. It is possible to reverse but see the Attention verbiage above.

DASDUNIT(3400 | ***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.

FORCER | NOFORCER
Mutually exclusive, optional parameters 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 need not be registered.

CHECK17 | CHECK44 | NOCHECK
Mutually exclusive, optional parameters 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 is 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(001 | *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 between a tenth of a second and 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('01 12:30')  = 10 days, 12 hours, 30 seconds
```

Related Reading:

- See “DELETE.LOG (for RLDS and SLDS)” on page 168 for more information on deleting inactive logs.
- See “Standard Time Stamp Format” on page 93 for more information on time stamps.

LISTDL | NOLISTDL
Mutually exclusive, optional parameters you use to specify whether data set names deleted from the RECON (by `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.

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 you use to specify whether new jobs are to be started when fewer than two RECONs are available.

STARTNEW
Specifies that new jobs are to be started.

NONEW
Specifies that new jobs are not to be started.

TAPEUNIT(3400 | *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.

UPGRADE
Optional parameter you use to specify that the RECON is to be upgraded for V6. It is required whenever a new Version 6 level RECON must be created for use by the upgrade utility. This parameter sets a safety switch that makes the new RECON unusable until the Upgrade utility is run. Likewise, the Upgrade utility requires that the switch has been set in the new RECON.

Related Reading: For more information on the upgrade process, see “Chapter 4. RECON Upgrade Utility (DSPURU00)” on page 63.

Example of Setting a No-Compare Flag

In this example, RECON header records are written in the two RECONs identified in the RECON1 and RECON2 DD statements. A flag is set in the header indicating that log data set names in RECON are not to be compared to the log data set names provided in the ddname of the utility JCL.

```
//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

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

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.

Parameters

ACTIVE | TRACKING
Mutually exclusive, required parameters 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 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

LIST.BKOUT

```
                  ┌─ALL──────────┐
►►──LIST.BKOUT────┼──────────────┼──┬──────────────────┬──────────────────────────────►◄
                  └─SSID(name)───┘  └─TIMEFMT(sublist)─┘
```

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 C. Sample Listings of RECONs” on page 347.

Parameters

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 Sublist” on page 95 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 93 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

```
           ┌─ALL──────────┐
►►──LIST.CAGRP─┼──────────────┼──┬──────────────────┬────────────────────►◄
           └─GRPNAME(name)─┘  └─TIMEFMT(sublist)─┘
```

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 C. Sample Listings of RECONs” on page 347.

Parameters

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*.

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 Sublist” on page 95 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 93 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

```
              ┌─ALL───────┐
►►──LIST.DB───┼───────────┼──┬─────────────────┬──┬──────┬──┬──────────────────┬──────────►◄
              ├─TYPEIMS───┤  ├─COVERED─────────┤  └─DBDS─┘  └─TIMEFMT(sublist)─┘
              ├─TYPEFP────┤  └─GSG(gsgname)────┘
              └─DBD(name)─┘
```

Use a `LIST.DB` command to receive a list of the database records 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 C. Sample Listings of RECONs” on page 347.

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

Parameters

ALL | TYPEIMS | TYPEFP | DBD(*name*)
Mutually exclusive, optional parameters you use to specify which database records in RECON are to be displayed.

ALL
Specifies that all database records in RECON are to be displayed. If you specify the DBDS parameter with this parameter, all database records and their associated DBDS and DEDB area records are to be displayed.

TYPEIMS
Specifies that all database records in RECON that describe a DL/I database are to be displayed. If you specify the DBDS parameter with this parameter, all database records for DL/I databases and their associated DBDS records are displayed.

TYPEFP
Specifies that all database records in RECON that describe a Fast Path DEDB are to be displayed. If you specify the DBDS parameter with this parameter, all database records for Fast Path DEDBs and their associated area records are displayed.

DBD(*name*)
Displays a specific 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. 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 Sublist" on page 95 for a description of the TIMEFMT parameter sublist format.
- See "Standard Time Stamp Format" on page 93 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

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

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 C. Sample Listings of RECONs" on page 347.

Parameters

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.

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.

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.

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 Sublist” on page 95 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 93 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

```
                    ┌─ALL───────────┐
►►──LIST.DBDSGRP────┼───────────────┼──┬──────────────────┬──────────────────►◄
                    └─GRPNAME(name)─┘  └─TIMEFMT(sublist)─┘
```

Use a `LIST.DBDSGRP` command to display either a list of :
- All DBDSGRP records in RECON.
- The members of a single DBDSGRP record.

Related Commands:
- “INIT.DBDSGRP” on page 221
- “CHANGE.DBDSGRP” on page 123

Parameters

GRPNAME(*name*) | ALL
Mutually exclusive, optional parameters you use to specify the groups to be listed.

GRPNAME
Produces a list of the members of the specified DBDS group. The named group must exist in RECON.

ALL
Produces a list of all the DBDS and DB groups identified in RECON.

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 Sublist” on page 95 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 93 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

```
             ┌─ALL─────────────────┐
►►──LIST.GSG─┼─────────────────────┼──┬──────────────────┬──────────────────►◄
             └─GSGNAME(gsgname)────┘  └─TIMEFMT(sublist)─┘
```

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 C. Sample Listings of RECONs” on page 347.

Parameters

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 Sublist” on page 95 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 93 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

```
►►──LIST.HISTORY──┬─DBD(name)───┬──┬───────────┬──┬───────────────────────┬──────►
                  └─GROUP(name)─┘  ├─DDN(name)─┤  └─FROMTIME(timestamp)─┘
                                   └─AREA(name)┘

►──┬───────────────────┬──┬──────────────────┬────────────────────────────────►◄
   └─TOTIME(timestamp)─┘  └─TIMEFMT(sublist)─┘
```

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 CA group record, if the DBDS is a member of a CA group.
- Available change accumulation data set records, if the DBDS is a member of a CA group.
- Available image copy data set records.
- The following records (if present), listed in time-sequence order:
  - DBDS allocation records and all PRILOG and SECLOG records associated with each ALLOC record
  - Image copy records
  - Change accumulation execution records
  - DBDS recovery records
  - DBDS reorganization records

Parameters

DBD(*name*) | GROUP(*name*)

Mutually exclusive, required parameters you use to identify the DBDS or DEDB area being list.

DBD

Specifies the database name of the DBDS or DEDB area to be listed.

GROUP
Specifies that all the DBDSs or DEDB areas of a DBDS 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.

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.

FROMTIME(*timestamp*)
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 93). 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(*timestamp*)
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 93). 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 Sublist" on page 95 for a description of the TIMEFMT parameter sublist format.

- See “Standard Time Stamp Format” on page 93 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 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)

```
►►──LIST.LOG──┬─────────────────────┬──┬────────────┬──┬──────────────┬──────────►
              └─STARTIME(timestamp)─┘  └─SSID(name)─┘  └─GSG(gsgname)─┘

►──┬─────────────────┬──────────────────────────────────────────────────────────►◄
   └─TIMEFMT(sublist)─┘
```

This command displays the PRILOG record and any of the following records having the specified start time.

- LOGALL
- SECLOG
- PRISLD
- SECSLD

Parameters

STARTIME(*timestamp*)
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*)
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.

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 Sublist” on page 95 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 93 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.

```
             ┌─ALL──┐
►►──LIST.LOG─┴──────┴─────────────────────────────────────────────►◄
```

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

```
             ┌─ALL──┐
►►──LIST.LOG─┴──────┴──INTERIM────────────────────────────────────►◄
```

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

```
             ┌─ALL──┐
►►──LIST.LOG─┴──────┴──TRACKING──────────────────────────────────►◄
```

- PRITSLDS
- SECTSLDS

```
             ┌─ALL──┐
►►──LIST.LOG─┴──────┴──TRACKING──INTERIM─────────────────────────►◄
```

- IPRITSLD
- ISECTSLD

```
►►──LIST.LOG──ALLOLDS────────────────────────────────────────────►◄
```

- PRIOLD
- SECOLD

```
►►──LIST.LOG──ALLOLDS──INTERIM───────────────────────────────────►◄
```

- IPRIOL
- ISECOL

```
►►──LIST.LOG──OLDS(ddname)──┬──────────────┬─────────────────────►◄
                            └─SSID(name)───┘
```

- 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 parameters for LIST.LOG ALL, ALLOLDS, and OLDS are:

```
►►──┬─────────────────────┬──┬───────────────────┬──SSID(name)──┬──────────────┬──►
    └─FROMTIME(timestamp)─┘  └─TOTIME(timestamp)─┘              └─GSG(gsgname)─┘

►──┬──────┬──┬───────┬──┬────────┬───────────────────────────────►◄
   └─OPEN─┘  └─ERROR─┘  └─UNARCH─┘
```

The 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.

Parameters

FROMTIME(*timestamp*)
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 93).

TOTIME(*timestamp***)**
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 93).

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 Sublist" on page 95 for a description of the TIMEFMT parameter sublist format.
- See "Standard Time Stamp Format" on page 93 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

```
►►──LIST.RECON──┬────────┬──┬──────────────────┬──────────────────────►◄
                └─STATUS─┘  └─TIMEFMT(sublist)─┘
```

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.
- Whether coexistence of this RECON with pre-V6 IMS is enabled.
- If coexistence is enabled, the contents of the Time History Table.
- The status of each of the three RECONs.

RECON status	Meaning
COPY1	PRIMARY ACTIVE RECON
COPY2	BACKUP RECON
SPARE	AVAILABLE RECON
UNAVAILABLE	UNAVAILABLE RECON
DISCARDED	UNUSABLE RECONS

The RECON is unavailable when the resource is allocated in another system.

Parameters

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 Sublist" on page 95 for a description of the TIMEFMT parameter sublist format.
- See "Standard Time Stamp Format" on page 93 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 C. Sample Listings of RECONs" on page 347.

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 C. Sample Listings of RECONs" on page 347 is produced in this case.

Related Reading: For an explanation of the possible RECON states, see "Chapter 3. Initializing and Maintaining the RECON" on page 43.

LIST.SUBSYS

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

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 C. Sample Listings of RECONs” on page 347.

Parameters

SSID(*name*) | ALL | <u>ONLINE</u>| 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.

<u>Related Reading:</u>

- See “TIMEFMT Parameter Sublist” on page 95 for a description of the TIMEFMT parameter sublist format.
- See “Standard Time Stamp Format” on page 93 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

NOTIFY.ALLOC

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

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

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 or in use.

Parameters

ALLTIME(*timestamp***)**
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 93).

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.

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(*timestamp***) | STARTIME(***timestamp***)**
Mutually exclusive, required parameters. The time stamp must be in standard form (see "Standard Time Stamp Format" on page 93).

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

```
►►──NOTIFY.BKOUT──SSID(name)──UOR(uor)──UORTIME(timestamp)──PSB(name)──────────────►

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

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.

Parameters

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(*timestamp***)**
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 93).

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.

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(E2E8E2F3404040400000000600000003)
                   UORTIME(930931345027) PSB(APPL34)
                   DBD(DATA1,DATA2,DATA3C)
                   BKO(DATA4,DATA5,DATA3A)
/*
```

NOTIFY.CA

```
►►──NOTIFY.CA──CADSN(name)──GRPNAME(name)──RUNTIME(timestamp)──STOPTIME(timestamp)──────────►

           ┌─,────────┐
           ▼          │
►─VOLLIST(───volser───┴──)──┬──────────────────────────┬──────────────────────────────────►
                            │               ┌─1─────┐  │
                            └─FILESEQ(──────┴─value─┴──)─┘
```

```
►──┬────────────────────────────────────────────────────────────────────┬──────►
   │              ┌─,──────────────────────────────────────────────┐    │
   │              ▼                ┌─NO──┐        ┌─COMPLETE───┐    │    │
   └─PURGLIST(───(──timestamp,───┴─YES─┴────,───┴─INCOMPLETE─┴──)─┴──)─┘

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

Use a `NOTIFY.CA` command to add information about a run of the Database Change Accumulation utility for a specified CA group.

Parameters

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(*timestamp*)
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 93).

STOPTIME(*timestamp*)
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 93).

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(*timestamp*,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 93). 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 232 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

```
>>--NOTIFY.IC--DBD(name)--+-DDN(name)--+--ICDSN(name)--RUNTIME(timestamp)------------->
                          '-AREA(name)-'

                                                                     .-BATCH----.
>--+---------------------------+--+----------------------------+--+-----------------+--+----------+-->
   |            .-1-----.      |  |             .-1-----.      |  '-ICDSN2(name)-'  +-ONLINE---+
   '-FILESEQ(---+-value-+---)--'  '-FILESEQ2(---+-value-+---)--'                    +-CIC------+
                                                                                    +-SMSCIC---+
                                                                                    '-SMSNOCIC-'

>--+---------------+--+-------------------+--+-------------------------+------------->
   '-RECDCT(value)-'  '-STOPTIME(timestamp)-'  |        .-3400-----.     |
                                               '-UNIT(--+-unittype-+--)-'

>--+--------------------------+--+--------------------------+----------------------->
   |         .-3400-----.     |  |           .-,------.     |
   '-UNIT2(--+-unittype-+--)--'  |           V        |     |
                                 '-VOLLIST(----volser-+--)--'
```

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

Use a `NOTIFY.IC` command to add information run of the Database Image Copy utility for a RECON registered DBDS or DEDB area that is identified in the command.

Parameters

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.

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(*timestamp***)**
: 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 93).

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.

  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 may also be specified to record the output of the HISAM Reorganization Unload utility image copy. If the database or area is assigned to a GSG, you are entering the `NOTIFY.IC` command at the tracking site, and the image copy was taken

at the active site after a database reorganization, BATCH must be specified regardless of the true image copy type.

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(*timestamp***)**
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 93).

UNIT(<u>3400</u> | *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(<u>3400</u> | *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)

```
>>--NOTIFY.PRILOG--OLDS(ddname)--+-DSN(name)-----------+--STARTIME(timestamp)-------->
                                 '-RUNTIME(timestamp)--'

>--+-------------------+--+--------------+--+---------+--+------------------+------->
   '-FIRSTREC(number)--'  '-GSG(gsgname)-'  '-INTERIM-'  '-LASTREC(number)--'

>--+-------------------+--+-----------------+--+-----------+------------------------><
   '-LOGTOKEN(number)--'  '-NXTOLDS(ddname)-'  '-SSID(name)-'
```

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.

Parameters

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 258). *ddname* is the DD statement that the IMS online control region used when it used the OLDS.

DSN(*name*) | RUNTIME(*timestamp*)
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 93).

These two parameters are used in conjunction with the STARTIME and NXTOLDS parameters to identify what type of primary OLDS entry is to be added to RECON. Table 4 indicates which parameter combinations are required for each type of primary OLDS entry:

Table 4. 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(*timestamp***)**
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 93.

See Table 4 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(*timestamp*) 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)
/*
```

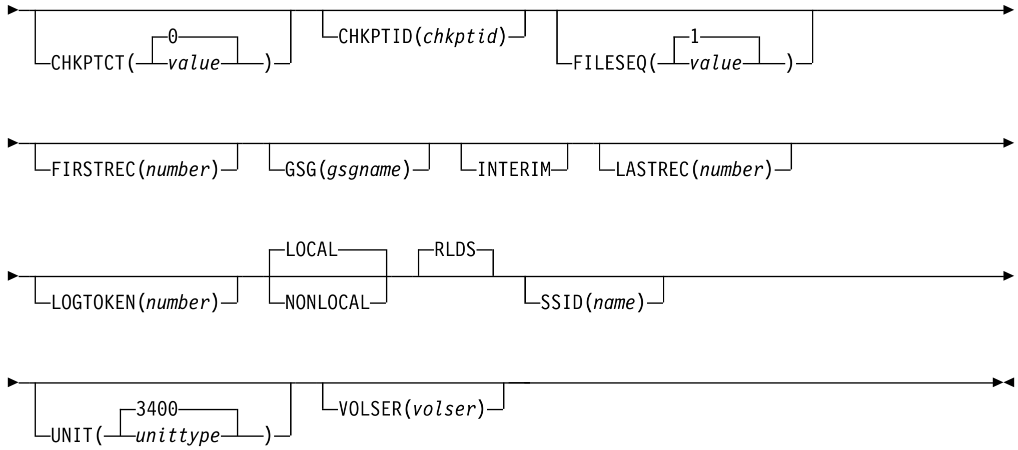
Example of Creating a PRILOG to Record 2 OLDSs Opening and Closing

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)

```
>>--NOTIFY.PRILOG--+-DSN(name)-------------+--STARTIME(timestamp)------------------------><
                   +-RUNTIME(timestamp)----+
```



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.

Parameters

DSN(*name*) | RUNTIME(*timestamp*)
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 93).

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 5 indicates which parameter combinations are required for each type of primary-recovery-log-data-set entry:

Table 5. 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

Table 5. Parameters of NOTIFY.PRILOG (for RLDS) Command for Open, EOV, and Close (continued)

Type of Log Entry	Required Keywords
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(*timestamp*)
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 93.

See Table 5 on page 259 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 93).

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)

```
►►──NOTIFY.PRILOG──┬─SLDS──┬──┬─DSN(name)────────────┬─STARTIME(timestamp)──────────────►
                   └─TSLDS─┘  └─RUNTIME(timestamp)───┘

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

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

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

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.

NOTIFY.PRILOG (for SLDS and TSLDS)

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(SLDS) or CHANGE.PRILOG(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.

Parameters

SLDS
Required parameter you use to specify that a record is to be created or updated for a SLDS.

Attention: If you do not specify SLDS or TSLDS, the default is RLDS (See “NOTIFY.PRILOG (for RLDS)” on page 258).

TSLDS
Required parameter you use to specify that a record is to be created or updated for a SLDS on an RSR tracking subsystem.

Attention: If you do not specify SLDS or TSLDS, the default is RLDS (See “NOTIFY.PRILOG (for RLDS)” on page 258)

DSN(*name*) | RUNTIME(*timestamp*)
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 93).

These two parameters are used in conjunction with the STARTIME and VOLSER parameters to identify what type of primary-system-log-data-set entry is to be added to RECON. Table 6 indicates which parameter combinations are required for each type of primary-system-log-data set entry.

Table 6. 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(*timestamp*)
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 93).

See Table 6 on page 264 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 93).

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

```
►►──NOTIFY.RECOV──DBD(name)──┬─DDN(name)──┬──┬──────────────────────┬──┬─────────────────┬──────►
                             └─AREA(name)─┘  └─RCVTIME(timestamp)───┘  └─RCVUSID(usid)───┘

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

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: For more information about recovery in a data sharing environment, see *IMS/ESA Operations Guide*.

Parameters

DBD(*name*)
Required parameter you use to specify the database name of the DBDS or DEDB area.

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(*timestamp*)
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 93).

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 timestamp recovery.

TRACK
Specifies that the recovery was a track recovery.

If you specify this parameter and the RCVTIME parameter, the command fails.

RCVTIME(*timestamp*)
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.

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

See 200 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.

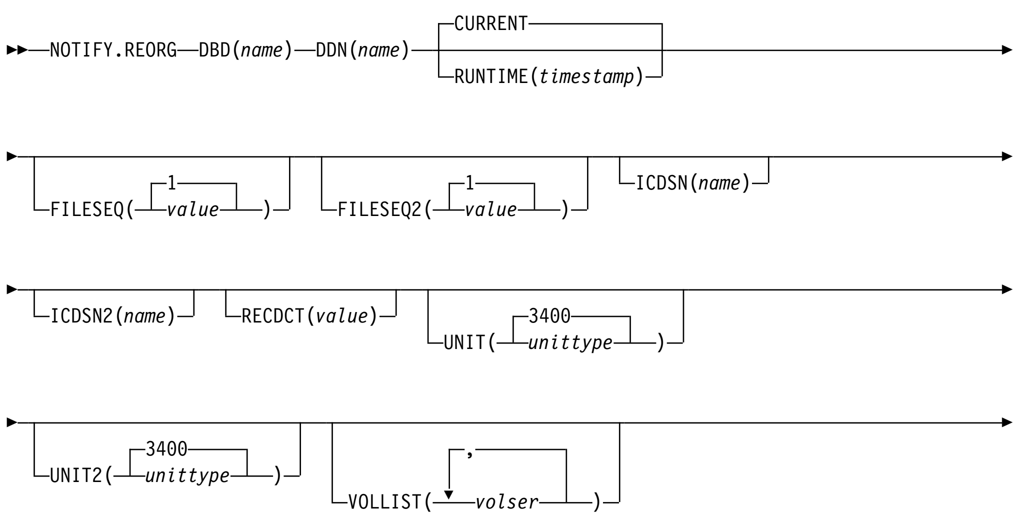
RUNUSID must be specified for recovery of an RSR-covered database or DEDB area.

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 RCVTIME parameter specifies the time stamp to which the specified DBDS was recovered.

```
//NFYRECOV JOB
 .
 .
 .
//SYSIN    DD   *
       NOTIFY.RECOV   DBD(DB1)  DDN(DDN1)  -
                      RUNTIME(820670201010)  -
                      RCVTIME(840220308200)
/*
```

NOTIFY.REORG



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

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 data bases can be recovered across 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.

Parameters

DBD(*name***)**
: Required parameter you use to identify the database name of the DBDS that was reorganized.

DDN(*name***)**
: Required parameter you use to identify the data set ddname of the DBDS that was reorganized.

CURRENT | RUNTIME(*timestamp***)**
: 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 93).

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.

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)

```
►►──NOTIFY.SECLOG──OLDS(ddname)──┬─DSN(name)───────────┬──STARTIME(timestamp)──────────────►
                                 └─RUNTIME(timestamp)──┘

►──┬───────────────────┬──┬───────────────┬──┬─────────┬──┬──────────────────┬──────────────►
   └─FIRSTREC(number)──┘  └─GSG(gsgname)──┘  └─INTERIM─┘  └─LASTREC(number)──┘

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

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.

Parameters

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(*timestamp*)
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 93).

DSN and RUNTIME are used in conjunction with the STARTIME and NXTOLDS parameters to identify which type of secondary-online-log-data-set entry is to be added to RECON. Table 7 indicates the keyword combinations that correspond to the type of secondary-online-log data-set entry:

Table 7. 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(*timestamp***)**
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 93).

See Table 7 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)

```
>>--NOTIFY.SECLOG--+-DSN(name)-----------+--STARTIME(timestamp)--------------->
                   +-RUNTIME(timestamp)--+

>--+---------------------------+--+------------------+--+--------------------------+-->
   |            +-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-+-)-+
```

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.

Parameters

DSN(*name*) | RUNTIME(*timestamp*)
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 93).

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

Table 8. 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(*timestamp***)**
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 93).

If you issue a subsequent STARTIME parameter, that time is the start time of the volume. See Table 8 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 93).

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 8 on page 276 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)

```
►►──NOTIFY.SECLOG──┬─SLDS──┬──┬─DSN(name)──────────┬──STARTIME(timestamp)──────────►
                   └─TSLDS─┘  └─RUNTIME(timestamp)─┘

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

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

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

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 (SLDS)command or the CHANGE.SECLOG(TSLDS) command.

Parameters

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 258 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 258 for more information on the RLDS parameter.

DSN(*name*) | RUNTIME(*timestamp*)
: 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 93).

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

Table 9. 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(*timestamp*)
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 93).

See Table 9 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 93).

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(<u>3400</u> | *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 9 on page 280 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

```
                                                                           ┌─NORMAL───┐
►►──NOTIFY.SUBSYS──SSID(name)──┬──────────────────┬──┬────────────────┬──┤          ├──────────►
                               └─BACKIRLM(name)─┘  └─IRLMID(name)─┘  └─ABNORMAL─┘
```

```
   ┌─ONLINE─┐  ┌─ENDRECOV─┐
►──┤        ├──┤          ├──────────────────────────────────────────────►◄
   └─BATCH──┘  └─STARTRCV─┘
```

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.

Parameters

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 MVS 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.

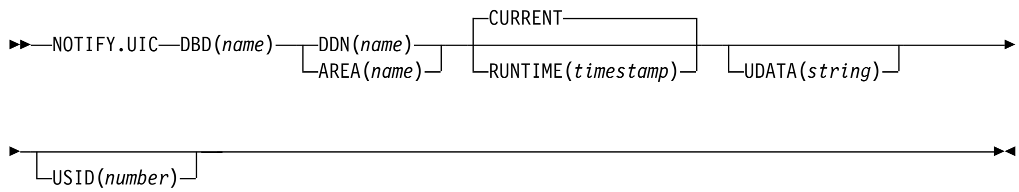
ENDRECOV
Specifies that the subsystem has signed on normally or that a sign-on recovery-complete call was successful.

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.

Restriction:

A nonstandard image copy data set cannot be used as input to the Database Recovery utility.

For more information, see 200.

Parameters

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.

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(*timestamp*)
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 93).

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. *string* must be in character format in order to be visible in a listing of RECON. It must be enclosed in parentheses.

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

RESET.GSG

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

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 MVS 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.

Parameters

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. Appendixes

Appendix A. Introduction

The following appendixes are reference materials that may be helpful to you in using DBRC.

In These Appendixes:

Appendix B. Understanding Skeletal JCL

Using the Commands to Generate JCL and User-Defined Output

Eight GENJCL commands are part of the DBRC 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 10 lists the eight GENJCL user commands and what they do.

Table 10. What the GENJCL Commands Do

Command (PDS Member)	What the Command Generates
GENJCL.ARCHIVE (ARCHJCL)	Log Archive utility JCL and control statements
GENJCL.CA (CAJCL)	Database Change Accumulation utility JCL and control statements
GENJCL.CLOSE (LOGCLJCL)	Log Recovery utility JCL and control statements
GENJCL.IC (ICJCL)	Database Image Copy or Database Image Copy 2 utility JCL and control statements
GENJCL.OIC (OICJCL)	Online Database Image Copy utility JCL and control statements
GENJCL.RECEIVE (ICRCVJCL)	Database Recovery utility JCL and control statements
GENJCL.RECOV (RECOVJCL)	Database Recovery utility JCL and control statements
GENJCL.USER (Not Provided)	User-defined output, including JCL and control statements

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 RECON 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.

DBRC verifies that the input to any of the utilities is correct according to the contents of RECON when the utility runs. This verification could identify errors you have made in writing your own JCL, or if any time has elapsed between the automatic generation of the JCL and its execution during which the contents of RECON changed, the verification step might also fail.

IBM supplies skeletal JCL execution members for all GENJCL commands except GENJCL.USER. IBM also 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” on page 294.

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 294.

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 323. 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 MVS 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 MVS 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

This section describes the things you need to know before writing your own skeletal JCL or developing simple keywords to modify the IBM-supplied skeletal JCL. You must write your own skeletal JCL or simple 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.

Understanding the Skeletal JCL Data Set

As shown in Figure 16 on page 295, 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 the previous section) 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 (explained in detail later).

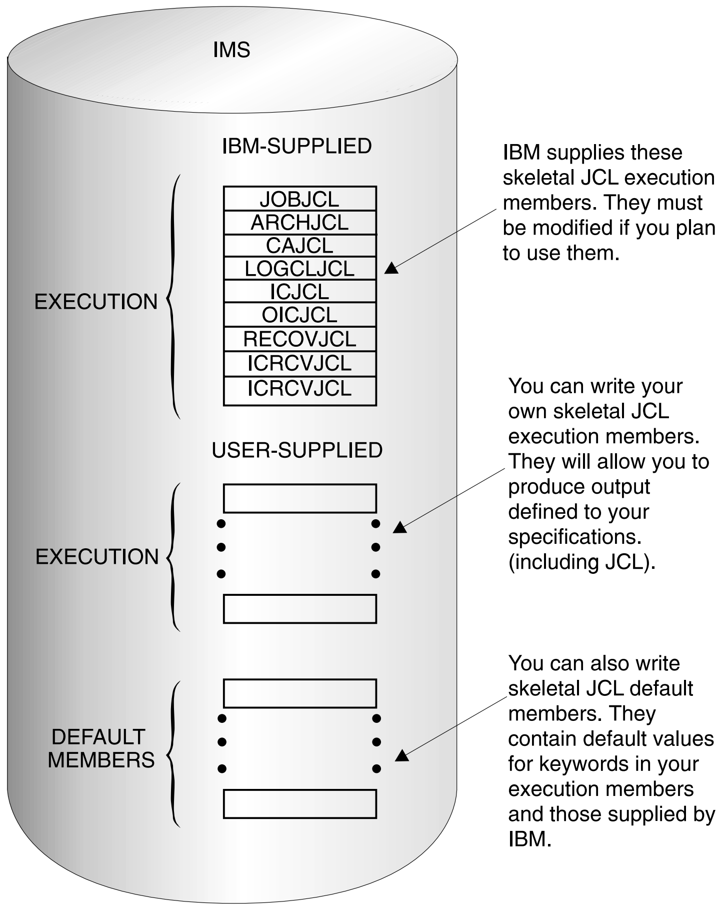


Figure 16. Skeletal JCL Data Set

Understanding Skeletal JCL Syntax

Skeletal JCL execution members are models of the output to be produced. Typically, execution members contain symbolic keywords. The symbolic keywords represent information that DBRC is to provide. The two types of symbolic keywords are simple keywords and control keywords. Simple keywords are keywords for which DBRC substitutes a value in the output stream (for example, data set names). Control keywords control what output is generated (for example, what RECON records are to be used for keyword substitution).

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

The keywords in skeletal JCL members must be in uppercase.

Understanding Simple Keywords

When JCL is generated, simple keywords in the skeletal JCL execution members are replaced with the current keyword value. For example, the IBM-supplied skeletal

Skeletal JCL Syntax

JCL execution members use %TIME as a simple keyword. When DBRC encounters %TIME, it replaces it with the time of day. Keyword substitution occurs each time DBRC encounters a simple keyword. Multiple simple keywords can exist in a skeletal JCL execution member.

Simple keywords must be assigned a value before you use them. Keyword values are assigned (or set) in several different ways, as shown in Figure 17 on page 297.

- The GENJCL command specifies values for some of the simple 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 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 simple keywords as well as use the simple keywords already recognized by DBRC. For a list of the simple keywords that DBRC recognizes, see "Understanding the Symbolic Keywords Recognized by DBRC" on page 315. You can also define your own simple 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 simple 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 simple 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.

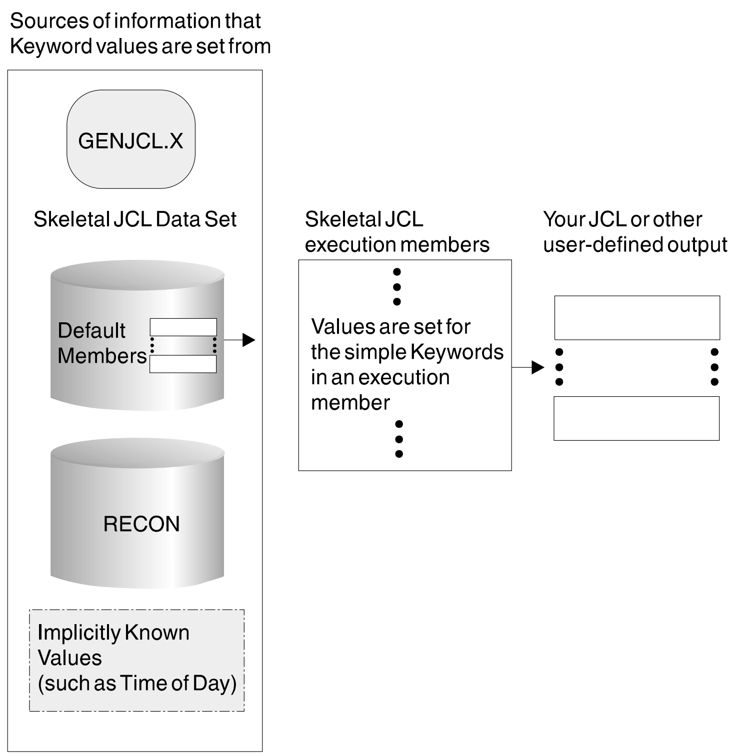


Figure 17. How Simple Keyword Values Are Set

Using Control Keywords

Use control keywords to regulate what JCL (or other output) is generated. The control keywords are:

```
%SELECT
%ENDSEL
%DELETE
%ENDDEL
%SET MEMBER
%SET TIMEFMT
```

The `%SELECT` keyword selects the RECON records that are needed in order to resolve simple 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.

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.

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

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 RECON. The selected records identify IMS data sets or events tracked by DBRC. Information from the selected records is used to resolve simple keywords in the select group. Simple keywords can occur in the *execution member records* or in the %SELECT keyword itself.

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 simple keywords.

As RECON records are selected, information from them is used to set the values of simple keywords. The keyword set depend on the type of record being selected and are described in following sections.

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 simple 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 11.

Table 11. 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 RECON 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 RECON 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 93 in “Chapter 7. DBRC Commands” on page 91.

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 simple 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 93 in "Chapter 7. DBRC Commands" on page 91.
- `%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 multi-step 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.

For GENJCL.USER, the default is:

```
TIMEFMT(O,O,C,2,1)
```

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 are unpredictable.

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 subsequent 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 Sublist” on page 95 but 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 simple 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 307.

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 the section "Understanding the Selection Criteria Parameter" on page 300.

In the execution member records following the %SELECT keyword, you specify (using simple 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}.

%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 first log record of the SLDS.

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 in one of the following ways:

```
►►─%SELECT─RLDS(ssid,time_qualifier──┬──────────────┬─)──────────────►◄
                                     └─,max_volumes─┘
```

```
►►─%SELECT─SRLDS(ssid,time_qualifier──┬──────────────┬─)─────────────►◄
                                      └─,max_volumes─┘
```

```
►►─%SELECT─RLDS(dbds_qualifier,time_qualifier──┬──────────────┬─)────►◄
                                               └─,max_volumes─┘
```

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 the section "Understanding the Selection Criteria Parameter" on page 300.

dbds_qualifier
DBDS qualifier as specified in the section "Understanding the Selection Criteria Parameter" on page 300. 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 RECON.) 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 simple 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.

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 RECON 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 300.

In the execution member records following the `%SELECT` keyword, you specify (using simple 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 the section “Understanding the Selection Criteria Parameter” on page 300.

In the execution member records following the %SELECT keyword, you use simple 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 the section “Understanding the Selection Criteria Parameter” on page 300. 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 simple 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 the section “Understanding the Selection Criteria Parameter” on page 300. 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 simple 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 for DEDB areas and DL/I for other types of databases.

%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 RECON. 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 MVS 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.

Understanding the Symbolic Keywords Recognized by DBRC

Table 12 through Table 18 on page 322 show the symbolic keywords that DBRC recognizes in the IBM-supplied skeletal JCL execution members.

Table 12 shows the keywords that all supported utilities recognize.

Table 13 on page 316 through Table 18 on page 322 show the keywords that each individual utility recognizes.

These figures and the figures in "Understanding the IBM-Supplied Skeletal JCL Execution Members" on page 323 are presented in the order of the `GENJCL` commands:

- `GENJCL.ARCHIVE` (Log Archive utility; see "GENJCL.ARCHIVE" on page 175 for details of the command usage.)
- `GENJCL.CA` (Database Change Accumulation utility; see "GENJCL.CA" on page 179 for details of the command usage.)
- `GENJCL.CLOSE` (Log Recovery utility; see "GENJCL.CLOSE" on page 183 for details of the command usage.)
- `GENJCL.IC` and `GENJCL.OIC` (Database Image Copy utility, and Database Image Copy 2 utility; Online Database Image Copy utility; see "GENJCL.IC" on page 186 and "GENJCL.OIC" on page 192 for details of the commands usage.)
- `GENJCL.RECEIVE` (Database Recovery utility; see "GENJCL.RECEIVE" on page 196 for details of the command usage.)
- `GENJCL.RECOV` (Database Recovery utility; see "GENJCL.RECOV" on page 200 for details of the command usage.)

All Supported Utilities

Table 12 explains the symbolic keywords recognized by all the supported utilities.

Table 12. Symbolic Keywords in Skeletal JCL

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.
`%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.

Table 12. Symbolic Keywords in Skeletal JCL (continued)

Keyword	Description
%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 13 explains the symbolic keywords recognized by the Log Archive utility.

Table 13. 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.
%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

Table 13. Symbolic Keywords for Log Archive Utility (continued)

Keyword	Description
%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 14 explains the symbolic keywords recognized by the Database Change Accumulation utility.

Table 14. 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.
%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.

Table 14. Symbolic Keywords for Database Change Accumulation Utility (continued)

Keyword	Description
%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 15 explains the symbolic keywords recognized by the Log Recovery utility.

Table 15. 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 16 on page 319 explains the symbolic keywords recognized by the Database Image Copy utilities.

Table 16. Symbolic Keywords for Database Image Copy and Online Database Image Copy Utility

Keyword	Description
%PSB	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.
%DBNAME	The database name, which is set from the DBD parameter on the GENJCL command.
%DBDDN	The DBDS ddname, which is set from the DDN parameter on the GENJCL command.
%DBDSN	The DBDS data set name, which is set from the DBDS record in RECON.
%DBDSAM	This keyword is set to VSAM for VSAM DBDS. Otherwise, it is set to null.
%DBADDN	For DEDBs, the ddname of the ADS. Otherwise, set to null. This keyword is applicable only for the Database Image Copy utility.
%DBADSAV	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.
%COPIES	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.
%SMS	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.
%ICSYSIN	The Database Image Copy utility control statement. Columns in the statement are set as follows: **Column** **Setting** **1** D **2** Number of image copy data sets to be produced (either 1 or 2) **4-11** Database name **13-20** ddname of the DBDS **22-30** ddname of the primary image copy data set **31-38** ddname of the duplicate image copy data set, if one is produced. **40-43** Checkpoint interval (applicable only for Online Database Image Copy utility). All other columns are set to blanks.
%ICDSN1, %ICDSN2%ICDSN3%ICDSN4	%ICDSN1 is set to the data set name of the image copy data set. 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 multiple image copy data sets are to be produced, %ICDSN2, %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.

Table 16. Symbolic Keywords for Database Image Copy and Online Database Image Copy Utility (continued)

Keyword	Description
%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.

Database Recovery Utility- Receive (ICRCVJCL)

Table 17 explains the symbolic keywords recognized by the Database Recovery (Receive) utility.

Table 17. Symbolic Keywords for Database Recovery 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.
%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 17. Symbolic Keywords for Database Recovery 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 17. Symbolic Keywords for Database Recovery 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-Recover (RECOVJCL)

Table 18 explains the symbolic keywords recognized by the Database Recovery (Recover) utility.

Table 18. Symbolic Keywords for Database Recovery Utility

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.
%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.
%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.
%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.

Table 18. Symbolic Keywords for Database Recovery Utility (continued)

Keyword	Description
%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 294.

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) it took for the GENJCL command to be executed.

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.

A description of the statements in ARCHJCL follows Figure 18.

```
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 18. 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 18. 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 18. 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.

Restriction: The `%ARVERS` keyword is not supported for the SLDS archive process and must not be used.

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.

A description of the statements in CAJCL follows Figure 19 on page 330.

```
//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    (%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
//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 19. 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. Each generated DB0 control statement has:

columns 1-3 DB0

columns 4-11 The database name

columns 12-20 The image-copy time stamp (in the form yydddhhmm)

columns 21-28 The database ddname

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.

A description of the statements in LOGCLJCL follows Figure 20 on page 333.

```
//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           (%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
%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 20. 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 DFSWADSn. n+1 is the number of WADSs that the online IMS control region uses. The WADS DD statements are contained in a select group that is controlled by the keyword %WADS. The GENJCL.CLOSE command processor sets the value of the %WADS keyword to YES if the OLDS is to be closed using the WADS. The command processor sets the value to NO if the OLDS is to be closed using the next OLDS. The WADS DD statements are, therefore, deleted if the OLDS is to be closed using the next OLDS.

Next OLDS DD Statements
If the OLDS is to be closed using the next OLDS, these DD statements identify the next OLDSs. These statements are contained in a delete group that is controlled by the %WADS keyword. Thus, if the OLDS is to be closed using the WADS, these statements are deleted. A select group is used in order to select the next OLDSs. The %SSID keyword identifies the subsystem ID. The %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. You can specify an execution member other than ICJCL by using the ICJCL parameter on the INIT.DBDS or CHANGE.DBDS commands.

A description of the statements in ICJCL follows Figure 21 on page 335.

```
//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
%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
%SELECT  DBDS((%DBNAME,%DBDDN))
%DELETE  (%DBADSAV NE 'AVAIL')
//%DBADDN   DD DSN=%DBDSN,DISP=%CICDISP
%ENDDEL
%DELETE  (%DBADSAV NE '')
//%DBDDN    DD DSN=%DBDSN,DISP=%CICDISP
%ENDDEL
%ENDSEL
//DATAOUT1  DD DSN=%ICDSN1,UNIT=%ICUNIT1,
//             VOL=(PRIVATE,,,%ICVCNT1,SER=(%ICVOLS1)),
//             LABEL=(%ICFSEQ1,SL),
%DELETE (%SMS EQ '1')
//             DISP=(NEW,KEEP),DCB=BUFNO=10
%ENDDEL
%DELETE (%SMS NE '1')
//             DISP=(NEW,KEEP)
%ENDDEL
```

Figure 21. IBM-Supplied Skeletal JCL for the Database Image Copy Utility (Part 1 of 2)

```
%DELETE  (%COPIES EQ '1' | %SMS EQ '1')
//DATAOUT2  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')
//DATAOUT2  DD DSN=%ICDSN2,UNIT=%ICUNIT2,
//             VOL=(PRIVATE,,,%ICVCNT2,SER=(%ICVOLS2)),
//             LABEL=(%ICFSEQ2,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%COPIES LE '2' | %SMS EQ '1')
//DATAOUT3  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')
//DATAOUT3  DD DSN=%ICDSN3,UNIT=%ICUNIT3,
//             VOL=(PRIVATE,,,%ICVCNT3,SER=(%ICVOLS3)),
//             LABEL=(%ICFSEQ3,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%COPIES LE '3' | %SMS EQ '1')
//DATAOUT4  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')
//DATAOUT4  DD DSN=%ICDSN4,UNIT=%ICUNIT4,
//             VOL=(PRIVATE,,,%ICVCNT4,SER=(%ICVOLS4)),
//             LABEL=(%ICFSEQ4,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
//SYSIN     DD *
%ICSYSIN
/*
```

Figure 21. IBM-Supplied Skeletal JCL for the Database Image Copy 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.

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.

DATAOUT1 DD Statement
This statement identifies 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

DATAOUT2 | 3 | 4 DD Statement
These statement identifies 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 `%DATAOUT1` 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.

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.

A description of the statements in OICJCL follows Figure 22 on page 338.

```
//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/ESA DATABASE RECOVERY CONTROL FEATURE.
//*
//*                JCL FOR ONLINE IMAGE COPY
//*
//STEPLIB       DD  DSN=IMS.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=IMS.DBDLIB,DISP=SHR
//DATAOUT1 DD  DSN=%ICDSN1,UNIT=%ICUNIT1,
//             VOL=(PRIVATE,,,%ICVCNT1,SER=(%ICVOLS1)),
//             LABEL=(%ICFSEQ1,SL),
//             DISP=(NEW,KEEP)
%DELETE        (%COPIES, EQ '1')
//DATAOUT2 DD  DSN=%ICDSN2,UNIT=%ICUNIT2,
//             VOL=(PRIVATE,,,%ICVCNT2,SER=(%ICVOLS2)),
//             LABEL=(%ICFSEQ2,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
//DFSUCKPT DD  DSN=IMS.%DBNAME.%DBDDN.CHECKPT.IC%time,
//             UNIT=SYSDA,SPACE=(TRK,1),DISP=(NEW,CATLG)
//SYSIN    DD  *
%ICSYSIN
/*
```

Figure 22. 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.

DATAOUT1 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

DATAOUT2 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 %DATAOUT1 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.

A description of the statements in ICRCVJCL follows Figure 23.

```
//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 23. IBM-Supplied Skeletal JCL for the Database Recovery Utility

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.

A description of the statements in RECOVJCL follows Figure 24 on page 342.

```
//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 RECOVERY.
//*
//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' | %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 24. 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 24. 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. The %DBNAME keyword is replaced with the database name of the DBDS that is being recovered.

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 (timestamp 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
```

Appendix C. 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:

Figure 25 on page 348 begins a listing of a RECON from an active site in an RSR environment. Figure 35 on page 367 is a listing of a RECON from a tracking site in an RSR environment.

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,

    and RECOV records

To find the DSECTS defining the formats of the RECON records in DBRC in GENLIBB and MACLIB, run the generate job with `MACLIB=ALL`. The member names are:

DSPALLRC ALLOC record

DSPBKORC BACKOUT record

DSPCAGRC CAGRP record

DSPCHGRC CA record

DSPDBHRC DB record

DSPDGRC DBDSGRP record

DSPGSGRC Global Service Group record

DSPDSHRC DBDS record

DSPIMGRC IC record

Sample RECON Listing Active Site

DSPLGARC LOGALL record

DSPLOGRC PRILOG / SECLOG record

DSPOLDRC PRIOLD / SECOLD record

DSPTHTRC THT record

DSPRCR1 HEADEREXT record

DSPRCNRC HEADER record

DSPRCVRC RECOV record

DSPRRGRC REORG record

DSPSLDRC PRISLD / SECSLD / PRITSLDS / SECTSLDS record

DSPSSRC SUBSYS record

Sample Listing of a RECON at the Active Site

Beginning with Figure 25, the following figures comprise a listing of a RECON from an active site in an RSR environment. "Fields Present in a Listing of a RECON by Record Type" on page 376 describes the fields that can be present in a listing of the RECON.

RECON Status Record

```
        IMS/ESA VERSION 6 RELEASE 1  DATA BASE RECOVERY CONTROL
 /* LIST IN LOCAL TIME                */
 LIST.RECON TIMEFMT(L,O,P,4)
1997.274 09:27:24.1 -09:00          LISTING OF RECON
--------------------------------------------------------------------------------
 RECON
  RECOVERY CONTROL DATA SET, IMS/ESA V6R1
  DMB#=9                             INIT TOKEN=97274F1724077F
  NOFORCER  LOG DSN CHECK=CHECK17    STARTNEW=NO
  TAPE UNIT=3400      DASD UNIT=SYSDA     TRACEOFF   SSID=**NULL**
  LIST DLOG=YES                CA/IC/LOG DATA SETS CATALOGED=NO
  LOG RETENTION PERIOD=00.000 00:15:00.0

  TIME STAMP INFORMATION:

    TIMEZIN = %SYS             -LABEL- -OFFSET-
                                PDT     -07:00
                                PST     -08:00

    OUTPUT FORMAT:  DEFAULT = LOCORG NONE   PUNC YY
                    CURRENT = LOCAL  OFFSET PUNC YYYY


 -DDNAME-      -STATUS-       -DATA SET NAME-
  RECON1        COPY1          IMSTESTL.IMS.RECON1
  RECON2        COPY2          IMSTESTL.IMS.RECON2
  RECON3        SPARE          IMSTESTL.IMS.RECON3
```

Figure 25. Sample Listing of a RECON at the Active Site - RECON Status

Log Records

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
--------------------------------------------------------------------------------
 PRILOG
  START = 1997.274 08:47:55.2 -09:00     *    SSID=SYS3     VERSION=6.1
  STOP  = 1997.274 09:08:18.6 -09:00          #DSN=4
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 2
  EARLIEST CHECKPOINT = 1997.274 08:51:02.4 -09:00

  DSN=**** COMPRESSED DATA SET ****                        UNIT=
  START = 1997.274 08:47:55.2 -09:00       FIRST DS LSN= 0000000000000001
  STOP  = 1997.274 08:49:27.4 -09:00       LAST  DS LSN= 00000000000002DA
  FILE SEQ=0000    #VOLUMES=0000

  DSN=IMSVS.RLDSP.SYS3.D97274.T0949274.V01                 UNIT=SYSDA
  START = 1997.274 08:49:27.4 -09:00       FIRST DS LSN= 00000000000002DB
  STOP  = 1997.274 08:50:33.2 -09:00       LAST  DS LSN= 000000000000048E
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 1997.274 08:50:33.2 -09:00
      CKPTCT=1    CHKPT ID = 1997.274 08:49:27.8 -09:00

  DSN=IMSVS.RLDSP.SYS3.D97274.T0950332.V00                 UNIT=SYSDA
  START = 1997.274 08:50:33.2 -09:00       FIRST DS LSN= 000000000000048F
  STOP  = 1997.274 08:51:15.3 -09:00       LAST  DS LSN= 000000000000058E
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 1997.274 08:51:15.3 -09:00
      CKPTCT=1    CHKPT ID = 1997.274 08:51:02.4 -09:00

  DSN=IMSVS.RLDSP.SYS3.D97274.T0951153.V01                 UNIT=SYSDA
  START = 1997.274 08:51:15.3 -09:00       FIRST DS LSN= 000000000000058F
  STOP  = 1997.274 09:08:18.6 -09:00       LAST  DS LSN= 00000000000007DF
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 1997.274 09:08:18.6 -09:00
      CKPTCT=2    CHKPT ID = 1997.274 09:08:17.1 -09:00

 LOGALL
  START   = 1997.274 08:47:55.2 -09:00     *
  DBDS ALLOC=1                       -DBD-    -DDN-    -ALLOC-
                                     DBVHDJ05 CJVHDG1E  1
```

Figure 26. Sample Listing of a RECON at the Active Site - Log Records (Part 1 of 7)

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
--------------------------------------------------------------------------------
 PRISLD
  START = 1997.274 08:47:55.2 -09:00     *    SSID=SYS3     VERSION=6.1
  STOP  = 1997.274 09:08:18.6 -09:00          #DSN=4
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 2

  DSN=**** COMPRESSED DATA SET ****                         UNIT=
  START = 1997.274 08:47:55.2 -09:00       FIRST DS LSN= 0000000000000001
  STOP  = 1997.274 08:49:27.4 -09:00       LAST  DS LSN= 00000000000002DA
  FILE SEQ=0000    #VOLUMES=0000

  DSN=IMSVS.SLDSP.SYS3.D97274.T0949274.V01                  UNIT=SYSDA
  START = 1997.274 08:49:27.4 -09:00       FIRST DS LSN= 00000000000002DB
  STOP  = 1997.274 08:50:33.2 -09:00       LAST  DS LSN= 000000000000048E
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 1997.274 08:50:33.2 -09:00
      CKPTCT=1    CHKPT ID = 1997.274 08:49:27.8 -09:00

  DSN=IMSVS.SLDSP.SYS3.D97274.T0950332.V00                  UNIT=SYSDA
  START = 1997.274 08:50:33.2 -09:00       FIRST DS LSN= 000000000000048F
  STOP  = 1997.274 08:51:15.3 -09:00       LAST  DS LSN= 000000000000058E
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 1997.274 08:51:15.3 -09:00
      CKPTCT=1    CHKPT ID = 1997.274 08:51:02.4 -09:00

  DSN=IMSVS.SLDSP.SYS3.D97274.T0951153.V01                  UNIT=SYSDA
  START = 1997.274 08:51:15.3 -09:00       FIRST DS LSN= 000000000000058F
  STOP  = 1997.274 09:08:18.6 -09:00       LAST  DS LSN= 00000000000007DF
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 1997.274 09:08:18.6 -09:00
      CKPTCT=2    CHKPT ID = 1997.274 09:08:17.1 -09:00
```

Figure 26. Sample Listing of a RECON at the Active Site - Log Records (Part 2 of 7)

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
-------------------------------------------------------------------------------
 PRILOG
  START = 1997.274 09:08:59.4 -09:00     *    SSID=BATCH1   VERSION=6.1
  STOP  = 1997.274 09:09:05.8 -09:00          #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 3

  DSN=BATCH1.UPDATEF.LOG                                   UNIT=SYSDA
  START = 1997.274 09:08:59.4 -09:00       FIRST DS LSN= 0000000000000001
  STOP  = 1997.274 09:09:05.8 -09:00       LAST  DS LSN= 000000000000009C
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 1997.274 09:09:05.8 -09:00
      CKPTCT=0    CHKPT ID = 0000.000 00:00:00.0 +00:00

 LOGALL
  START   = 1997.274 09:08:59.4 -09:00     *
  DBDS ALLOC=3                         -DBD-    -DDN-    -ALLOC-
                                       DHVNTZ02 HIDAM     1
                                       DXVNTZ02 XDLBT04I  1
                                       DIVNTZ02 DBHVSAM1  1

 SECLOG
  START = 1997.274 09:08:59.4 -09:00     *    SSID=BATCH1   VERSION=6.1
  STOP  = 1997.274 09:09:05.8 -09:00          #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 3

  DSN=IMSTESTL.IMS01.OLDSP1                                UNIT=SYSDA
  START = 1997.274 09:08:59.4 -09:00       FIRST DS LSN= 0000000000000001
  STOP  = 1997.274 09:09:05.8 -09:00       LAST  DS LSN= 000000000000009C
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=USER03 STOPTIME = 1997.274 09:09:05.8 -09:00
      CKPTCT=0    CHKPT ID = 0000.000 00:00:00.0 +00:00
```

Figure 26. Sample Listing of a RECON at the Active Site - Log Records (Part 3 of 7)

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
--------------------------------------------------------------------------------
 PRILOG
  START = 1997.274 09:12:08.5 -09:00     *    SSID=SYS3     VERSION=6.1
  STOP  = 1997.274 09:17:30.8 -09:00          #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 4
  EARLIEST CHECKPOINT = 1997.274 08:51:02.4 -09:00

  DSN=IMSVS.RLDSP.SYS3.D97274.T1012085.V02                 UNIT=SYSDA
  START = 1997.274 09:12:08.5 -09:00       FIRST DS LSN= 0000000000000001
  STOP  = 1997.274 09:17:30.8 -09:00       LAST  DS LSN= 0000000000000251
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 1997.274 09:17:30.8 -09:00
      CKPTCT=1    CHKPT ID = 1997.274 09:12:14.9 -09:00

 LOGALL
  START   = 1997.274 09:12:08.5 -09:00     *
  DBDS ALLOC=5                         -DBD-    -DDN-    -ALLOC-
                                       DEDBDD01 DD01AR0   1
                                       DHVNTZ02 HIDAM     1
                                       DIVNTZ02 DBHVSAM1  1
                                       DBOHIDK5 CKOHIG1O  1
                                       DBVHDJ05 CJVHDG1E  1

 PRISLD
  START = 1997.274 09:12:08.5 -09:00     *    SSID=SYS3     VERSION=6.1
  STOP  = 1997.274 09:17:30.8 -09:00          #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 4

  DSN=IMSVS.SLDSP.SYS3.D97274.T1012085.V02                 UNIT=SYSDA
  START = 1997.274 09:12:08.5 -09:00       FIRST DS LSN= 0000000000000001
  STOP  = 1997.274 09:17:30.8 -09:00       LAST  DS LSN= 0000000000000251
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 1997.274 09:17:30.8 -09:00
      CKPTCT=1    CHKPT ID = 1997.274 09:12:14.9 -09:00
```

Figure 26. Sample Listing of a RECON at the Active Site - Log Records (Part 4 of 7)

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
-------------------------------------------------------------------------------
 PRILOG
  START = 1997.274 09:17:31.2 -09:00     *    SSID=SYS3     VERSION=6.1
  STOP  = 0000.000 00:00:00.0 +00:00          #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000252       PRILOG TOKEN= 5
  EARLIEST CHECKPOINT = 1997.274 09:17:42.2 -09:00

  DSN=IMSVS.RLDSP.SYS3.D97274.T0917312.V01                 UNIT=SYSDA
  START = 1997.274 09:17:31.2 -09:00       FIRST DS LSN= 0000000000000252
  STOP  = 1997.274 09:20:26.3 -09:00       LAST  DS LSN= 00000000000003D4
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 1997.274 09:20:26.3 -09:00
      CKPTCT=2    CHKPT ID = 1997.274 09:17:42.2 -09:00

 LOGALL
  START   = 1997.274 09:17:31.2 -09:00     *
  DBDS ALLOC=1                        -DBD-    -DDN-    -ALLOC-
                                      DEDBDD01 DD01AR0   1
 PRISLD
  START = 1997.274 09:17:31.2 -09:00     *    SSID=SYS3     VERSION=6.1
  STOP  = 0000.000 00:00:00.0 +00:00          #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000252       PRILOG TOKEN= 5

  DSN=IMSVS.SLDSP.SYS3.D97274.T0917312.V01                 UNIT=SYSDA
  START = 1997.274 09:17:31.2 -09:00       FIRST DS LSN= 0000000000000252
  STOP  = 1997.274 09:20:26.3 -09:00       LAST  DS LSN= 00000000000003D4
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 1997.274 09:20:26.3 -09:00
      CKPTCT=2    CHKPT ID = 1997.274 09:17:42.2 -09:00
```

Figure 26. Sample Listing of a RECON at the Active Site - Log Records (Part 5 of 7)

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
-------------------------------------------------------------------------------
 PRILOG
  START = 1997.274 09:25:15.8 -09:00     *    SSID=IMS2     VERSION=6.1
  STOP  = 1997.274 09:25:58.5 -09:00          #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 6
  EARLIEST CHECKPOINT = 1997.274 09:25:23.0 -09:00

  DSN=IMSVS.RLDSP.IMS2.D97274.T0925158.V00                 UNIT=SYSDA
  START = 1997.274 09:25:15.8 -09:00       FIRST DS LSN= 0000000000000001
  STOP  = 1997.274 09:25:58.5 -09:00       LAST  DS LSN= 00000000000001B9
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 1997.274 09:25:58.5 -09:00
      CKPTCT=2    CHKPT ID = 1997.274 09:25:57.3 -09:00

 LOGALL
  START   = 1997.274 09:25:15.8 -09:00     *
  DBDS ALLOC=0
 PRISLD
  START = 1997.274 09:25:15.8 -09:00     *    SSID=IMS2     VERSION=6.1
  STOP  = 1997.274 09:25:58.5 -09:00          #DSN=1
  GSGNAME=IMSGSG1
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 6

  DSN=IMSVS.SLDSP.IMS2.D97274.T0925158.V00                 UNIT=SYSDA
  START = 1997.274 09:25:15.8 -09:00       FIRST DS LSN= 0000000000000001
  STOP  = 1997.274 09:25:58.5 -09:00       LAST  DS LSN= 00000000000001B9
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 1997.274 09:25:58.5 -09:00
      CKPTCT=2    CHKPT ID = 1997.274 09:25:57.3 -09:00

 PRIOLD
  SSID=IMS2            # DD ENTRIES=1
  EARLIEST CHECKPOINT = 1997.274 09:25:23.0 -09:00

  DDNAME=DFSOLP00   DSN=IMSTESTL.IMS02.OLDSP0
  START = 1997.274 09:25:15.8 -09:00       FIRST DS LSN= 0000000000000001
  STOP  = 1997.274 09:25:58.5 -09:00       LAST  DS LSN= 00000000000001B9
  STATUS=ARC COMPLT                                FEOV=NO    AVAIL
  PRILOG TIME=1997.274 09:25:15.8 -09:00       ARCHIVE JOB NAME=JT092559
  VERSION=6.1
```

Figure 26. Sample Listing of a RECON at the Active Site - Log Records (Part 6 of 7)

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
-------------------------------------------------------------------------------
 PRIOLD
  SSID=SYS3            # DD ENTRIES=4
  EARLIEST CHECKPOINT = 1997.274 09:17:42.2 -09:00

  DDNAME=DFSOLP03   DSN=IMSTESTL.IMS01.OLDSP3
  START = 1997.274 08:51:15.3 -09:00       FIRST DS LSN= 000000000000058F
  STOP  = 1997.274 09:08:17.3 -09:00       LAST  DS LSN= 0000000000000792
  STATUS=ARC COMPLT                                FEOV=NO    AVAIL
  PRILOG TIME=1997.274 08:47:55.2 -09:00       ARCHIVE JOB NAME=JT091734
  VERSION=6.1

  DDNAME=DFSOLP01   DSN=IMSTESTL.IMS01.OLDSP1
  START = 1997.274 09:12:08.5 -09:00       FIRST DS LSN= 0000000000000001
  STOP  = 1997.274 09:17:30.8 -09:00       LAST  DS LSN= 0000000000000251
  STATUS=ARC COMPLT                                FEOV=NO    AVAIL
  PRILOG TIME=1997.274 09:12:08.5 -09:00       ARCHIVE JOB NAME=JT091734
  VERSION=6.1

  DDNAME=DFSOLP02   DSN=IMSTESTL.IMS01.OLDSP2
  START = 1997.274 09:17:31.2 -09:00       FIRST DS LSN= 0000000000000252
  STOP  = 1997.274 09:20:26.3 -09:00       LAST  DS LSN= 00000000000003D4
  STATUS=ARC COMPLT                                FEOV=NO    AVAIL
  PRILOG TIME=1997.274 09:17:31.2 -09:00       ARCHIVE JOB NAME=JT092027
  VERSION=6.1

  DDNAME=DFSOLP00   DSN=IMSTESTL.IMS01.OLDSP0
  START = 1997.274 09:20:26.3 -09:00       FIRST DS LSN= 00000000000003D5
  STOP  = 0000.000 00:00:00.0 +00:00       LAST  DS LSN= 0000000000000000
  STATUS=ACTIVE                                    FEOV=NO    AVAIL
  PRILOG TIME=1997.274 09:17:31.2 -09:00
  VERSION=6.1
DSP0260I  NO INTERIM RLDS/SLDS RECORDS FOUND IN RECON
DSP0260I  NO INT-ONLINE LOG RECORDS FOUND IN RECON
```

Figure 26. Sample Listing of a RECON at the Active Site - Log Records (Part 7 of 7)

GSG Record

```
1997.274 09:27:24.1 -09:00          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 = 1997.274 09:25:15.8 -09:00
  HIGHEST ACTIVE SITE TIME  = 0000.000 00:00:00.0 +00:00
  TRACKING SUBSYSTEM ID     = **NULL**
  TAKEOVER IN PROGRESS
```

Figure 27. Sample Listing of a RECON at the Active Site - GSG Record

SSYS Record

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
--------------------------------------------------------------------------------
 SSYS
  SSID=SYS3      LOG START=1997.274 09:17:31.2 -09:00
  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=6.1
                                                    ENCODED
    -DBD-        -AREA-   -LEVEL-  -ACCESS INTENT-  -STATE-
    DEDBDD01     DD01AR0     1        UPDATE           6
```

Figure 28. Sample Listing of a RECON at the Active Site - SSYS Record

BACKOUT Record

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
--------------------------------------------------------------------------------
 BACKOUT
  SSID=SYS3      #UORS=2

    RECOVERY TOKEN=E2E8E2F340404040000000300000002
    TIME=1997.274 09:13:27.0 -09:00       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=E2E8E2F340404040000000400000000
    TIME=1997.274 09:13:28.0 -09:00       PSB=PSBEJK05
             INFLT   BMP   COLDEND
      ASSOCIATED DATA BASES=2

                   BACKED   DYN BKOUT
      -DBD-        -OUT -   -FAILURE-
      DBOHIDK5      NO         NO
      DXVHIDK5      NO         NO
```

Figure 29. Sample Listing of a RECON at the Active Site - BACKOUT Record

CAGRP and CA Records

```
1997.274 09:27:24.1 -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    = 1997.274 08:40:43.5 -09:00     *
                     UNIT=SYSDA      VOLS DEF=1    VOLS USED=1
                                     VOLSER=222222
  RUN     = 1997.274 09:26:19.2 -09:00
   DBD=DEDBJN21 DDN=DB21AR1  PURGETIME = 1997.274 08:38:21.4 -09:00
     CHANGES ACCUMULATED=YES COMPLETE CA=YES INDOUBT EEQES=NO
     LSN    = 000000000000      DSSN   = 0000000001
     LRID   = 0000000000000414  USID   = 0000000002
   DBD=DEDBJN21 DDN=DB21AR3  PURGETIME = 1997.274 08:38:29.2 -09:00
     CHANGES ACCUMULATED=YES COMPLETE CA=YES INDOUBT EEQES=NO
     LSN    = 000000000000      DSSN   = 0000000001
     LRID   = 0000000000000428  USID   = 0000000002
   DBD=DEDBJN21 DDN=DB21AR6  PURGETIME = 1997.274 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 = 1997.274 08:31:07.0 -09:00
     CHANGES ACCUMULATED=NO  COMPLETE CA=YES INDOUBT EEQES=NO
     LSN    = 000000000000      DSSN   = 0000000000
     LRID   = 0000000000000000  USID   = 0000000000
```

Figure 30. Sample Listing of a RECON at the Active Site - CAGRP and CARecords

DBDSGRP Records

```
1997.274 09:27:24.1 -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 31. Sample Listing of a RECON at the Active Site - DBDSGRP Records

DB (IMS) and Related Records

```
1997.274 09:27:24.1 -09:00          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
  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
```

Figure 32. Sample Listing of a RECON at the Active Site - DB (IMS) and Related Records (Part 1 of 2)

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
--------------------------------------------------------------------------------
 ALLOC
  ALLOC   =1997.274 08:50:23.5 -09:00     *  ALLOC LRID =0000000000000000
  DSSN=0000000004 USID=0000000005 START = 1997.274 08:47:55.2 -09:00
  DEALLOC =1997.274 08:51:14.1 -09:00      DEALLOC LRID =0000000000000000

 ALLOC
  ALLOC   =1997.274 09:14:29.7 -09:00     *  ALLOC LRID =0000000000000000
  DSSN=0000000005 USID=0000000006 START = 1997.274 09:12:08.5 -09:00

 IMAGE
  RUN     = 1997.274 08:49:59.2 -09:00     *  RECORD COUNT =31
  STOP    = 0000.000 00:00:00.0 +00:00        BATCH      USID=0000000004

 IC1
  DSN=IMSVS.DBVHDJ05.CJVHDG1E.IC.IC094956          FILE SEQ=0001
  UNIT=SYSDA                           VOLS DEF=0001 VOLS USED=0001
                                       VOLSER=222222



 IMAGE
  RUN     = 1997.274 08:52:02.7 -09:00     *  RECORD COUNT =31
  STOP    = 1997.274 08:52:19.4 -09:00        CONCUR     USID=0000000005

 IC1
  DSN=IMSVS.DBVHDJ05.CJVHDG1E.IC.IC095155          FILE SEQ=0001
  UNIT=SYSDA                           VOLS DEF=0001 VOLS USED=0001
                                       VOLSER=222222
```

Figure 32. Sample Listing of a RECON at the Active Site - DB (IMS) and Related Records (Part 2 of 2)

DB (FP) and Related Records

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
--------------------------------------------------------------------------------
 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
                                         EEQE COUNT              =0

 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
  DEFLTJCL=**NULL**  ICJCL=ICJCL    RECVJCL=ICRCVJCL  RECOVJCL=RECOVJCL
```

Figure 33. Sample Listing of a RECON at the Active Site - DB (FP) and Related Records (Part 1 of 2)

```
 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

1997.274 09:27:24.1 -09:00          LISTING OF RECON
-------------------------------------------------------------------------------
 ALLOC
  ALLOC   =1997.274 09:12:17.0 -09:00     *  ALLOC LRID =0000000000000000
  DSSN=0000000001 USID=0000000002 START = 1997.274 09:12:08.5 -09:00

 ALLOC
  ALLOC   =1997.274 09:17:51.9 -09:00     *  ALLOC LRID =0000000000000000
  DSSN=0000000002 USID=0000000003 START = 1997.274 09:17:31.2 -09:00

 IMAGE
  RUN     = 1996.100 07:15:11.2 -09:00     *
  STOP    = 1996.100 07:16:12.3 -09:00       SMSCIC     USID=0000000001

 IC1
  DSN=IMSVS.DEDBDD01.SMSCIC.DSN1                   FILE SEQ=0001
  UNIT=SYSDA                           VOLS DEF=0001 VOLS USED=0001
                                       VOLSER=IMSCC1
```

Figure 33. Sample Listing of a RECON at the Active Site - DB (FP) and Related Records (Part 2 of 2)

IMS DB Records

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
--------------------------------------------------------------------------------
 DB
  DBD=DHVNTZ02                 IRLMID=*NULL        DMB#=6        TYPE=IMS
  SHARE LEVEL=3               GSGNAME=IMSGSG1      USID=0000000003
  AUTHORIZED USID=0000000003  RECEIVE USID=0000000003 HARD USID=0000000003
  RECEIVE NEEDED USID=0000000000
  FLAGS:                             COUNTERS:
    BACKOUT NEEDED        =ON          RECOVERY NEEDED COUNT   =0
    READ ONLY             =OFF         IMAGE COPY NEEDED COUNT =0
    PROHIBIT AUTHORIZATION=OFF         AUTHORIZED SUBSYSTEMS   =1
    RECOVERABLE           =YES         HELD AUTHORIZATION STATE=0
    DATABASE LEVEL TRACK  =YES         EEQE COUNT              =0
    TRACKING SUSPENDED    =NO          RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =NO
```

Figure 34. Sample Listing of a RECON at the Active Site - More IMS DB Records (Part 1 of 5)

```
  ASSOCIATED SUBSYSTEM INFORMATION:
                            ENCODED  B/O NEEDED
    -SSID-   -ACCESS INTENT-  -STATE-   -COUNT-  -SS ROLE-
   SYS3                         0         1      ACTIVE

 DBDS
  DSN=DHVNTZ02.FKXXI01E                                        TYPE=IMS
  DBD=DHVNTZ02  DDN=HIDAM    DSID=001 DBORG=HIDAM  DSORG=VSAM

1997.274 09:27:24.1 -09:00          LISTING OF RECON
-------------------------------------------------------------------------------
 CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     DSSN=00000002
  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   =1997.274 09:09:02.5 -09:00     *  ALLOC LRID =0000000000000000
  DSSN=0000000001 USID=0000000002 START = 1997.274 09:08:59.4 -09:00

 ALLOC
  ALLOC   =1997.274 09:13:25.2 -09:00     *  ALLOC LRID =0000000000000000
  DSSN=0000000002 USID=0000000003 START = 1997.274 09:12:08.5 -09:00

 IMAGE
  RUN     = 1997.274 08:31:07.0 -09:00     *  RECORD COUNT =0
  STOP    = 0000.000 00:00:00.0 +00:00        BATCH      USID=0000000001

 IC1
  DSN=IMSTESTG.DHVNTZ02.HIDAM.BASE.IC              FILE SEQ=0001
  UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=IMSRAW



 DBDS
  DSN=DHVNTZ02.FKXXI02E                                        TYPE=IMS
  DBD=DHVNTZ02  DDN=HIDAM2   DSID=002 DBORG=HIDAM  DSORG=VSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     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


 IMAGE
  RUN     = 1997.274 08:31:07.0 -09:00     *  RECORD COUNT =0
  STOP    = 0000.000 00:00:00.0 +00:00        BATCH      USID=0000000001

 IC1
  DSN=IMSTESTG.DHVNTZ02.HIDAM2.BASE.IC             FILE SEQ=0001
  UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=IMSRAW
```

Figure 34. Sample Listing of a RECON at the Active Site - More IMS DB Records (Part 2 of 5)

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
--------------------------------------------------------------------------------
 DB
  DBD=DIVNTZ02                 IRLMID=*NULL        DMB#=5        TYPE=IMS
  SHARE LEVEL=3               GSGNAME=IMSGSG1      USID=0000000003
  AUTHORIZED USID=0000000003  RECEIVE USID=0000000003 HARD USID=0000000003
  RECEIVE NEEDED USID=0000000000
  FLAGS:                             COUNTERS:
    BACKOUT NEEDED        =ON          RECOVERY NEEDED COUNT   =0
    READ ONLY             =OFF         IMAGE COPY NEEDED COUNT =0
    PROHIBIT AUTHORIZATION=OFF         AUTHORIZED SUBSYSTEMS   =1
    RECOVERABLE           =YES         HELD AUTHORIZATION STATE=0
    DATABASE LEVEL TRACK  =YES         EEQE COUNT              =0
    TRACKING SUSPENDED    =NO          RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =NO

  ASSOCIATED SUBSYSTEM INFORMATION:
                               ENCODED  B/O NEEDED
     -SSID-   -ACCESS INTENT-  -STATE-   -COUNT-  -SS ROLE-
    SYS3                          0         1      ACTIVE

 DBDS
  DSN=DIVNTZ02.FJXXS01K                                          TYPE=IMS
  DBD=DIVNTZ02  DDN=DBHVSAM1 DSID=001 DBORG=HISAM  DSORG=VSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=2     DSSN=00000002
  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 34. Sample Listing of a RECON at the Active Site - More IMS DB Records (Part 3 of 5)

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
-------------------------------------------------------------------------------
 ALLOC
  ALLOC   =1997.274 09:09:04.5 -09:00     *  ALLOC LRID =0000000000000000
  DSSN=0000000001 USID=0000000002 START = 1997.274 09:08:59.4 -09:00

 ALLOC
  ALLOC   =1997.274 09:13:27.2 -09:00     *  ALLOC LRID =0000000000000000
  DSSN=0000000002 USID=0000000003 START = 1997.274 09:12:08.5 -09:00

 IMAGE
  RUN     = 1997.274 08:31:07.0 -09:00     *  RECORD COUNT =0
  STOP    = 0000.000 00:00:00.0 +00:00        BATCH      USID=0000000001

 IC1
  DSN=IMSTESTG.DIVNTZ02.DBHVSAM1.BASE.IC           FILE SEQ=0001
  UNIT=SYSDA                        VOLS DEF=0001 VOLS USED=0001
                                    VOLSER=IMSRAW



 IMAGE
  RUN     = 1997.274 09:12:39.9 -09:00     *  RECORD COUNT =106
  STOP    = 1997.274 09:12:43.6 -09:00        ONLINE     USID=0000000002

 IC1
  DSN=IMSVS.DIVNTZ02.DBHVSAM1.IC.IC101230          FILE SEQ=0001
  UNIT=SYSDA                        VOLS DEF=0001 VOLS USED=0001
                                    VOLSER=222222



 DBDS
  DSN=DIVNTZ02.FJXXS01E                                           TYPE=IMS
  DBD=DIVNTZ02  DDN=DBHVSAM2 DSID=002 DBORG=HISAM  DSORG=VSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=2     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 34. Sample Listing of a RECON at the Active Site - More IMS DB Records (Part 4 of 5)

```
1997.274 09:27:24.1 -09:00          LISTING OF RECON
--------------------------------------------------------------------------------
 IMAGE
  RUN     = 1997.274 08:31:07.0 -09:00     *  RECORD COUNT =0
  STOP    = 0000.000 00:00:00.0 +00:00        BATCH      USID=0000000001

 IC1
  DSN=IMSTESTG.DIVNTZ02.DBHVSAM2.BASE.IC           FILE SEQ=0001
  UNIT=SYSDA                          VOLS DEF=0001 VOLS USED=0001
                                      VOLSER=IMSRAW



 IMAGE
  RUN     = 1997.274 09:13:06.2 -09:00     *  RECORD COUNT =5
  STOP    = 1997.274 09:13:09.0 -09:00        ONLINE     USID=0000000002

 IC1
  DSN=IMSVS.DIVNTZ02.DBHVSAM2.IC.IC101300          FILE SEQ=0001
  UNIT=SYSDA                          VOLS DEF=0001 VOLS USED=0001
                                      VOLSER=222222
```

Figure 34. Sample Listing of a RECON at the Active Site - More IMS DB Records (Part 5 of 5)

Sample Listing of a RECON at the Tracking Site

Beginning iwth Figure 35 on page 367, 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 376 describes the fields that can be present in a listing of the RECON.

RECON Status Record

```
        IMS/ESA VERSION 6 RELEASE 1  DATA BASE RECOVERY CONTROL
 /* LIST IN LOCAL TIME */
 LIST.RECON TIMEFMT(L,O,P,4)
1997.274 09:27:48.3 -09:00          LISTING OF RECON
-------------------------------------------------------------------------------
 RECON
  RECOVERY CONTROL DATA SET, IMS/ESA V6R1
  DMB#=8                             INIT TOKEN=97274F1725180F
  NOFORCER  LOG DSN CHECK=CHECK17    STARTNEW=NO
  TAPE UNIT=3400      DASD UNIT=SYSDA     TRACEOFF   SSID=**NULL**
  LIST DLOG=YES                CA/IC/LOG DATA SETS CATALOGED=NO
  LOG RETENTION PERIOD=00.000 00:15:00.0

  TIME STAMP INFORMATION:

    TIMEZIN = %SYS             -LABEL- -OFFSET-
                                PDT     -07:00
                                PST     -08:00

    OUTPUT FORMAT:  DEFAULT = LOCORG NONE   PUNC YY
                    CURRENT = LOCAL  OFFSET PUNC YYYY


 -DDNAME-      -STATUS-       -DATA SET NAME-
  RECON1        COPY1          IMSTESTL.IMS.RECON1
  RECON2        COPY2          IMSTESTL.IMS.RECON2
  RECON3        SPARE          IMSTESTL.IMS.RECON3
```

Figure 35. Sample Listing of a RECON at the Tracking Site - RECON Status Record

Log Records

```
1997.274 09:27:48.3 -09:00          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 = 33
  START TIME OF CURRENT LOG = 1997.274 09:25:15.8 -09:00
  HIGHEST ACTIVE SITE TIME  = 1997.274 09:25:58.2 -09:00
  TRACKING SUBSYSTEM ID     = SYS3
```

Figure 36. Sample Listing of a RECON at the Tracking Site - Log Records

SSYS Record

```
1997.274 09:27:48.3 -09:00          LISTING OF RECON
--------------------------------------------------------------------------------
 SSYS
  SSID=SYS3      LOG START=1997.274 08:35:15.0 -09:00
  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=4      VERSION=6.1
                                                    ENCODED
    -DBD-        -AREA-   -LEVEL-  -ACCESS INTENT-  -STATE-
    DEDBJN21     DB21AR0     1        EXCLUSIVE        7
    DHVNTZ02     **NULL**    3        EXCLUSIVE        7
    DXVNTZ02     **NULL**    3        EXCLUSIVE        7
    DIVNTZ02     **NULL**    3        EXCLUSIVE        7

 SSYS
  SSID=SYS3      LOG START=1997.274 09:12:08.5 -09:00
  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=6.1
                                                    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 37. Sample Listing of a RECON at the Tracking Site - SSYS Record

BACKOUT Record

```
1997.274 09:27:48.3 -09:00          LISTING OF RECON
--------------------------------------------------------------------------------
 BACKOUT
  SSID=SYS3      #UORS=1

    RECOVERY TOKEN=E2E8E2F340404040000000300000002
    TIME=1997.274 09:13:27.0 -09:00       PSB=PLVAPZ12
                INFLT   BMP
      ASSOCIATED DATA BASES=3

                   BACKED   DYN BKOUT
      -DBD-        -OUT -   -FAILURE-
      DHVNTZ02      NO         NO
      DXVNTZ02      NO         NO
      DIVNTZ02      NO         NO
```

Figure 38. Sample Listing of a RECON at the Tracking Site - BACKOUT Record

CAGRP and CA Records

```
1997.274 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    = 1997.274 08:40:44.3 -09:00     *
                     UNIT=SYSDA      VOLS DEF=1    VOLS USED=1
                                     VOLSER=222222
  RUN     = 1997.274 09:26:39.6 -09:00
   DBD=DEDBJN21 DDN=DB21AR1  PURGETIME = 1997.274 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 = 1997.274 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 = 1997.274 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 = 1997.274 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 39. Sample Listing of a RECON at the Tracking Site - CAGRP and CA Records

DBDSGRP Records

```
1997.274 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 40. Sample Listing of a RECON at the Tracking Site - DBDSGRP Records

DB (IMS) and Related Records

```
1997.274 09:27:48.3 -09:00          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
  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
    RECOV NEEDED   =OFF
    RECEIVE NEEDED =OFF                EEQE COUNT              =0
```

Figure 41. Sample Listing of a RECON at the Tracking Site - DB (IMS) and Related Records

DB (FP) and Related Records

```
1997.274 09:27:48.3 -09:00          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
                                        EEQE COUNT              =0
 DBDS
  DBD=DEDBDD01  AREA=DD01AR0                                    TYPE=FP
  SHARE LEVEL=1             DSID=001 DBORG=DEDB   DSORG=VSAM
  GSGNAME=IMSGSG1              USID=0000000003
  AUTHORIZED USID=0000000000  RECEIVE USID=0000000000 HARD USID=0000000000
  RECEIVE NEEDED USID=0000000000
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=0     DSSN=00000002
  NOREUSE         RECOVPD=0      NOVSO  PREOPEN    NOPRELOAD
  CFSTR1=**NULL**         CFSTR2=**NULL**          NOLKASID
  DEFLTJCL=**NULL**  ICJCL=ICJCL    RECVJCL=ICRCVJCL  RECOVJCL=RECOVJCL
  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   =1997.274 09:12:17.0 -09:00     *  ALLOC LRID =00000000000000D7
  DSSN=0000000001 USID=0000000002 START = 1997.274 09:12:08.5 -09:00

 ALLOC
  ALLOC   =1997.274 09:17:51.9 -09:00     *  ALLOC LRID =0000000000000332
  DSSN=0000000002 USID=0000000003 START = 1997.274 09:17:31.2 -09:00
```

Figure 42. Sample Listing of a RECON at the Tracking Site - DB (FP) and Related Records

IMS DB Records

```
1997.274 09:27:48.3 -09:00          LISTING OF RECON


--------------------------------------------------------------------------------
 DB
  DBD=DHVNTZ02                IRLMID=*NULL        DMB#=5        TYPE=IMS
  SHARE LEVEL=3              GSGNAME=IMSGSG1      USID=0000000003
  AUTHORIZED USID=0000000003  RECEIVE USID=0000000001 HARD USID=0000000003
  RECEIVE NEEDED USID=0000000000
  FLAGS:                                COUNTERS:
    BACKOUT NEEDED        =ON             RECOVERY NEEDED COUNT   =0
    READ ONLY             =OFF            IMAGE COPY NEEDED COUNT =0
    PROHIBIT AUTHORIZATION=OFF            AUTHORIZED SUBSYSTEMS   =2
    RECOVERABLE           =YES            HELD AUTHORIZATION STATE=6
    DATABASE LEVEL TRACK  =YES            EEQE COUNT              =0
    TRACKING SUSPENDED    =NO             RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =NO

  ASSOCIATED SUBSYSTEM INFORMATION:
                               ENCODED  B/O NEEDED
     -SSID-   -ACCESS INTENT-  -STATE-   -COUNT-  -SS ROLE-
    SYS3         EXCLUSIVE        7         1      TRACKING
    SYS3         UPDATE           6         0      ACTIVE
```

Figure 43. Sample Listing of a RECON at the Tracking Site - More IMS DB Records (Part 1 of 5)

```
1997.274 09:27:48.3 -09:00          LISTING OF RECON
------------------------------------------------------------------------------
 DBDS
  DSN=DHVNTZ02.FKXXI01E                                          TYPE=IMS
  DBD=DHVNTZ02  DDN=HIDAM    DSID=001 DBORG=HIDAM  DSORG=VSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     DSSN=00000002
  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   =1997.274 09:09:02.5 -09:00     *  ALLOC LRID =0000000000000009
  DSSN=0000000001 USID=0000000002 START = 1997.274 09:08:59.4 -09:00

 ALLOC
  ALLOC   =1997.274 09:13:25.2 -09:00     *  ALLOC LRID =0000000000000134
  DSSN=0000000002 USID=0000000003 START = 1997.274 09:12:08.5 -09:00

 IMAGE
  RUN     = 1997.274 08:31:07.0 -09:00     *  RECORD COUNT =0
  STOP    = 0000.000 00:00:00.0 +00:00        BATCH      USID=0000000001

 IC1
  DSN=IMSTESTG.DHVNTZ02.HIDAM.BASE.IC              FILE SEQ=0001
  UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=IMSRAW



 RECOV
  RUN     = 1997.274 08:32:06.4 -09:00     *  RUN USID      = 0000000001
```

Figure 43. Sample Listing of a RECON at the Tracking Site - More IMS DB Records (Part 2 of 5)

```
1997.274 09:27:48.3 -09:00          LISTING OF RECON
--------------------------------------------------------------------------------
 DBDS
  DSN=DHVNTZ02.FKXXI02E                                           TYPE=IMS
  DBD=DHVNTZ02  DDN=HIDAM2   DSID=002 DBORG=HIDAM  DSORG=VSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     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 =ON                 EEQE COUNT              =0


 IMAGE
  RUN     = 1997.274 08:31:07.0 -09:00     *  RECORD COUNT =0
  STOP    = 0000.000 00:00:00.0 +00:00        BATCH      USID=0000000001

 IC1
  DSN=IMSTESTG.DHVNTZ02.HIDAM2.BASE.IC             FILE SEQ=0001
  UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=IMSRAW



 RECOV
  RUN     = 1997.274 08:32:10.6 -09:00     *  RUN USID      = 0000000001
 DB
  DBD=DIVNTZ02                IRLMID=*NULL        DMB#=4        TYPE=IMS
  SHARE LEVEL=3              GSGNAME=IMSGSG1      USID=0000000003
  AUTHORIZED USID=0000000003  RECEIVE USID=0000000001 HARD USID=0000000003
  RECEIVE NEEDED USID=0000000000
  FLAGS:                             COUNTERS:
    BACKOUT NEEDED        =ON          RECOVERY NEEDED COUNT   =0
    READ ONLY             =OFF         IMAGE COPY NEEDED COUNT =0
    PROHIBIT AUTHORIZATION=OFF         AUTHORIZED SUBSYSTEMS   =2
    RECOVERABLE           =YES         HELD AUTHORIZATION STATE=6
    DATABASE LEVEL TRACK  =YES         EEQE COUNT              =0
    TRACKING SUSPENDED    =NO          RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =NO

  ASSOCIATED SUBSYSTEM INFORMATION:
                              ENCODED  B/O NEEDED
     -SSID-   -ACCESS INTENT-  -STATE-   -COUNT-  -SS ROLE-
    SYS3        EXCLUSIVE       7        1      TRACKING
    SYS3        UPDATE          6        0      ACTIVE
```

Figure 43. Sample Listing of a RECON at the Tracking Site - More IMS DB Records (Part 3 of 5)

```
1997.274 09:27:48.3 -09:00          LISTING OF RECON
-------------------------------------------------------------------------------
 DBDS
  DSN=DIVNTZ02.FJXXS01K                                               TYPE=IMS
  DBD=DIVNTZ02  DDN=DBHVSAM1 DSID=001 DBORG=HISAM  DSORG=VSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     DSSN=00000002
  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   =1997.274 09:09:04.5 -09:00     *  ALLOC LRID =000000000000006C
  DSSN=0000000001 USID=0000000002 START = 1997.274 09:08:59.4 -09:00

 ALLOC
  ALLOC   =1997.274 09:13:27.2 -09:00     *  ALLOC LRID =00000000000001A1
  DSSN=0000000002 USID=0000000003 START = 1997.274 09:12:08.5 -09:00

 IMAGE
  RUN     = 1997.274 08:31:07.0 -09:00     *  RECORD COUNT =0
  STOP    = 0000.000 00:00:00.0 +00:00        BATCH      USID=0000000001

 IC1
  DSN=IMSTESTG.DIVNTZ02.DBHVSAM1.BASE.IC           FILE SEQ=0001
  UNIT=SYSDA                        VOLS DEF=0001 VOLS USED=0001
                                    VOLSER=IMSRAW



 RECOV
  RUN     = 1997.274 08:31:59.7 -09:00     *  RUN USID      = 0000000001
```

Figure 43. Sample Listing of a RECON at the Tracking Site - More IMS DB Records (Part 4 of 5)

```
1997.274 09:27:48.3 -09:00          LISTING OF RECON
-------------------------------------------------------------------------------
 DBDS
  DSN=DIVNTZ02.FJXXS01E                                          TYPE=IMS
  DBD=DIVNTZ02  DDN=DBHVSAM2 DSID=002 DBORG=HISAM  DSORG=VSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     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 =ON               EEQE COUNT             =0

 IMAGE
  RUN     = 1997.274 08:31:07.0 -09:00     *  RECORD COUNT =0
  STOP    = 0000.000 00:00:00.0 +00:00        BATCH      USID=0000000001

 IC1
  DSN=IMSTESTG.DIVNTZ02.DBHVSAM2.BASE.IC           FILE SEQ=0001
  UNIT=SYSDA                       VOLS DEF=0001 VOLS USED=0001
                                   VOLSER=IMSRAW



 RECOV
  RUN     = 1997.274 08:32:03.7 -09:00     *  RUN USID      = 0000000001
```

Figure 43. Sample Listing of a RECON at the Tracking Site - More IMS DB Records (Part 5 of 5)

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:

Examples of many of these fields appear in Figure 25 on page 348 and Figure 35 on page 367.

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 19.

Table 19. Fields Present in the RECON Record

Record Type	Line Number	Field	Contents
RECON	1		
	2	RECOVERY CONTROL DATA SET, IMS/ESA VxRx	Identifies the version and release of the RECON.
		COEXISTENCE ENABLED	Specifies that pre-version 6 and version 6 RECONs may coexist.
	3	DMB#= *nnn*	Represents the last value assigned to a new database record.
		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.
		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.

Table 19. Fields Present in the RECON Record (continued)

Record Type	Line Number	Field	Contents
	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	LOG RETENTION PERIOD= *yy.ddd hh:mm:ss.t*	Indicates the minimum amount of time that DBRC is to keep log records in RECON.
The following line is printed only if a failure occurred during a multiple update to the RECON:			
	8	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.
	9	OLD RECORD KEY	This field is printed only if the multiple update involves a key change.
	10	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= *timestamp*	The time stamp of the key of the record.

Table 19. Fields Present in the RECON Record (continued)

Record Type	Line Number	Field	Contents
	11-12	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.
	13-14	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.
	15-16	SSID= *ssidname* IRLMID= *irlmidname* SHARE LVL #= *n* #DB= *n* B/O#= *n*	Information about the multiple update.
	17	FLAGS:	A heading for the following lines.
	18	RECOV= ON | OFF NORECOV= ON | OFF ICON= ON | OFF ICOFF= ON | OFF	Information about the multiple update.
	19	NORMAL= ON | OFF ABNORMAL= ON | OFF STARTRCV= ON | OFF ENDRECOV= ON | OFF	Information about the multiple update.
	20	READON= ON | OFF READOFF= ON | OFF AUTH= ON | OFF NOAUTH= ON | OFF	Information about the multiple update.
	21	B/O DONE= ON | OFF PASS 1= ON | OFF BACKOUT= ON | OFF SHARELVL= ON | OFF	Information about the multiple update.
	22	TIME STAMP INFORMATION:	Heading for section containing RECON time stamp information.
	23	TIMEZIN= *offset*	User-specified input time stamp offset default. If no offset is specified, this field displays %SYS meaning that the offset of the MVS clock is to be used.
		-LABEL- -OFFSET-	These column headings are printed if a Time Zone Label Table has been defined.
	24-nnn	*label offset*	The list of defined time zone offset labels with their corresponding offset values.
	25	OUTPUT FORMAT:	Time stamp output format settings.
		DEFAULT= *offset display form yearsize*	Default time stamp output format settings.
	26	CURRENT= *offset display form yearsize*	Current time stamp output format settings.

Table 19. Fields Present in the RECON Record (continued)

Record Type	Line Number	Field	Contents
	27	-DDNAME- -STATUS- -DATA SET NAME-	Column headings for the following one to three lines.
	28	RECON1 | RECON2 RECON3 | COPY1 *dsname*	The ddname, status, and data set name of the Copy1 RECON.
	29-30	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 THT Record

A time history table record's fields and their corresponding line numbers are described in Table 20.

Table 20. Fields Present in the THT Record

Record Type	Line Number	Field	Contents
THT	1		
	2	-LOCAL START- -OFFSET-	Column headings for the Time History Table.
	3-nnn	*yyyy.ddd hh:mm:ss.t ±hh:mm*	The local start time of the period in which the offset is in effect.

Fields Present in a Log Record

The log records include the types listed under `Record Type` in the Table 21 on page 381.

A log record's fields and their corresponding line numbers are described in Table 21 on page 381.

Table 21. 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		
	2	START= *timestamp**	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= *timestamp*	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.

Table 21. Fields Present in a Log Record (continued)

Record Type	Line Number	Field	Contents
		PRILOG TOKEN= *n*	The numeric log token assigned sequentially to PRILOG records for the same GSG.
	6	EARLIEST CHECKPOINT= *timestamp*	This line is printed only for online PRILOGs. It indicates the earliest checkpoint required by IMS emergency restart (`/ERE` or `/ERE BUILDQ`).
	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= *timestamp*	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= *timestamp*	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* *CLOSED FEOV*	Count of checkpoint records in the log data set. This field is only printed for tracking logs. Indicates that the log data set is closed. This field is only printed for tracking PRILOGs. Indicates that the corresponding log data set at the active site was closed with a /DBR FEOV. This field is only printed for tracking PRILOGs.
	11	VOLSER= *volser* STOPTIME= *timestamp*	Volume serial number and stop time. This field is not printed for tracking logs.
	12	CKPTCT= *n* CHKPT ID= *timestamp*	

Table 21. 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 only printed for tracking PRILOGs.

Fields Present in a LOGALL Record

A log allocation record's fields and their corresponding line numbers are described in Table 22.

Table 22. Fields Present in the LOGALL Record

Record Type	Line Number	Field	Contents
LOGALL	1		
	2	START= *timestamp**	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	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.
	4-*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 23 on page 384.

Fields in an Online Log Record

Table 23. Fields Present in an Online 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		
	2	START= *timestamp**	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= *timestamp*	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.

Table 23. Fields Present in an Online Log Record (continued)

Record Type	Line Number	Field	Contents
		PRILOG TOKEN= *n*	The numeric log token assigned sequentially to PRILOG records for the same GSG.
	6	EARLIEST CHECKPOINT= *timestamp*	This line is printed only for online PRILOGs. It indicates the earliest checkpoint required by IMS emergency restart (`/ERE` or `/ERE BUILDQ`).
	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= *timestamp*	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= *timestamp*	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*	Count of checkpoint records in the log data set. This field is only printed for tracking logs.
	11	VOLSER= *volser* STOPTIME= *timestamp*	Volume serial number and stop time. This field is not printed for tracking logs.
	12	CKPTCT= *n* CHKPT ID= *timestamp*	Checkpoint count and ID value.
	blank line		These blank lines are printed as visual separators only when the log data set has more than one data set entry. The format of each data set entry is a repetition of lines 7 through 12 above.
LOGALL	1		

Table 23. Fields Present in an Online Log Record (continued)

Record Type	Line Number	Field	Contents
	2	START= *timestamp**	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	DBDS ALLOC= *nnnn*	The number of allocated DBDSs for this log data set.
		-DBD- -DDN- -ALLOC-	These headings 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.
	4-*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.
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= *timestamp*	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= *timestamp*	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= *timestamp*	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.

Table 23. Fields Present in an Online Log Record (continued)

Record Type	Line Number	Field	Contents
		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= *timestamp*	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 24.

Table 24. 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.

Table 24. Fields Present in a GSG Record (continued)

Record Type	Line Number	Field	Contents
	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= *timestamp*	The start time of the PRILOG with the current log token.
	8	HIGHEST ACTIVE SITE TIME= *timestamp*	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 an SSYS Record

A subsystem record's fields and their corresponding line numbers are described in Table 25.

Table 25. Fields Present in an SSYS Record

Record Type	Line Number	Field	Contents
SSYS	1		
	2	SSID= *ssidname*	The name of the IMS subsystem.
		LOG START= *timestamp*	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.
		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.

Table 25. Fields Present in an SSYS Record (continued)

Record Type	Line Number	Field	Contents
		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.
		VERSION= *n*	Identifies the IMS release level through which the subsystem signed on.
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:			
	8-9	-DBD- -AREA- -LEVEL- -ACCESSINTENT- ENCODED -STATE-	Column headings for the following line or lines that describe the currently authorized databases or areas.
	10-*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 26.

Table 26. Fields Present in a BACKOUT Record

Record Type	Line Number	Field	Contents
BACKOUT	1		

Table 26. Fields Present in a BACKOUT Record (continued)

Record Type	Line Number	Field	Contents
	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= *timestamp*	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 27.

Table 27. Fields Present in a CAGRP Record

Record Type	Line Number	Field	Contents
CAGRP	1		
	2	GRPNAME= *cagrpnam*	The name of the CA group.

Table 27. Fields Present in a CAGRP Record (continued)

Record Type	Line Number	Field	Contents
		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.
	3	REUSE | 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 28.

Table 28. 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.

Table 28. Fields Present in a CA Record (continued)

Record Type	Line Number	Field	Contents
	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 | STOP= *timestamp**	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.
		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= *timestamp*	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.			

Table 28. Fields Present in a CA Record (continued)

Record Type	Line Number	Field	Contents
	7	DBD= *dbdname* DDN= *ddname*	The database name and data set ddname of the DBDS.
		PURGETIME= *timestamp*	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.
		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 DBDSGRP Record

A database data set group record's fields and their corresponding line numbers are described in Table 29.

Table 29. Fields Present in the DBDSGRP Record

Record Type	Line Number	Field	Contents
DBDSGRP	1		
	2	GRPNAME= *dbgroup* | *dbdsgrpname*	Name of database or DBDS group.
		#MEMBERS= *nnn*	Number of members belonging to the group.
		-DBD- -DDN/AREA-	These headings and the following line (3) are printed only if some members of this group exist.

Fields in a DBDSGRP Record

Table 29. Fields Present in the DBDSGRP Record (continued)

Record Type	Line Number	Field	Contents
	3-*nnn*	*dbdname* *ddname/areaname*	The database name and ddname or area name for this member of the group. For database groups (DBGROUPSs), *dbdname* is the DBD or area name and the. *ddname/areaname* columns contains `**NULL**`.

Fields Present in a DB (non-Fast Path) Record

A non-Fast Path database record's fields and their corresponding line numbers are described in Table 30.

Table 30. Fields Present in the DB (non-Fast Path) Record

Record Type	Line Number	Field	Contents
DB (non-Fast Path)	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.
	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.

Table 30. Fields Present in the DB (non-Fast Path) Record (continued)

Record Type	Line Number	Field	Contents
	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	FLAGS: COUNTERS:	This heading line is printed for lines 7 through 13, which describe the status of this database.
	7	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.
	8	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.
	9	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.
	10	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 30. Fields Present in the DB (non-Fast Path) Record (continued)

Record Type	Line Number	Field	Contents
	11	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.
	12	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.
	13	OFR REQUIRED= YES | NO	Indicates whether online forward recovery is required for this shadow database.
The following lines 14 through 17 are printed only when one or more subsystems are currently authorized to this database:			
	14, 15, 16	ASSOCIATED SUBSYSTEM INFORMATION: -SSID- -AC INTENT- ENCODED -STATE- B/O NEEDED -COUNT- -SS ROLE-	Headings for lines 17-nnn. These headings are printed if subsystems exist
	17-*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 DB (Fast Path) Record

A Fast Path database record's fields and their corresponding line numbers are described in Table 31 on page 397.

Table 31. 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 this database 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 32.

Table 32. 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.
		TYPE= IMS	Indicates that this is a DBDS rather than a FP DEDB area.

Fields in a DBDS (non-Fast Path) Record

Table 32. Fields Present in the DBDS (non-Fast Path) Record (continued)

Record Type	Line Number	Field	Contents
	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.
		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.
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.

Table 32. Fields Present in the DBDS (non-Fast Path) Record (continued)

Record Type	Line Number	Field	Contents
	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.
	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	RECOV NEEDED= ON | OFF	Indicates whether the DBDS needs to be recovered.
	11	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 12 through 14, are printed only if one or more extended error queue elements exist.			
	12, 13	ERROR QUEUE ELEMENTS: -EQERBA-EEQE TYPE-SSID-	Heading for the following list of extended error queue elements (lines 14-nnn).
	14-*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 33.

Table 33. 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.

Fields in a DBDS (Fast Path) Record

Table 33. Fields Present in the DBDS (Fast Path) Record (continued)

Record Type	Line Number	Field	Contents
	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.
		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.

Table 33. Fields Present in the DBDS (Fast Path) Record (continued)

Record Type	Line Number	Field	Contents
		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.
	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.
		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	FLAGS: COUNTERS:	This heading line is printed for lines 11 through 19 describing the status of this area.
	12	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.

Fields in a DBDS (Fast Path) Record

Table 33. Fields Present in the DBDS (Fast Path) Record (continued)

Record Type	Line Number	Field	Contents
	13	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.
	14	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.
	15	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.
	16	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.
	17	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.
	18	OFR REQUIRED= YES | NO	Indicates whether online forward recovery is required for this shadow area.
	19	TRACKING SUSPENDED= YES | NO	Indicates whether tracking has been suspended for this shadow area.
	20	HSSP CIC IN PROGRESS= YES | NO	Indicates whether an HSSP concurrent image copy is in progress.
The following lines 21 through 23, are printed only if one or more error queue elements exist.			
	21 22	ERROR QUEUE ELEMENTS: -EQERBA-EEQE TYPE-SSID-	Heading for the following list of error queue elements.
	23-*nnn*	*eeqe rba*	The type of extended error queue element.
		eeqe type	The type of extended error queue element.

Table 33. Fields Present in the DBDS (Fast Path) Record (continued)

Record Type	Line Number	Field	Contents
		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 24 through 26, list those area data sets that are currently registered in this area record:			
	24 25 26	ADS LIST: -ADS DDN- -ADS DSN- -STAT- CREATE -RUNNING-	These lines 24 through 26, represent the column headings for the following line 27.
The following line 27 is repeated for each area data set that is registered for this area.			
	27-*nn*	*adsddn* *adsdsn* AVAIL | UNAVAIL YES | 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 | 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 34.

Table 34. Fields Present in the ALLOC Record

Record Type	Line Number	Field	Contents
ALLOC	1		

Fields in an ALLOC Record

Table 34. Fields Present in the ALLOC Record (continued)

Record Type	Line Number	Field	Contents
	2	ALLOC= *timestamp**	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= *timestamp*	The start time of the log data set that was logging change records when this DBDS was allocated.
	4	DEALLOC= *timestamp*	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 35.

Table 35. Fields Present in the IMAGE Record

Record Type	Line Number	Field	Contents
IMAGE	1		

Table 35. Fields Present in the IMAGE Record (continued)

Record Type	Line Number	Field	Contents
	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).
		CREATE | RUN= *timestamp**	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= *timestamp*	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 that a system managed storage image copy was run with shared access. SMSNOCIC indicates that an SMS image copy was run with exclusive access. This field applies to standard image copy data sets, but is not printed for available image copy data sets.

Fields in an IMAGE Record

Table 35. Fields Present in the IMAGE Record (continued)

Record Type	Line Number	Field	Contents
		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).
		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 36 on page 407.

Table 36. Fields Present in the REORG Record

Record Type	Line Number	Field	Contents
REORG	1		
	2	RUN= *timestamp**	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.

Fields Present in a RECOV Record

A recovery record's fields and their corresponding line numbers are described in Table 37.

Table 37. Fields Present in the RECOV Record

Record Type	Line Number	Field	Contents
RECOV	1		
	2	RUN= *timestamp**	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= *timestamp*	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 D. Considering IMS DBRC RECON Data Set Placement

This appendix consists of recommendations for the placement of the IMS DBRC RECON data sets. Shared DASD considerations and the use of GRS (Global Resource Serialization) are also discussed.

The RECON data sets consist of three VSAM KSDSs. These data sets can be considered to be among the most important IMS system data sets in an installation, since they are required to control the recovery of registered data base data sets, and also required to control logging and system restart operations for IMS DC and DBCTL environments. Also, since multiple IMS subsystems can share the same set of RECONs, the loss of the RECON data sets could impact multiple IMS subsystems.

The RECON data sets are managed with a ″pair and a spare″ concept. There are two active KSDSs and a spare KSDS that is used when one of the active RECON data sets is lost (DBRC automatically copies the remaining good copy to the spare in order to maintain dual processing). In this situation the loss of one of the RECONs in a RECON set of data sets is not a major problem. However, the concurrent loss of both active RECON KSDSs in a RECON set of data sets negatively impacts IMS system availability.

To avoid losing both active RECONs at the same time, it is generally recommended that any common points of failure for a set of RECONs be eliminated. In particular, for a given set of RECONs:

1. Place each of the RECON data sets on a separate device and string. In addition, make sure that a control unit, channel, or director failure does not cause the only path to multiple RECON data sets to be lost.
2. Catalog each of the RECON data sets in unique (separate) catalogs and make sure that the catalogs are on separate devices (preferably, the same device as the RECON data set). This ensures that a catalog failure does not cause multiple RECONs in the RECON set of data sets to be lost.

The RECON data sets can be shared across multiple IMS systems on multiple processors or MVS images. When an IMS subsystem accesses the RECONs, DBRC issues reserves against the RECONs to maintain integrity. While all three RECON data sets are not always reserved, the data sets that are reserved are reserved in DD name (RECON1, RECON2, RECON3) sequence.

These reserves can be either hardware reserves or can be converted to software reserves (SYSTEMS ENQs) by GRS. If hardware reserves are used, then one must allocate the RECON data sets, in a shared DASD environment, carefully to avoid shared DASD deadlocks or intolerable contention with other data sets that might be on the same volumes as the RECONs. Note that hardware reserves lock out access to the entire volume from other CPUs or MVS images.

Shared DASD hardware reserve deadlocks can be avoided by ensuring that:

1. The catalog (BCS) used for a RECON data set is on the same volume as the RECON data set.
2. There are no catalogs (BCS) on a RECON volume that point to VSAM data sets on another RECON volume.

3. If multiple sets of RECONs (RECON set A and RECON set B) exist and you wish to share DASD volumes for the multiple sets (use 3 volumes rather than n * 3 volumes), all of the RECON1 data sets from the multiple RECON sets are on one volume, all of the RECON2 data sets are on the second volume, and all of the RECON3 data sets are on the third volume.

Implementing the above recommendations should greatly reduce all known shared DASD hardware reserve deadlock exposures involving the RECON data sets.

GRS can be used to convert reserve requests from hardware reserves to SYSTEMS ENQ requests and to then propagate the SYSTEMS ENQ request to all CPUs or MVS images in the sharing Sysplex. When this facility is used, the entire volume is not locked out from access by other CPUs or MVS images; only the logical resource (the particular RECON data set in the case of DBRC) is serialized.

So, in a GRS environment, the reserve requests issued by DBRC do not prevent access from other CPUs or MVS images to other data sets that might reside on the RECON volumes. In addition, by using the GRS conversion of RECON reserves, you eliminate any potential DBRC reserve deadlock situations, even when the previous recommendations are not followed.

The use of GRS is not without overhead. It takes longer for a ″software″ (GRS converted) reserve to be processed than it takes for a hardware reserve to be processed. The amount of elapsed time increase varies by the number of CPUs or MVS images in the Sysplex and installation specified tuning parameters. Using an example in *MVS/ESA Planning: Global Resource Serialization for MVS/SP Version 3* the average amount of increase for a 4-processor Sysplex would vary from 7 to 12 milliseconds. Most invocations of DBRC only reserve the active pair of RECONs (two reserves) so this time must be doubled (occasionally tripled when the spare is also reserved). Therefore, the use of GRS to convert the DBRC RECON reserves would typically add less than 25 milliseconds (about one random I/O) to the duration of an IMS DBRC operation.

Since this increase is probably much less than 10 percent when compared to the duration of an IMS DBRC operation, the potential negative performance impact of GRS usage should not be a serious consideration for most IMS DBRC users, especially when the positive performance impacts (entire volume no longer held) considered. However, environments could exist where there is an extremely high degree of contention among multiple IMS subsystems for the RECON data sets. In these environments, the negative performance impact of GRS usage could be noticeable.

The GRS manual (referenced earlier) should be consulted for additional information regarding the implementation, usage, and tuning considerations for GRS.

Bibliography

This bibliography includes all the publications cited in this book, including the publications in the IMS library.

- *CICS/ESA-CICS IMS Database Control Guide*
- *CICS/ESA-CICS Supplied Transactions*
- *MVS/DFP V3R3 System Program Reference Manual*
- *MVS/ESA System Programming Library: Application Development Guide*
- *OS/390 MVS Planning: Global Resource Serialization*
- *OS/390 MVS Programming: Authorized Assembler Services Reference*

IMS/ESA Version 6 Library

SC26-8725	ADB	Administration Guide: Database Manager
SC26-8730	AS	Administration Guide: System
SC26-8731	ATM	Administration Guide: Transaction Manager
SC26-8727	APDB	Application Programming: Database Manager
SC26-8728	APDG	Application Programming: Design Guide
SC26-8726	APCICS	Application Programming: EXEC DLI Commands for CICS and IMS
SC26-8729	APTM	Application Programming: Transaction Manager
SC26-8732	CG	Customization Guide
SC26-9517	CQS	Common Queue Server Reference
SC26-8733	DBRC	Database Recovery Control Guide and Reference
LY37-3731	DGR	Diagnosis Guide and Reference
LY37-3732	FAST	Failure Analysis Structure Tables (FAST) for Dump Analysis
GC26-8736	IIV	Installation Volume 1: Installation and Verification
GC26-8737	ISDT	Installation Volume 2: System Definition and Tailoring
SC26-8740	MIG	Master Index and Glossary
GC26-8739	MC	Messages and Codes
SC26-8743	OTMA	Open Transaction Manager Access Guide
SC26-8741	OG	Operations Guide
SC26-8742	OR	Operator’s Reference
GC26-8744	RPG	Release Planning Guide
SC26-8767	SOP	Sample Operating Procedures
SC26-8769	URDB	Utilities Reference: Database Manager
SC26-8770	URS	Utilities Reference: System
SC26-8771	URTM	Utilities Reference: Transaction Manager

Supplementary Publications

GC26-8738	LPS	Licensed Program Specifications
SC26-8766	SOC	Summary of Operator Commmands

Online Softcopy Publications

LK3T-2326	CDROM	IMS/ESA Version 6 Softcopy Library
SK2T-0730	CDROM	IBM Online Library: Transaction Processing and Data
SK2T-0710	CDROM	MVS Collection
SK2T-6700	CDROM	OS/390 Collection

Index

Special Characters

C

D

E

F

G

H

I

J

K

L

M

N

O

S

T

Readers' Comments — We'd Like to Hear from You

IMS/ESA
DBRC Guide and Reference
Version 6

Publication No. SC26-8733-04

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name	Address
Company or Organization	
Phone No.	

Readers' Comments — We'd Like to Hear from You
SC26-8733-04

Cut or Fold Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International
Business Machines Corporation
Department BWE/H3
P.O. Box 49023
San Jose, CA
95161-9945

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold Along Line

Program Number: 5655-158

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC26-8733-04

Spine information:

IBM IMS/ESA IMS/ESA V6 DBRC Guide & Reference Version 6