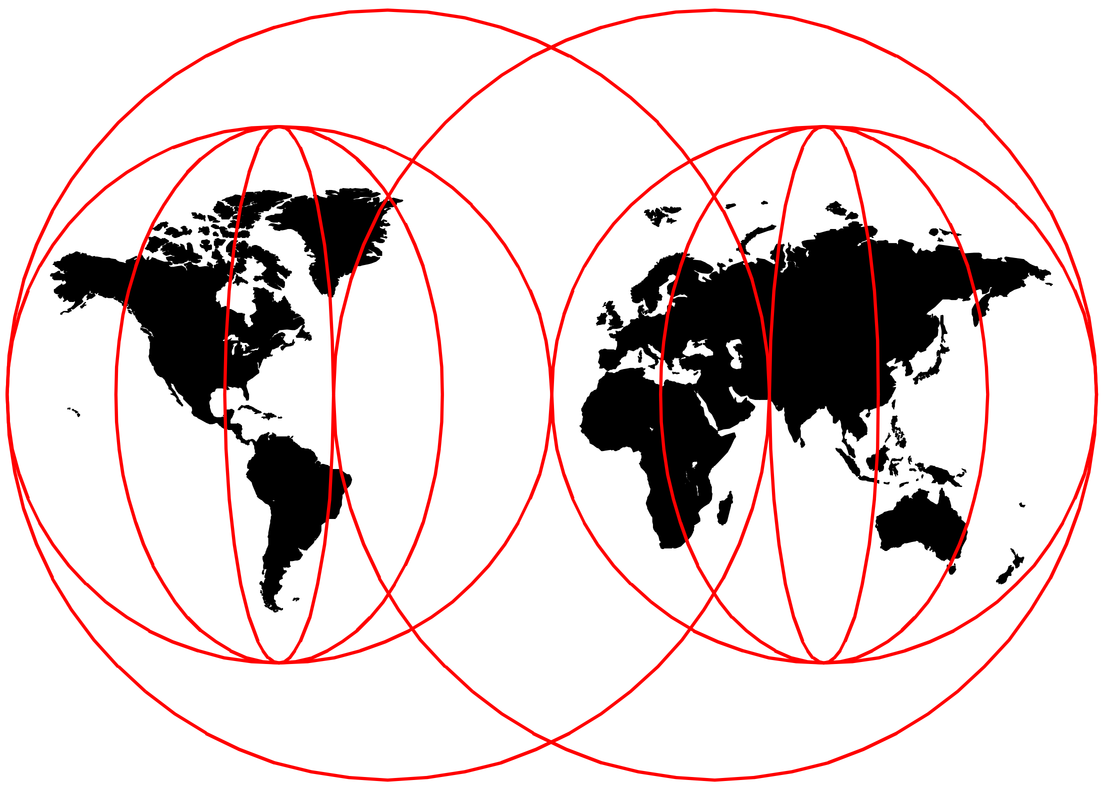
Database Recovery Control (DBRC) Examples and Usage Hints

Rick Long, Horst Mueller, Marco Scudeletti



International Technical Support Organization

www.redbooks.ibm.com

SG24-3333-01

International Technical Support Organization

SG24-3333-01

Database Recovery Control (DBRC) Examples and Usage Hints

July 1999

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, “Special Notices” on page 125.

Second Edition (July 1999)

This edition applies to IMS/VS Version 2 (5665-332).

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept.QXXE Building 80-E2
650 Harry Road
San Jose, California 95120-6099

Contents

Figures

Preface

This book was orginally published in 1989. Since it is a valuable source of information, we have rereleased this publication. Although some of the information is a little dated, it is mainly correct and usable with today's releases of IMS.

This redbook provides information on how to implement and use IMS/VS Database Recovery Control (DBRC) in complex environments. It also furnishes practical examples about environments based on the current level of DBRC.

Current IMS bulletins typically address only simple situations when dealing with DBRC examples. Also, IMS/VS DB Version 2 introduces several enhancements for DBRC, but existing technical bulletins on DBRC do not fully reflect these changes.

This redbook is intended for system programmers, application programmers, security administrators, database adminstrators (DBAs), and data processing (DP) operation personnel involved with database recovery.

The Team That Wrote This Redbook

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

Rick Long from IBM Australia

Horst Mueller from IBM Germany

Marco Scudeletti from the International Technical Support Center

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

- Alan Tippett
- Bill Keene
- Rich Lewis
- Loy Prather
- Patrick Schroeck
- Larry Sullivan
- Dave Tran
- Steve Zemblowsky

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “ITSO Redbook Evaluation” on page 135 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

  For Internet users `http://www.redbooks.ibm.com/`
  For IBM Intranet users `http://w3.itso.ibm.com/`

- Send us a note at the following address:

  `redbook@us.ibm.com`

Chapter 1. Introduction

This chapter contains descriptions of some of the terms used in a DBRC environment. It also describes briefly the sample application used for creating most of the examples shown in this document.

1.1 Elements of DBRC

DBRC uses three types of data sets:

- The VSAM Recovery Control (RECON) data set:

  It contains the information that DBRC uses to control the recovery of IMS database data sets (DBDS).

- The Data Set Log (DSLOG):

  It contains the change records and the permanent I/O errors for an individual DBDS. This type of data set will not be supported by future releases of IMS.

- A partitioned data set (PDS):

  It contains skeletal job control language (JCL) used by DBRC to generate JCL (GENJCL function).

The RECON data set holds information about recovery-related events and data sets used by the IMS recovery utilities.

The generic term Recovery Utilities refers to the following IMS utilities:

- Database Image Copy
- Change Accumulation
- Online Database Image Copy
- Batch Backout
- Database Recovery
- Log Archive
- Log Recovery
- Data Set Log (DSLOG)

1.2 DBRC Environments

DBRC is automatically included with IMS, but the user can choose the level of control that DBRC must provide. Three possible DBRC environments can be set up:

- Log Control
- Recovery Control
- Share Control

1.2.1 Log Control

The log control environment is used to allow DBRC to control IMS logging. With log control, the user is not obliged to register the databases.

Log control is required for an IMS online system. In a batch system, log control is optional.

Note: Because the log control environment is very simple and a subset of the others, this document deals only with the other two types of control.

1.2.2 Recovery Control

The recovery control environment records information about data sets related to recovery of the databases. Recovery control includes all the functions of log control. The GENJCL function for creating recovery related JCL is available. The authorization process, for protecting the databases from external use, is not enforced.

1.2.3 Share Control

The share control environment is the full DBRC environment. Log control and recovery control are included in this environment. Authorization access is implemented with share control. This kind of environment is mandatory when using data sharing of any type.

Important: The recovery control environment does not protect the databases from use outside of DBRC control. Therefore it is highly recommended that share control be used.

Even if the user has no plans to use data sharing, it is still worthwhile to choose this type of DBRC environment. As long as no data sharing is required, use share control and register all databases with a share level of 0 (no sharing).

1.3 Definition of Terms

This section contains the most common terms used throughout this document and in other manuals about DBRC. If more details are required, use the appropriate reference manuals.

1.3.1 Database Authorization

A DBRC sharing environment introduces a new concept of database authorization. This process determines if a subsystem (IMS online, CICS/OS, or IMS batch) can have access to the requested databases. DBRC authorizes or refuses to authorize the databases, depending on the current authorizations and the access intent of the subsystem.

1.3.2 Access Intent

Access intent is determined by DBRC when a subsystem tries to allocate a database:

- For a batch job, DBRC uses the processing option (PROCOPT) of the PSB for each database to determine the access intent. If the SB has multiple PCBs for the same database, the highest intent for that database is used.
- For an IMS DC on-line system, the ACCESS parameter of the DATABASE macro sets the access intent.
- For a CICS system, the ACCESS parameter of the DFHDLDBD macro sets the access intent.

There are four processing intent attributes. They are listed below in reverse order, from the highest access intent (the most restrictive), to the lowest (the least restrictive):

1. **Exclusive (EX)**

   The subsystem requires exclusive access of the database and no sharing is allowed, regardless of the share options registered in DBRC.

   - PROCOPT of L or xE (batch) (where x = A,D,G,I,R)
   - ACCESS of Ex (online)

2. **Update (UP)**

   The subsystem may update the database. Even if no updates actually take place, the database is held in update mode. Any logs created with actual changes during this process are required for recovery or change accumulation.

   - PROCOPT of A,I,R,D (batch)
   - ACCESS of UP on-line

3. **Read with Integrity (RD)**

   The subsystem only reads the database, but it also checks any enqueue or lock held by other subsystems. It waits for the lock to be released before proceding.

   - PROCOPT of G (batch)
   - ACCESS of RD (online)

4. **Read without Integrity (RO)**

   The subsystem only reads the database and it does not check any lock or enqueue held by other subsystems.

   - PROCOPT of GO (batch)
   - ACCESS of GO (online)

1.3.3 Database Groups

DBRC can group databases into logical groups to control the backup and recovery of these databases. This ensures integrity between them. These could be logically related databases, or databases which are kept in line by programming. This typically includes a database and all of its indices (primary and secondary). A single GENJCL command can be used to create a multiple step job or multiple jobs to back up or recover all the databases in that group.

1.3.4 Database Registration

A database can take part in a sharing environment only if it is registered with DBRC.

Should the installation have multiple systems, each with a unique set of RECONs, then a database should be registered to only one set of RECONs.

1.3.5 Concurrent Database Update

Program Isolation (**PI**) of IMS or IMS Resource Lock Manager (**IRLM**) allow different dependent regions to concurrently update the same databases.

The mechanism used to control this concurrent update is called:

- **Enqueue/Dequeue** when the manager is PI
- **Locking** when the manager is IRLM

1.3.6 Program Isolation and Enqueue

The protocol of enqueueing segments with PI depends on the segment type involved.

- **Root Level**

  A complete database record (a root segment and all its dependents) is enqueued when:

  - A PSB is currently positioned on the root of the database record.
  - An update to the root has taken place.

  This means that a PCB from another program is prevented from accessing that database record while it is enqueued.

- **Dependent Segment Level**

  When a program updates a dependent segment, only the segment and not the entire database record is enqueued until sync point.

1.3.7 Locking Performance Improvements

With IMS Version 2, enhancements have been made to the locking protocol when IRLM is used. For further details, refer to IMSVS/ Version 2 Installation Notebook under the topics:

- Data Sharing Availability Enhancements
- Locking and Data Sharing Performance Improvements

1.3.8 Locking Segments Using IRLM

With IMS Version 2, the mechanism of locking segments, when IRLM is used as a lock manager, has been changed. After these changes, when a program updates a dependent segment, all the database record is locked, because the root is held at the single update level.

This change improves performance. Concurrency may have been reduced because no other program can access the database record until the program which performed the update commits the changes.

1.3.9 “Q” Command Code Processing

The “Q” command code can also be used in DL/I calls to lock segments. It is used by a program to ensure that no other program modifies a segment while the first program moves its position to other database records. The segments locked by the “Q” command code are not freed (regardless of whether the update has occurred or not) until:

- An explicit DEQ call with the same class is issued.
- A commit point is reached.
- The program terminates normally.

1.3.10 PROCOPT=GO

In an online environment it is critical that the enqueue (or locking) be freed as quickly as possible. If online transactions are not waiting for enqueue (or locking) to be released, the transaction response time is not necessarily increased.

PROCOPT=GO (get only) means that database enqueue (or locking) is not occurring. If a program uses the GO option against a database that is updated by another program, status codes may be returned and the program should handle those error situations.

GO options should be used only if all programs accessing the database are read only.

1.3.11 When Database Segments Are Released

If no updates occurred, the database record is released when the PCB moves to another database record.

If updates have occurred, the changed segments are released only when a commit point is reached.

1.3.12 Commit Points

A commit point is the point at which data input/output queues are committed. Since then, these changes are definitive and no automatic blackout can be performed. Commit points occur at the following times:

- Checkpoint call.
- Transaction/Program completion.
- SYNC point. A GU is issued on the IO PCB for a transaction defined as single mode processing (SNGL) in the IMS generation.

1.4 Data Sharing

DBRC data sharing facilities allow application programs, running in different subsystems, to concurrently process the same IMS databases.

There are two possible types of database sharing:

- Database level sharing
- Block level sharing

1.4.1 Database Level Sharing

Database level sharing is an environment where multiple subsystems can concurrently access IMS databases. In this environment the following cases can exist:

- One of the subsystems can have update access (UP), while others can have read access (RO).
- Many subsystems can have read access (RD or RO) when no other subsystem has update access.

The subsystems may be on the same MVS system or on another MVS system. The access authorization is based on access intent. The database integrity is ensured by DBRC (a prerequisite).

1.4.2 Block Level Sharing

Block level sharing is the sharing environment where more than two subsystems may have update access to the same database. The locking manager must be the IRLM. For this kind of sharing environment, the access is controlled at block or CI Level rather than at a database level.

1.5 Description of the Sample Application

A sample application has been created to provide “real” examples for this bulletin. All examples have been run on an IMS 2.2 system. To better understand the examples, a description of the sample application follows.

1.5.1 Databases

The sample application consists of:

- Five physical databases
- Two logical databases
- Two application programs which run in online and batch mode

From an application point of view, the databases are used to manage a system consisting of:

- Courses
- Class room locations
- Students enrolled in courses
- Instructors to teach each course
- Course prerequisites

The database structure consists of two physical HIDAM databases logically related and one secondary index database.

DBDNAME	Description
DBGAMAP	HIDAM database for student information, logically related to DBGAMBP
DBGAMAX	Primary index to DBGAMAP database
DBGAMBP	HIDAM database to course information, logically related to DBGAMAP
DBGAMBX	Primary index database to DBGAMBP
DBGAMBY	Secondary index database to DBGAMBP
DBGAMEL	Logical database with the root segment of DBGAMAP as the root segment
DBGAMFL	Logical database with the root segment of DBGAMBP as the root segment.

1.5.2 Log Data Sets

When running in batch mode, DASD logging is used in the sample application. Single logging has been chosen and no archiving or manipulation of the log data sets is done, dynamic blackout is also invoked when DL/I failures occur. If required, the programs can be restarted from a given checkpoint.

For most of the tests, IRLM was not used unless specifically stated.

The IMS online system has the following characteristics:

- Dual Online Data Set (OLDS) logging
- Archives to SLDS and RLDS on DASD every time an OLDS switches
- Dynamic allocation of RECONs and databases
- RECON status of SHARE CONTROL used unless specifically stated otherwise

1.5.3 Database Registration

Each database in the sample is registered to only one DBDSGRP. One group is used (DBGGRP1).

All GENJCL functions are issued using the DBDSGRP rather than any individual database.

One change accumulation group is used for all the databases (DBGCA).

The registration information is sometimes changed in order to create a specific situation.

Chapter 2. RECON Data Set

The RECON data set is the most important data set for the operation of DBRC and data sharing. The RECON data set holds all resource information and event tracking information used. See Figure 1 below.

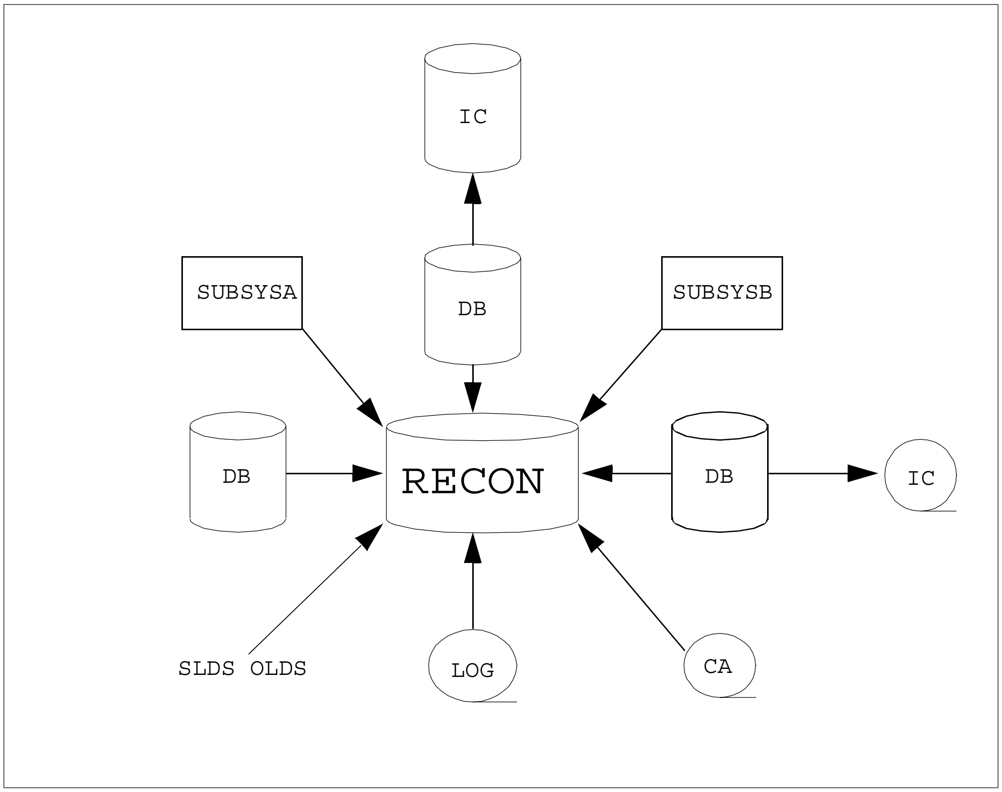


Figure 1. The RECON Data Set

Note: For more details about the record types found in the RECON data set, see Chapter 4, "RECON Record Types" on page 43.

2.1 Number of RECON Data Sets

The RECON data set can consist of one, two, or three data sets:

1. The original data set
2. The copy of the original data set
3. The spare data set

Important: The best solution, from an availability point of view, is to use all three data sets. This is strongly recommended. Using three data sets for the RECON causes DBRC to use them in the following way:

- The first data set is known as copy1. It contains the current information. DBRC always reads from this data set, and when some change has to be applied, the change is written database first to this data set.

- The second data set is known as copy2. It contains the same information as the copy1 data set. All changes to the RECON data sets are applied to this copy2 only after the copy1 has been updated.
- The third data set (the spare) is used in the following cases:
  - A physical I/O error occurs on either copy1 or copy2.
  - DBRC finds, when logically opening the copy1 RECON data set, that a spare RECON has became available, and that no copy2 RECON data set is currently in use.
  - The following command is executed:

    ```
    CHANGE.RECON REPLACE(RECONn)
    ```

  When the third RECON data set is used, the remaining valid data set is copied to the spare. When the copy is finished the spare becomes whichever of the data sets was lost, missing or in error.

Note: From the RECON point of view, the copy1 and the copy2 are normally identified by a 1 or a 2 in a field of the RECON header information.

2.2 NONEW and STARTNEW Parameters

When the RECON data set is initialized, either of the following parameters can be specified:

1. **STARTNEW** means that new jobs can be started even when only one RECON data set is available.
2. **NONEW** means that a subsystem can continue running with only one RECON data set, but that no new subsystems can start unless at least two RECON data sets are available.

Important: The use of the *NONEW* parameter is strongly recommended. The *STARTNEW* parameter should be used for emergency situations only.

Note: The existence of more than one RECON data set is only known by the RECON access routines. For most purposes the RECON data set can be considered as a single data set. In this bulletin, it is very often referred to as a single data set.

2.3 Groups of RECON

The basic rules for determining the requirement for groups of RECON data sets are explained in this section.

2.3.1 Database Related Information

A database and its associated data sets should only be defined in one RECON data set.

The fundamental principle behind the RECON data set is to store all recovery related information for a database in one place. It is not possible to use multiple RECON data sets in the recovery processing for the same database.

2.3.2 Subsystem

An IMS or CICS subsystem can only be connected to one set of RECON data sets.

All databases that are accessed by IMS DC or CICS subsystems under the control of DBRC must be registered in the RECON referenced by the online subsystem.

All batch subsystems that access any database accessed by the online subsystem must reference the same RECONs referenced by the online subsystem.

2.3.3 Database Name

The database names (DBD names) defined in one RECON data set must all be unique.

The database records, stored in the RECON data set, are registered with a key based on the DBD name. Therefore, DBRC cannot be used to control both test and production databases, using the same RECON data sets, unless some naming convention is adopted.

The rule can be simplified as follows. More than one RECON data set is necessary if all the following conditions are true:

1. Multiple versions of the same database exist (for example, test and production).
2. The same DBD name is used for the different versions of the database.
3. At least one version of the database must be registered to DBRC in the RECON data set.

The application of the previous rules usually results in the need for at least two different sets of RECON data sets, one shared between the production subsystems and one for the test subsystems.

Note: On the INIT.DBDS command, which is used to create the database data set record in the RECON, the user must supply the database data set name (DSN). When IMS opens the database, DBRC checks the DSN against the name registered in the RECON. If this name does not match, DBRC treats this database as if it was not registered. In this case, the test database—with a DSN different than the production database, even if with the same DBD name and data set name—can coexist with the production environment, but not under the control of the DBRC.

2.3.4 Sharing Environment

In a sharing environment every set of RECONs must have its own IRLM.

This is because the Global DMB number, which is used as part of the IRLM lock, is generated and maintained in the RECON.

2.4 RECON Definition and Creation

The RECON data sets are VSAM KSDSs. They must be created by using the VSAM AMS utilities.

The same record size and CI size must be used for all the RECON data sets.

The RECON data sets should be given different FREESPACE values so that CA and CI splits do not occur at the same time for both active RECON data sets.

For availability, all three data sets should have different space allocation specifications. The spare data set should be at least as large as the largest RECON data set. Figure 2 shows an example of the RECON data set definition that was used to define the RECON for the hands-on.

```
DELETE STIMS220.RECONB

SET LASTCC=0

DEFINE CLUSTER (NAME(STIMS220.RECONB) -
       VOLUMES (SBV010)              -
       INDEXED                       -
       KEYS (24 0)                   -
       CYLINDERS C5 2)               -
       RECORDSIZE (128 4089)         -
       NONSPANNED                    -
       FREESPACE (30 80)             -
       CISZ(4096) BUFSP(24576)       -
       NOREUSE                       -
       NERAS SPEED REPL IMBD         -
       UNORDERED                     -
       SHAREOPTIONS (3 3))           -
INDEX (NAME(STIMS220.RECONB.INDEX))  -
 DATA (NAME(STIMS220.RECONB.DATA))
```

Figure 2. Example of RECON Data Set Definition

2.5 Cataloging RECON Data Sets

Cataloging the RECON data sets is one of the most important steps with a multi-host environment that allows access to the RECON data sets from multiple subsystems.

2.5.1 ICF Catalog

The RECON data sets should be cataloged using ICF catalogs.

Important: Each RECON data set should be cataloged in a separate ICF catalog for maximum availability. Thus, the loss of one ICF catalog will only cause the loss of one RECON data set.

2.5.2 Deadlock Situation

It is possible to get into a shared DASD deadlock situation if a RECON data set and other VSAM data sets reside on the same volume, and if they are cataloged in an ICF catalog that resides on another volume. The deadlock situation can occur as follows:

1. **Host A:** DBRC issues a RESERVE for a RECON data set. A hardware reserve is issued for the volume containing the RECON data set, and the reserve is successful. Host A now owns the RECON volume.
2. **Host B:** A normal VSAM OPEN is issued. VSAM OPEN issues a RESERVE for the catalog (BCS), and the reserve is successful. Host B now owns the BCS volume. VSAM OPEN then issues a RESERVE for the VVDS on the RECON volume. A hardware reserve is issued for the volume. Since the volume is owned by Host A, Host B waits for the volume.
3. **Host A:** DBRC issues a VSAM OPEN for the RECON data set. VSAM OPEN issues a RESERVE for the catalog (BCS). Since Host B owns the BCS volume, Host A waits.

The result is that Host A is waiting for a resource owned by Host B, and Host B is waiting for a resource owned by Host A. Both tasks are now involved in a deadlock that can only be resolved by cancelling one of the tasks.

To prevent this kind of problem, follow these rules:

- Each RECON data set must be placed on the same volume as the ICF catalog BCS and VVDS components in which the RECON data set is cataloged.
- The volume housing those data sets should contain no other VSAM data sets.

2.6 RECON Data Sets and RACF

If the system is using RACF or other similar security package, the user should try to protect both the RECON data sets and their ICF catalogs using ***GENERIC*** **PROFILES.**

The use of generic profiles reduces the accesses to the RACF data set during the physical open process of the RECON.

2.7 Initializing RECON Data Sets

After the RECON data sets are created, they must be initialized by using the **INIT.RECON** command of the DBRC recovery control utility. This causes the RECON header records to be written in both current RECON data sets.

The RECON header records must be the first records written to the RECON data sets because they identify the RECON data sets to DBRC.

When the **INIT.RECON** command is used to initialize the RECON, specify either the RECOVCTL parameter or the SHARECTL parameter to select either:

- Recovery control environment
- Share control environment

Once one of these environments has been selected, it applies to all databases.

2.8 Allocation of RECON Data Sets to Subsystems

To allocate the RECON data set to an IMS subsystem, the user must choose one of the following two ways:

- Point to the RECON data sets by inserting the DD statements in the start-up JCL for the various subsystems.
- Use dynamic allocation.

If a DD statement is specified for RECON, DBRC does not use dynamic allocation. Otherwise, DBRC uses dynamic allocation.

With multiple subsystems sharing the same databases and RECON data sets, dynamic allocation for both the RECON data sets and the associated databases should be used. This ensures that:

- The correct and current RECON data sets are used.
- The correct RECON data sets are associated with the correct set of databases.

It also allows recovery of a failed RECON data set, since DBRC dynamically de-allocates a RECON data set if a problem is encountered with it.

To establish dynamic allocation, a special member naming the RECON data sets must be added to IMS RESLIB or to an authorized library that is concatenated to IMS RESLIB. This is done using the IMS DFSMDA macro. Figure 3 shows an example of the required macros for dynamic allocation of the RECON data sets.

```
//DYNALL JOB..
//STEP     EXEC IMSDALOC
//SYSIN    DD  *
     DFSMDA TYPE=INITIAL
     DFSMDA TYPE=RECON,DSNAME=PROD.RECONO1,
                  DDNAME=RECON1
     DFSMDA TYPE=RECON,DSNAME=PROD.RECONO2,
                  DDNAME=RECON2
     DFSMDA TYPE=RECON,DSNAME=PROD.RECONO3,
                  DDNAME=RECON3
     DFSMDA TYPE=FINAL
```

Figure 3. Dynamic Allocation of RECON Data Sets

RECON data sets are always dynamically allocated with DISP=SHR specified.

When using multiple RECON data sets (for example, test and production), be sure that each subsystem uses the correct RECON data set group. This can be done by altering the SYSLMOD DD in the procedure IMSDALOC to place the dynamic allocation parameter lists for the various RECON data set groups in different IMS RESLIBs. The appropriate RESLIB or concatenated RESLIBs must be included for each subsystem start-up JCL.

Important: When multiple processors are accessing the same RECON data set, the dynamic allocation parameter lists must be kept synchronized in the IMS RESLIBs being used by the different processors. This does not happen automatically.

Important: The usage of dynamic allocation in some subsystems and JCL allocation in others is not recommended.

2.9 Placement of RECON Data Sets

The placement of the RECON data sets in the DASD configuration is very important. The primary rule is to configure for availability. This means, for example, to place all three RECON data sets on:

- Different volumes
- Different control units
- Different channels
- Different channel directors

Review also the placement of the RECON data sets on volumes containing data sets that are processed by non-IMS utilities that issue hardware I RESERVEs.

Examples of these utilities are:

- Linkage Editor
- SUPERZAP
- DADSM

2.10 RECON Data Set Maintenance

There are several procedures and commands that can be used to maintain the RECON data set.

2.10.1 RECON Backup

Operational procedures should be set up to ensure that regular backups of the RECON data set are taken.

These backups should be performed using the BACKUP.RECON DBRC utility command. The command includes a reserve mechanism to ensure that no updating of the RECON takes place during the backup. If possible, the backup should be taken when there are no subsystems active.

The backup copy is created from the copy1 RECON data set. The command to create the backup copy invokes the AMS REPRO command, with its normal defaults and restrictions. For instance, the data set that is receiving the backup copy must be empty.

2.10.2 DELETE.LOG INACTIVE Command

The only recovery related records in the RECON data set that are not automatically deleted are the log records (PRILOG and LOGALL). These records can be deleted using the DELETE.LOG INACTIVE command. This command can be added to the job that takes a backup of the RECON data set.

A log is considered inactive when the following conditions are all true:

- The log volume does not contain any DBDS change records more recent than the oldest image copy data set known to DBRC. This check is performed on a database data set (DBDS) basis.
- The log volume was not opened in the last 24 hours.
- The log has either been terminated (nonzero stop time) or has the ERROR flag in the PRILOG and SECLOG record set on.

2.10.3 LIST.RECON STATUS Command

Regular use should be made of the LIST.RECON STATUS command to monitor the status of the individual RECON data sets.

Using the LIST.RECON command produces a formatted display of the contents of RECON. The copy1 RECON data set is used as a source. DBRC ensures that the second RECON data set contains the same information as the first RECON data set.

The optional parameter **STATUS** can be used to request the RECON header record information and the status of all RECON data sets. The use of this parameter suppresses the listing of the other records.

This command should be executed two or three times a day during the execution of an online system, to ensure that no problems have been encountered with these data sets.

2.11 RECON Reorganization

In addition to the regular backups, procedures to monitor utilization of the RECON data sets space should be put in place.

Since all current levels of VSAM support CI reclaim (and DBRC does not turn it off), the requirement to reorganize RECONs to reclaim space has diminished. For instance, when all the records in a CI have been erased, the CI is returned to the free CI pool in the CA. Some users have decided to perform a monthly reorganization.

A plan for reorganizing the RECON data sets to reclaim this space on a regular basis must be considered. The RECON data sets can be reorganized while the IMS online systems are active

2.12 Reorganizing RECON Data Sets

The RECON data sets can be reorganized easily and quickly with the use of a few DBRC and AMS commands. The AMS REPRO command copies one RECON data set to another, reorganizing it during the process. This command, combined with a DELETE and a DEFINE of the RECON data sets, is enough to complete a reorganization.

Additional information to consider when designing the RECON reorganization procedures, related to the IMS DC status, are as follows:

- If the online system is *active*:

  A reorganization of the RECON data sets should be scheduled:

- During a period of low RECON activity.
- When no BMPs are running.
- A LIST.RECON STATUS command must be issued from each online system which uses the RECON data sets, after the CHANGE.RECON REPLACE command is issued, in order to de-allocate the RECON before deleting and defining it again.

- If the online system is *not active*:

  A reorganization of the RECON data sets should be scheduled:

  - After a BACKUP.RECON has been taken.
  - When no subsystems are allocating the RECON data sets.

2.13 Recreating RECON Data Sets

The RECON data sets may need to be recreated, for instance:

- In a disaster recovery site
- After the loss of all the RECON data sets when no current backup is available

Recreating the RECON can be a long and slow process. When designing procedures to handle this process, there are two basic alternatives:

- Restore the RECON from the last backup (if available) and update it to the current status required.
- Recreate and reinitialize the RECON data sets.

Both of these procedures have advantages and disadvantages. Which alternative is best suited for an installation depends on:

- The time frame in which the system must be recovered and available
- The point-in-time to which it is acceptable to recover
- The type of processing environment (24 hours online availability or batch)
- The level of DBRC usage:
  - Log control
  - Recovery control
  - Share control

Further Details for Restoring RECON Data Sets:

Before deciding to recreate the RECON data sets from scratch, the following details must be well understood:

- GENJCL functions are normally used to create procedures.

  Without the RECON information, recovery procedures cannot be generated until the RECON information is correct. Likewise, image copy procedures cannot be generated until the database and image copy data set information has been recreated.

- Recreation of DB, DBDS, DBDSGRP and CAGRP information must be available.

If the original INIT commands were retained, then the registration can be done easily. Changes made with CHANGE commands must somehow be recorded and reapplied.

The DBDSGRP and CAGRP information is critical because any recovery image copy or change accumulations JCL generated can cause serious problems if incorrectly specified.

- Volume serial information is available.

  Unless cataloged data sets are used and the PTF for cataloged data set usage has been applied (see the discussion of DFHSM under 3.11, "DFHSM Considerations" on page 39 for more details), the volume serials of all image copies and log data sets must be corrected.

- Image copy time must be adequate.

  If the databases are restored with non-IMS utilities (pack restores), then the time required to take an image copy or to notify DBRC of the image copy data sets, also restored with pack restores, must be considered.

In summary:

- For those installations using only Log control, it is probably easier to reinitialize the RECON data sets and reapply the information than to update the RECON with the changed information.
- For those installations using Recovery or Share control where the physical restoration of the databases is done outside of the DBRC control, it might be easier to reinitialize the RECON data sets.
- For those installations which require the online subsystem to be warm restarted, the only alternative is to use the latest backup of the RECON and to bring all information current to the required point-in-time.

2.14 PRILOG Record Size

One PRILOG record is created for each subsystem execution. This record must contain all the information about the log data sets created during the life of this subsystem.

In IMS Version 2, the 64K limit for a PRILOG record, that is present in IMS 1.3, has been removed. Nonetheless, the size of the PRILOG record should not be greater than 32,760 bytes (32K).

The record size can be larger if spanned records are used; however, the following limitations should be considered before using spanned records:

- The maximum size of a record to be used by the VSAM REPRO command is 32,760 bytes if the output is a non-VSAM data set.
- RECON backup and transfer to off-site storage is normally performed with a sequential data set.
- PRILOG records are only deleted when every RLDS and SLDS data set within that record is no longer required.

This is a problem only for those installations which have a high volume of log data sets and the requirement for a continuous operation environment.

To calculate the size of the maximum required PRILOG record, the formula in Figure 4 can be used.

S = 52 + (80 D) + (14 V)

where:

S = the size for the PRILOG/PRISLDS record in bytes

52 = the required prefix of the PRILOG record

80 = the required number of bytes for each SLDS/RLDS entry

D = the number of SLDS/RLDS data sets created from archive for this execution of the subsystem

14 = the required number of bytes for each volume that contains SLDS/RLDS data sets

V = the number of volumes that can contain SLDS/RLDS data sets

Figure 4. PRILOG Record Size Calculation Formula

For example, assume that an installation has the following characteristics:

- An online subsystem is running for 23 hours a day.
- The subsystem fills up an OLDS every 30 minutes.
- Each OLDS is archived to one RLDS and one SLDS.
- There are 2 volumes that can contain RLDS or SLDS data sets.
- There are 46 RLDS and 46 SLDS data sets each day.

Using the formula in Figure 5, the size of the PRILOG record for this example is:

```
S  =  52  +  (80 D)   +   (14 V)
   =  52  +  (80 92)  +   (14 2)
   =  52  +  (7360)   +   (28)
S  =  7440
```

Figure 5. PRILOG Record Size Calculation Formula Example 1

This is well under the maximum size, so there is no problem with this subsystem.

Assume, however, that the environment changes to allow the IMS to run 24 hours a day for 6 days before being brought down. There are 48 RLDS and 48 SLDS data sets each day, and a total of 576 for the 6 days. The calculation now becomes like Figure 6:

```
S = 52 + (80  D)   + (14  V)

  = 52 + (80  576) + (14  2)

  = 52 + (46,080)  + (28)

S = 46,160
```

Figure 6. PRILOG Record Size Calculation Formula Example 2

This is now over the suggested maximum record size. One solution is to switch to archiving after two OLDS are full. This reduces the number of RLDS and SLDS data sets by half. This brings the PRILOG record size well below the maximum size.

2.15 Summary of Recommendations for RECON Data Sets

- Use three RECON data sets—two current and one spare.
- Define the three RECON data sets with different space allocations.
- Put the RECON data sets on different devices, channels, and so on.
- Put the RECON data sets on the same volume as the ICF user catalogs.
- Use dynamic allocation.
- Do not mix dynamic allocation and JCL allocation.
- Define the RECON data sets for AVAILABILITY, but keep performance implications in mind.

Chapter 3. Data Set Management

This chapter explains the two most commonly used types of database backups:

- Image copy
- Online image copy

The chapter also deals with:

- Log data sets
- Change accumulation data sets

Together with the image copy data sets, these data sets are essential for the recovery mechanism.

3.1 Database Image Copy

DBRC provides full support for standard image copies. It automatically captures the data set name, the volume serial number and file sequence number for the image copy output data set when the image copy utility is run. It registers this information in the RECON data set.

Database image copy (IC) is a batch job that is run with an access intent of read with integrity (RD). This permits processing in parallel with other subsystems with an access intent of RO or RD (and a SHARELVL greater than 0).

DBRC ensures that the image copy utility cannot be run when a subsystem is updating the database.

3.1.1 Image Copy Details

DBRC provides the following facilities to allow the user to perform and control the recording (or the deletion) of the image copy data sets.

- **Backup Group**

  For each group of related databases, either logically related or used together in an application system, specification to DBRC allows simultaneous DBRC processing for the members of the group.

- **Recovery Period**

  When a database data set is defined to DBRC through the INIT.DBDS command, a recovery period can be specified. This recovery period means that DBRC keeps details of an image copy for that data set for the specified period.

- **Number of Image Copy Generations**

  When a database data set is defined to DBRC through the INIT.DBDS command, the maximum number of image copy generations must be specified with the GENMAX parameter. This parameter provides the number of image copies that DBRC maintains for a given database data set.

- **Reusing Image Copy Data Sets**

  When defining the database data set to DBRC through the INIT.DBDS command, the parameter REUSE, for reusing an image copy output data set, can optionally be requested. If DBRC finds that the oldest record of an image copy can be reused, then the JCL for the new image copy uses that data set. See Figure 7 below.

```
//DBRC    EXEC PGM=DSPURX00
//*
//STEPLIB  DD DSN=SYSC.STIHS22O.RESLIB,DISP=SHR
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
   INIT.DB DBD(DBGAMAPl SHARELVL(2l
   INIT.DBDS DBD(DBGAMAP) DDN(DBGAMAP) DSN(STIMS220.DBG.DBGAMAPl -
        GENMAX(31 ICJCL(DBGIC) OICJCL(DBGOIC) RECOVJCL(DBGRECOVl
        RECOVPD(1l REUSE
   INIT.DB DBD(DBGAMGXl SHARELVL(2l
   INIT.DBDS DBD(DBGAMAX) DDN(DBGAMAX) DSN(STIMS220.DBG.DBGAMAX) -
        GENMAX(3) ICJCL(DBGIC) OICJCL(DBGOIC) RECOVJCL(DBGRECOVl
        RECOVPD(1) REUSE
   INIT.DB DBD(DBGAMBPl SHARELVL(2)
   INIT.DBDS DBD(DBGAMBP) DDN(DBGAMBP) DSN(STIMS220.DBG.DBGAHBP) -
        GENMAX(31 ICJCL(DBGIC) OICJCL(DBGOICl RECOVJCL(DBGRECOV)
        RECOVPD(1) REUSE
   INIT.DB DBD(DBGAMBX) SHARELVL(2)
   INIT.DBDS DBD(DLGAMBX) DDN(DBGAMBXl DSN(STIMS220.DBG.DBGAMBX) -
        GENMAX(3) ICJCL(DBGICl OICJCL(DBGOIC) RECOVJCL(DBGRECOV)
        RECOVPD(1) REUSE
   INIT.DB DBD(DBGAMBY1 SHARELVL(2)
   INI7.DBDS DBD(DBGAMBY) DDN(DBGAMBY1 DSN(STIMS220.DBG.DBGAMBYl-
        GENMAX(31 ICJCL(DBGIC) OICJCL(DBGOICl RECOVJCL(DBGRECOV)
        RECOVPD(1l REUSE
   INIT.IC DBD(DBGAMAP) DDN(DBGAMAP) ICDSN(STIMS220.DBG.DBGAMAP.IC1)
   INIT.IC DBD(DBGAMAX1 DDN(DBGAMAX) ICDSN(STIMS220.DBG.DBGAMAX.IC1l
   INIT.IC DBD(DBGAMBPl DDN(DBGAMBP) ICDSN(STIMS220.DBG.DBGAHBP.IC1)
   INIT.IC DBD(DBGAMBX) DDN(DBGAMBXl ICDSN(STIMS220.DBG.DBGAMBX.IC1)
   INIT.IC DBD(DBGAMBY) DDN(DBGAMBY) ICDSN(STIMS220.DBG.DBGAMBY.IC1)
   INIT.IC DBDfDBGAMAP) DDN(DBGAMAP1 ICDSN(STIMS220.DBG.DBGAMAP.IC2)
   INIT.IC DBD(DBGAMAX1 DDN(DBGAMAX) ICDSN(STIMS220.DBG.DBGAMAX.IC2)
   INIT.IC DBD(DBGAMBP) DDN(DBGAMBP) ICDSN(STIMS220.DBG.DBGAMBP.IC2l
   INIT.IC DBD(DBGAMBX) DDN(DBGAMBX) ICDSN(STIMS220.DBG.DBGAMBX.IC2)
   INIT.IC DBD(DBGAMBY) DDN(DBGAMBY) ICDSN(STIMS220.DBG.DBGAMBY.IC2)
   INIT.IC DBD(DBGAMAPl DDN(DBGAMAPl ICDSN(STIMS220.DBG.DBGAMAP.IC3)
   INIT.IC DBD(DBGAMAX) DDN(DBGAMAX) ICDSN(STIMS220.DBG.DBGAMAX.IC3)
   INIT.IC DBD(DBGAMBP) DDN(DBGAMBP) ICDSN(STIMS220.DBG.DBGAMBP.IC3)
   INIT.IC DBD(DBGAHBX) DDN(DBGAHBX) ICDSN(STIMS220.DBG.DBGAMBX.IC3)
   INIT.IC DBD(DBGAMBY) DDN(DBGAMBY) ICDSN(STIMS220.DBG.DBGAMBY.IC3l
```

Figure 7. Defining DBDS to RECON

3.1.2 REUSE Option

When the REUSE option is specified, consider the following details:

- If a recovery period has been specified for an image copy data set, then the image copy data set is not available for re-use until the recovery period has expired.
- The number of image copy data sets registered can be greater than the GENMAX value, but the recovery period (RECOVPD) may not yet have expired. In this case, an available predefined data set is used, a message is issued, and the GENMAX value is increased.
- The number of image copy data sets required should be enough to handle the maximum number of image copies taken in the recovery period of the database data set.
- DBRC selects an unused image copy data set until the GENMAX limit is reached before reusing any other.

- An image copy data set is not reused, even when the retention period is expired, if the number of image copy data set registered is less then the GENMAX parameter.
- If the oldest IC copy cannot be reused because its RECOVPD is not expired, a message is issued and the processing is stopped.

Note: Figure 7 shows all the commands used to register the sample databases, database data sets, and image copies. The REUSE option was used for the tests of this bulletin because it allowed an easier control on the outputs generated. The experience of the authors of this bulletin is that customers prefer to use the NOREUSE option.

3.1.3 NOREUSE Option

Generation Data Set Groups (GDGs) can be used for registering DB image copies. When using generation data set group (GDG) for the image copy data sets, the NOREUSE option must be used.

The retention of GDG data sets should also be carefully considered when defining to the system the model GDG. If GDGs are used, the maximum number of versions should be great enough to cover the worst cases. An example of one of these worst cases includes the following conditions:

- GDG is used for an image copy data set.
- The maximum number of versions has been defined when creating the model of the GDG.
- An image copy version GDG is deleted by the system because the maximum number of versions has been reached.
- DBRC still requires that version to be kept according to the chosen retention parameters.

Figure 8 on page 24 shows an example of the DBRC command and the skeletal JCL for generating image copy JCL.

Figure 9 on page 25 shows the JCL for image copy generated by the GENJCL.IC shown in Figure 8.

Note: In a data sharing environment, the databases should be available as much as possible. One way to improve availability is:

- Change the authorization to READON before running the image copy to specify that the database can be authorized for read processing only.
- When the IC is terminated, change the authorization to READOFF to specify that the database can be authorized to any type of processing.

```
DBRC COMMAND:

     GENJCL.IC GROUP(DBGGRP) ONEJOB

MEMBER(DBGIC):

//READON EXEC PGM=DSPURX00DSPURX00,COND=(O,NE)
//*
//* SET AUTHORITY TO READ ONLY WHILE IC IN PROCESS
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSINDD
   CHANGE.DB DBD(%DBNAME) READON
/*
//I%DBNAME EXEC PGM=DFSRRCOO,REGION=4096K,
//        PARM='ULU,DFSUDMPO,,,,,,,,,,,,Y'
//*
//* RUN IMAGE COPY FOR %DBNAME
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//%DBDDN   DD DSN=%DBDSN,DISP=SHR,AMP=t'BUFND=30')
//DATAOUT1 DD DSN=%ICDSN1,DISP=OLD,DCB=BUFNO=10)
//SYSIN    DD *
%ICSYSIN
/*
//READOFF EXEC PGM=DSPURX00,COND=(0,NE)
//*
//* SET AUTHORITY TO READ OFF AFTER IC COMPLETED
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
CHANGE.DB DBD(%DBNAME) READOFF
/*
```

Figure 8. Generating IC JCL (Command and Skeletal JCL)

3.1.4 SHARECTL and Concurrent Image Copy

When SHARELVL(0) is specified for a database, DBRC does not allow image copies of different data sets in the same database to be done in parallel.

Specify SHARELVL(1) or higher to bypass this problem.

```
//READON EXEC PGM=DSPURX00,COND=(O,NE)
//*
//* SET AUTHORITY TO READ ONLY WHILE IC IN PROCESS
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
CHANGE.DB DBD(DBGAMAP) READON
/*
//IDBGAMAP EXEC PGH=DFSRRC00,REGION=4096K,
//        PARM='ULU,DFSUDMP0 " " " " " " Y'
//*
//* RUN IMAGE COPY FOR DBGAMAP
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DBGAMAP  DD DSN=STIMS22O.DBG.DBGAMAP,DISP=SHR,AMP=('BUFND=30')
//DATAOUT1 DD DSN=STIMS220.DBG.DBGAMAP.IC3,DISP=OLD,DCB=BUFNO=10
//DFSVSAMP DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD *
D1 DBGAMAP DBGAMAP DATAOUT1
/*
//READOFF EXEC PGM=DSPURX00,COND=(O,NE)
//*
//* SET AUTHORITY TO READ OFF AFTER IC COMPLETED
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
CHANGE.DB DBD(DBGAMAP) READOFF
/*
//READON EXEC PGM=DSPURX00,COND=(O,NE)
//*
//* SET AUTHORITY TO READ ONLY WHILE IC IN PROCESS
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
CHANGE.DB DBD(DBGAMAX) READON
/*
//IDBGAMAX EXEC PGM=DFSDFSRC00,REGION=4096K,
//        PARM='ULU,DFSUDMP0,,,,,,,,,,,,,Y'
//*
//* RUN IMAGE COPY FOR DBGAMAX
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DBGAMAX  DD DSN=STIMS220.DBG.DBGAMAX,DISP=SHR,AMP=('BUFND=30')
//DATAOUT1 DD DSN=STIMS220.DBG.DBGAMAX.ICl,DISP=OLD,DCB=BUFNO=10
//DFSVSMP  DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD *
D1 DBGAMAX DBGAMAX DATAOUT1
/*STIMS220
//READOFF  EXEC PGM=DSPURXOO,COND=(O,NE1
//*
//* SET AUTHORITY TO READ OFF AFTER IC COMPLETED
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
CHANGE.DB DBD(DBGAMAX) READOFF
/*
//READON EXEC PGM=DSPURX00,COND=(0,NE)
//*
//* SET AUTHORITY TO READ ONLY WHILE IC IN PROCESS
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD
CHANGE.DB DBD(DBGAMBP) READON
/*
```

Figure 9. Example of JCL for an Image Copy (Part 1 of 3)

```
//IDBGAMBP EXEC PGM=DFSRRC00,REGION=4096K,
//              PARM='ULU,DFSUDMP0 " " " " " "V'
//*
//* RUN IMAGE COPY FOR DBGAMBP
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DBGAMBP  DD DSN=STIMS220.DBG.DBGAMBP,DISP=SHR,AMP=('BUFND=30')
//DATAOUT1 DD DSN=STIMS220.DBG.DBGAMBP.IC3,DISP=OLD,DCB=BUFNO=10
//DFSVSAMP DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD *
D1 DBGAMBP DBGAMBP  DATAOUT1
/*
//READOFF EXEC PGM=DSPURX00,COND=(0,NE)
//*
//* SET AUTHORITY TO READ OFF AFTER IC COHPLETED
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
CHANGE.DB DBD(DBGAMBP) READOFF
/*
READON EXEC PGM=DSPURX00,COND=(0,NE)
//*
//* SET AUTHORITY TO READ ONLY WHILE IC IN PROCESS
//*
//STEPLIB  DD DSN=SYSC.STIMS22O.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
CHANGE.DB DBD(DBGAMBX) READON
/*
//IDBGAMBX EXEC PGM=DfSRRC00,REGION=4096K,
//              PARM='ULU,DFSUDMP0 " " " " " "Y'
//*
//* RUN IMAGE COPY FOR DBGAMBX
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DBGAMBX  DD DSN=STIMS220.DBG.DBGAMBX,DISP=SHR,AMP=('BUFND=30')
//DATAOUT1 DD DSN=STIMS220.DBG.DBGAMBX.IC3,DISP=OLD,DCB=BUFNO=10
//DFSVSAMP DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD *
D1 DBGAMBX DBGAMBX DATAOUT1
/*
//READOFF EXEC PGM=DSPURX00,COND=(0,NE)
//*
//* SET AUTHORITY TO READ OFF AFTER IC COHPLETED
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
CHANGE.DB DBD(DBGAMBX) READOFF
/*
READON EXEC PGM=DSPURX00,COND=(0,NE)
//*
//* SET AUTHORITY TO READ ONLY WHILE IC IN PROCESS
//*
//STEPLIB  DD DSN=SYSC.STIMS22O.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
CHANGE.DB DBD(DBGAMBY) READON
/*
```

Figure 10. Example of JCL for an Image Copy (Part 2 of 3)

```
//IDBGAMBY EXEC PGM=DFSRRC00,REGION=4096K,
//              PARM='ULU,DFSUDMP0 " " " " " "Y'
//*
//* RUN IMAGE COPY FOR DBGAMBX
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DBGAMBY  DD DSN=STIMS220.DBG.DBGAMBY,DISP=SHR,AMP=('BUFND=30')
//DATAOUT1 DD DSN=STIMS220.DBG.DBGAMBY.IC3,DISP=OLD,DCB=BUFNO=10
//DFSVSAMP DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD *
D1 DBGAMBY DBGAMBY DATAOUT1
/*
//READOFF EXEC PGM=DSPURX00,COND=(0,NE)
//*
//* SET AUTHORITY TO READ OFF AFTER IC COHPLETED
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
CHANGE.DB DBD(DBGAMBY) READOFF
/*
```

Figure 11. Example of JCL for an Image Copy (Part 3 of 3)

3.1.5 NOTIFY.IC Command

The command NOTIFY.IC or the command NOTIFY.UIC performs a cleanup of the following recovery related records registered in RECON:

- IC
- ALLOC
- REORG
- RECOV

These commands may be issued explicitly or implicitly.

For instance, the NOTIFY.IC command is implicitly issued by executing the image copy utility.

The cleanup process, performed after an image copy utility, deletes records that are older than the oldest IC generation. DBRC can no longer use those records for recovery because there is no previous image copy that can be used with them.

This cleanup process is *not* performed by issuing the DELETE.IC command.

The DELETE.IC command only deletes a particular IC record without cleaning up all the related recovery records.

3.2 Online Image Copy

Online Image Copy (OIC) can only be run under IMS DC. In a sharing environment, no other sharing subsystem can have update authorization to the database while an online image copy of the database is being performed.

DBRC prevents the execution of an OIC if the database is authorized to a subsystem with update intent.

This utility runs as a batch message processing (BMP) program .

Figure 12 on page 29 shows an example of JCL produced for an *Online Image Copy:*

Figure 13 on page 30 shows the PSB required for OIC:

Note: The PSB contains PCB entries for the index databases that are not normally required for BMP programs.

3.3 Concurrent Image Copy

The image copy utility can run against a fast path DEDB while IMS DC is running. It requires the CIC option. It can also be used when sharing the DEDB between two or more IMS DC systems with either read or update intent.

When a recovery is run using this type of image copy, it requires the following data sets:

- The logs that were open at the time the image copy was taken
- All logs since the DEDB was opened, or since a subsequent change accumulation was performed
- The last change accumulation data set (if existing)

3.4 Other Backup Utilities

Other forms of backups are normally referred to as nonstandard image copies. Examples of other methods of backing up a database are:

- Volume dump
- AMS REPRO/EXPORT utility

Since nonstandard image copy data sets are produced by non-IMS utilities, no interface with DBRC is automatically offered.

DBRC must be informed of the user image copy (nonstandard) using the recovery utility command NOTIFY.UIC.

```
//*
//* JCL FOR ONLINE IMAGE COPY
//*
//IDBGAMAP EXEC PGM=DFSRRC00,REGION=4096K,
//              PARM=(BMP,DFSUICP0,DBGC01,,,N0000,,,,1,,0,I220,,)
//STEPLIB  DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//DFSRESLB DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DBGAMAP  DD DSN=STIMS220.DBG.DBGAMAP,DISP=SHR
//DATAOUT1 DD DSN=STIMS220.DBG.DBGAMAP.IC4,DISP=OLD,DCB=BUFNO=10
//DFSVSAMP DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD *
D1 DBGAMAP DBGAMAP DATAOUT1
//*
//IDBGAMAX EXEC PGM=DFSRRC00,REGION=4096K,
//              PARM=(BMP,DFSUICP0,DBGC01,,,N0000,,,,1,,0,I220,,)
//STEPLIB  DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//DFSRESLB DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DBGAMAX  DD DSN=STIMS220.DBG.DBGAMAX,DISP=SHR
//DATAOUT1 DD DSN=STIMS220.DBG.DBGAMAX.IC4,DISP=OLD,DCB=BUFNO=10
//DFSVSAMP DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD *
D1 DBGAMAX DBGAMAX DATAOUT1
//*
//IDBGAMBP EXEC PGM=DFSRRC00,REGION=4096K,
//              PARM=(BMP,DFSUICP0,DBGC01,,,N0000,,,,1,,0,I220,,)
//STEPLIB  DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//DFSRESLB DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DBGAMBP  DD DSN=STIMS220.DBG.DBGAMBP,DISP=SHR
//DATAOUT1 DD DSN=STIMS220.DBG.DBGAMBP.IC4,DISP=OLD,DCB=BUFNO=10
//DFSVSAMP DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD *
D1 DBGAMBP DBGAMBP DATAOUT1
//*
//IDBGAMBX EXEC PGM=DFSRRC00,REGION=4096K,
//              PARM=(BMP,DFSUICP0,DBGC01,,,N0000,,,,1,,0,I220,,)
//STEPLIB  DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//DFSRESLB DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DBGAMBX  DD DSN=STIMS220.DBG.DBGAMBX,DISP=SHR
//DATAOUT1 DD DSN=STIMS220.DBG.DBGAMBX.IC4,DISP=OLD,DCB=BUFNO=10
//DFSVSAMP DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD *
D1 DBGAMBX DBGAMBX DATAOUT1
//*
//IDBGAMBY EXEC PGM=DFSRRC00,REGION=4096K,
//              PARM=(BMP,DFSUICP0,DBGC01,,,N0000,,,,1,,0,I220,,)
//STEPLIB  DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//DFSRESLB DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DBGAMBY  DD DSN=STIMS220.DBG.DBGAMBY,DISP=SHR
//DATAOUT1 DD DSN=STIMS220.DBG.DBGAMBY.IC4,DISP=OLD,DCB=BUFNO=10
//DFSVSAMP DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD *
D1 DBGAMBY DBGAMBY DATAOUT1
//*
```

Figure 12. OIC JCL Produced by GENJCL.OIC (Part 1 of 2)

```
         PCB   TYPE=DB,DBDNAME=DBGAMAP,                                X
               PROCOPT=G,KEYLEN=24
               SENSEG NAME=AMASTUD,PARENT=0
               SENSEG NAME=AMACLAS,PARENT=AMASTUD
               SENSEG NAME=AMAADDR,PARENT=AMASTUD
*
         PCB   TYPE=DB,DBDNAME=DBGAMAX,                                X
               PROCOPT=G,KEYLEN=24
               SENSEG NAME=STUDINDX,PARENT=0
         PCB   TYPE=DB,DBDNAME=DBGAMBP,                                X
               PROCOPT=G,KEYLEN=24
               SENSEG NAME=AMBCLAS,PARENT=0
               SENSEG NAHE=AMBSTUD,PARENT=AMBCLAS
               SENSEG NAHE=AMBLOC,PARENT=AMBCLAS
               SENSEG NAHE=AMBPREQ,PARENT=AMBCLAS
               SENSEG NAME=AMBINSTR,PARENT=AMBCLAS
*
         PCB   TYPE=DB,DBDNAME=DBGAHBX,                                X
               PROCOPT=G,KEYLEN=24
               SENSEG NAME=CLASINDX,PARENT=0
*
         PCB   TYPE=DB,DBDNAME=DBGAMBY,                                X
               PROCOPT=G,KEYLEN=24
               SENSEG NAHE=AMBYl,PARENT=0
         PSBGEN PSBNAME=DBGCOl,                                     X
               LANG=ASSEM                                           X
               OLIC=YES,                                            X
               CMPAT=YES
         END
```

Figure 13. OIC JCL Produced by GENJCL.OIC (Part 2 of 2)

3.5 Batch Log

An IMS batch subsystem can create a System Log Data Set (SLDS) on DASD or tape. If the subsystem runs with DBRC active, then the RECON is updated with the correct information concerning this SLDS. This data set must be retained, for recovery purposes, until an image copy of all the updated databases has been performed.

DCB information should be specified on the IEFRDER DD statement pointing to the log data set. If no LRECL and BLKSIZE are specified on the IEFRDER DD, then a default of 1K is used (a pre-allocated data set is re-blocked to 1K).

To ensure efficient DASD and TAPE usage, the DCB parameters should be specified on *every* IEFRDER DD statement, as shown in Figure 14.

```
//IEFRDER DD DSN=STIHS22O.DBG.BOlLOG,DISP=(,CATLG,CATLG),
UNIT=SYSDA,SPACE=(TRK,(100,100),RLSE7,
DCB=(RECFH=VB,LRECL=23468,BLKSIZE=23472l
```

Figure 14. Example of IEFRDER DCB Information

If DASD SLDS are used and there is a need to migrate them to tape, the log archive utility should be used. This should be run with DBRC active to keep the RECON information properly up-to-date.

DBRC uses the archived SLDS during recovery processing even though the original SLDS is still available. Once the SLDS has been archived, the original DASD version should be deleted.

Note: Although most of the IMS literature states that batch jobs (and CICS) produce SLDSs, as far as DBRC is concerned, these subsystems produce RLDSs. DBRC PRILOG records describe RLDSs; and PRISLDS records describe SLDSs.

Figure 15 shows an example of the JCL to archive a batch SLDS.

```
//AR00001 EXEC PGM=DFSUARCO,PARM='DBRC=YES',REGION=4096K
//*
//*        JCL TO ARCHIVE BATCH SLDS
//*
//STEPLIB    DD DSN=SYSC.STIMS22X.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=
//DFSSLDSP DD DSN=STIMS22X.DBG.B01LOG(0),DISP=SHR,DCB=BUFNO=30
//DFSSLOGP DD DSN=STIMS22X.DBG.B01ARCH(+1),
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(STIMS22X.DCB,LRECL=22524,BLKSIZE=22528,
//            RECFM=VB,BUFNO=30),
//            UNIT=3480,V0L=SER=(TAPE01)
//SVSIN DD
SLDS
/*
//*
```

Figure 15. Archive of Batch SLDS

When the batch job runs without DBRC, the archive utility must also be run without DBRC.

To complete the recovery information on the RECON data set, perform the following steps:

- Run the recovery control utility with a NOTIFY command to update the RECON with the log information.
- Issue a NOTIFY.ALLOC command for each DBDS that has update records on this log.

3.6 Online Log

Online log data sets are archived to two types of log data sets:

- System Log Data Set (SLDS)
- Recovery Log Data Set (RLDS).

The archive utility can create both data sets at the same time. The use of RLDS data sets for recovery shortens the recovery time and saves DASD or tape usage.

Note: This section applies only to IMS DC, not to CICS.

3.6.1 System Log Data Set

The SLDS is produced as a result of archiving an OLDS. The SLDS contains all the information from the OLDS, unless some log records are specifically excluded during the archive processing. Figure 16 shows an example of a GENJCL.ARCHIVE command to archive an OLDS to SLDS and RLDS data sets.

3.6.2 Recovery Log Data Set

During the archive operation of a log data set, the creation of an output data set, containing all the log records needed for database recovery, can be requested. The output data set is referred to as an RLDS.

DBRC will record the creation of an RLDS. If the input data set contains records needed for a database recovery, it will be used in place of the SLDS by the GENJCL function. The same is also true for a GENJCL command for a change accumulation.

If the RLDS contains no records needed for database recovery, DBRC records the data set name and volume serial number of the SLDS, in place of the RLDS.

```
DBRC Command:

   GENJCL.ARCHIVE SSID(I220) MAXOLDS(1)

MEMBER (ARCHJCL):

//AR%STPNO EXEC PGM=DFSUARC0,PARM='%SSID',REGION=4096K
//*
//* JCL FOR ARCHIVE UTILITY
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
%SELECT OLDS(%SSID,(%DDNAMES))
//%OLDSDDN DD DSN=%OLDSDSN,DISP=SHR,DCB=BUFNO=20
%ENDSEL
//DFSSLOGP DD DSN=STIMS220.SLDS(+1),
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(STIM22X.DCB,LRECL=22524,22528),
//            UNIT=3390,VOL=SER=(STIMS6),
//            SPACE=(TRK,(15,15),RLSE),
//RLDSDD1  DD DSN=STIMS220.RLDS(+1),
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(STIM22X.DCB,LRECL=22524,22528),
//            UNIT=3390,VOL=SER=(STIMS6),
//            SPACE=(TRK,(15,15),RLSE),
//SYSIN    DD *
SLDS FEOV(08000)
COPY DDNOUT1 (RLDSDD1) DBRECOV
/*
```

Figure 16. Archive OLDS to SLDS and RLDS Data Sets

The generated JCL from the GENJCL command in Figure 16 is shown in Figure 17.

```
//AR00001 EXEC PGH=DfSUARCO,PARH='I220',REGION=4096K
//*
//* JCL FOR ARCHIVE UTILITY
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//DFSOLP00 DD DSN=STIMS220.OLP00,DISP=SHR,DCB=BUFNO=30
//DFSOLS00 DD DSN=STIMS220.OLS00,DISP=SHR,DCB=BUFNO=30
//DFSSLOGP DD DSN=STIHS22O.SLDS(+l),
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(STIMS220.DCB,LRECL=22524,8LKSIZE=22528,
//            RECFM=VB,BUFNO=30),
//            UNIT=3380,V0L=SER=(STIMS6),
//            SPACE=(TRK,(15,151,RLSE)
//RLDSDD1  DD DSN=STIMS220.RLDS(+l),
//            DISP=tNEW,CATLG,DELETEl,
//            DCB=(STIHS22O.DCB,LRECL=22524,8lKSIZE=22528,
//            RECFM=VB,BUFNO=30),
//            UNIT=3380,V0L=SER=(STIMS5),
//            SPACE=(TRK,(15,15),RLSE)
//SYSIN    DD *
SLDS FEOV(08000)
COPY DDNOUT1(RLDSDDl) DBRECOV
/*
```

Figure 17. Generated JCL for Archive OLDS to SLDS and RLDS

3.7 Rebuilding PRISLDS/PRILOG Records with Multiple Data Set Entries

There are no examples in the standard DBRC manuals with regard to adding information about PRISLDS and PRILOG records that have multiple data set entries to the RECON.

Assume that the PRILOG and PRISLDS records reflect the following data set scenario:

- The first data set runs from 1:00 PM to 2:00 PM.
- The second data set runs from 2:00 PM to 3:00 PM.
- The third data set runs from 3:00 PM to 4:00 PM.

Assume also that the subsystem that created the records was running from 1:00 PM to 4:00 PM.

The commands shown in Figure 18 must be issued in order to rebuild the PRILOG and PRISLDS records.

```
NOTIFY.PRILOG RLDS STARTTIME(880261300000) DSN(RLDS1) VOLSER(VO0001)
NOTIFY.PRILOG RLDS STARTTIME(880261300000) RUNTIME(880261400000)
NOTIFY.PRILOG RLDS STARTTIME(880261300000) DSN(RLDS2) VOLSER(VO0002)
NOTIFY.PRILOG RLDS STARTTIME(880261300000) RUNTIME(880261500000)
NOTIFY.PRILOG RLDS STARTTIME(880261300000) DSN(RLDS3) VOLSER(VO0003)
NOTIFY.PRILOG RLDS STARTTIME(880261300000) RUNTIME(880261600000)

NOTIFY.PRILOG RLDS STARTTIME(880261300000) DSN(SLDS1) VOLSER(VO0004)
NOTIFY.PRILOG RLDS STARTTIME(880261300000) RUNTIME(880261400000)
NOTIFY.PRILOG RLDS STARTTIME(880261300000) DSN(SLDS1) VOLSER(VO0005)
NOTIFY.PRILOG RLDS STARTTIME(880261300000) RUNTIME(880261500000)
NOTIFY.PRILOG RLDS STARTTIME(880261300000) DSN(SLDS1) VOLSER(VO0005)
NOTIFY.PRILOG RLDS STARTTIME(880261300000) RUNTIME(880261600000)
```

Figure 18. Command to Rebuild PRILOG/PRISLDS with Multiple Entries

Note: The RLDS entries must be built prior to the SLDS entries because a PRILOG record must exist for a corresponding PRISLDS record to exist.

3.8 Rearchiving an OLDS

For reasons external to DBRC, it might be necessary to rearchive an OLDS.

DBRC does not offer the capability of setting on the archive needed flag for an OLDS which has already been archived.

It is not possible to rearchive an OLDS under DBRC control. The archive utility fails if run under DBRC control for the following reasons:

- When the archive utility is run with an archived OLDS as input, it fails since the OLDS has already been archived.
- When the archive utility runs with an SLDS as input, it fails when it is unable to find a PRILOG ora PRISLDS record that has a corresponding entry to the input data set.

The rearchiving of an OLDS must be performed outside of the control of the DBRC. The best way to rearchive an OLDS is to copy it using the same data set name and VOLSER. Since this is not an actual rerunning of the ARCHIVE utility, the ability to omit records that are not needed for restart or recovery is lost.

DFSUARCO can be used to perform this task. Run DFUARC0 with PARM = 'DBRC = N'. Enter the OLDS on the input SLDS DD card, as if the OLDS were a batch log created outside DBRC. Figure 19 shows the required JCL.

```
//AR00001 EXEC PGH=DFSUARCO,PARH='DBRC=N',REGION=4096K
//*
//* JCL to Rearchive an OLDS
//*
//STEPLIB  DD DSN=SYSC.STIMS22X.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//DFSOLPxx DD DSN=STIMS220.OLP00,DISP=SHR,DCB=BUFNO=30
//DFSSLOGP DD DSN=STIHS220.SLDS(+l),
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(STIHS220.DCB,LRECL=22524,8LKSIZE=22528,
//            RECFM=VB,BUFNO=30),
//            UNIT=3480,V0L=SER=(TAPE01)
//SYSIN    DD *
SLDS
/*
//
```

Figure 19. Commands for Rearchiving an OLDS

3.9 Change Accumulation Data Set

During data sharing, more logs are generated when batch jobs are running using their own logs. In this case there must be some way of merging these logs. Change accumulation provides this facility.

When block level data sharing takes place, all logs required for recovery must have been processed by the change accumulation utility before a recovery job can be run.

DBRC Actions for Change Accumulation:

DBRC uses information recorded in the RECON data set for generating the JCL for the change accumulation (CA) utility.

When a database is updated for the first time by a subsystem, an ALLOC record is written as part of the database data set (DBDS) information in the RECON. The ALLOC record registers the time of the first update by the subsystem and the log start time of the log being run at the time of updates (the log start time of the PRILOG).

DBRC also records the data set sequence number (DSSN) in the ALLOC record. The DSSN is used to show the relative order in which the database is updated. The DSSN is held in the DBDS record in the RECON data set. If the database is not authorized to any other update, the DSSN is incremented by one and registered on the ALLOC record. If the database is authorized to another update, the same DSSN is used to show that there are concurrent updates.

In such a case, a merge-needed (MN) record is created with the time-stamp of the earliest log data set that updated the database.

Note: This MN record is NOT used by any recovery utility. In IMS Version 2, it continues to exist for compatibility only.

3.10 Change Accumulation Group

The DBRC DBDS concept for grouping the database data sets can also be used for change accumulation processing. In this case, the group of DBDSs is called the Change Accumulation Group. The RECON record which contains the change accumulation group information is a CAGRP.

A given DBDS can belong to only one change accumulation group.

A given CAGRP can contain up to 1,024 database data sets.

DBRC allows the adding and deleting of a member to an existing CAGRP without having to delete and redefine the entire group, and without the loss of recovery information about the group.

3.10.1 Adding a Member to a Change Accumulation Group

When a new member is added to a CAGRP, there is no effect until the next execution of the processing. Figure 20 shows an example of how an added member is treated in CA and recovery processing:

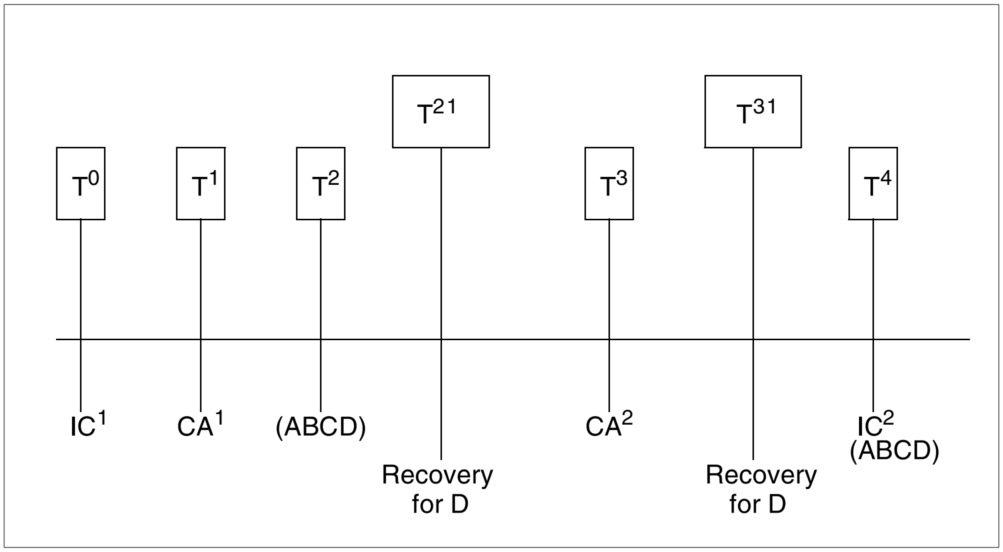


Figure 20. Adding a Member to a CAGRP

1. At time *T0, IC1* is taken for DBDS A, B, C, and D.
2. Some processing is done against those DBDSs, and their logs or DSLOGs are processed at *T1,* producing the change accumulation *CA1*. At time T1, DBDS D is not a member of the CAGRP, so it is not included in the CA1 group processing.
3. At time *T2*, DBDS D is added to the CAGRP.
4. At time *T3*, CA processing is done for the entire group, including the new member D, in order to produce *CA2*. DBRC uses:
   - The output data set from CA1 and the SLDSs, RLDSs, or DSLOGs since CA1 for members A, B, and C.
   - The SLDSs, RLDSs or DSLOGs since IC1 for member D.
5. If recovery for the DBDS D is required at the time *T21* (prior to CA2 processing, but after DBDS D has been added as a member to the CAGRP), then DBRC selects IC1 and the SLDSs and RLDSs since IC1 as input to the recovery utility.
6. If a recovery for the DBDS D is run at the time *T31* (after CA2 processing in which DBDS D has been included), then the DBDS D is treated in the same way as the other members of the CAGRP.

3.10.2 Deleting a Member from a CAGRP

When a member of a CAGRP is deleted, it is treated as if it never existed in the CAGRP. Figure 21 shows how a deleted member is treated in CA and recovery processing:

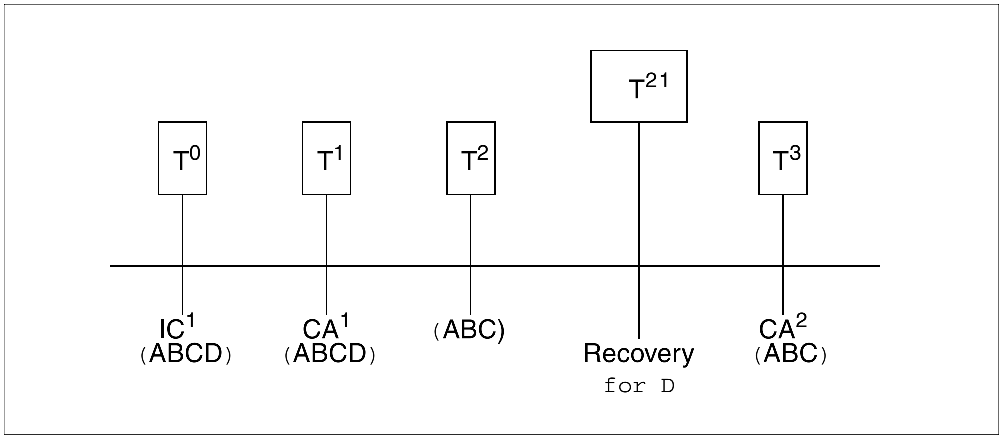


Figure 21. Deleting a Member from a CAGRP

1. At the time *T0, IC1* is taken for DBDS A, B, C, and D.
2. Some processing is done against these DBDSs, and their logs or DSLOGs are processed at the time T1 producing the *CA1* for the CAGRP which includes DBDSs A, B, C, and D.
3. At time *T2*, DBDS D is deleted from the CAGRP.
4. At time *T3*, a CA utility is run. DBRC does not include DBDS D to produce the new *CA2*.
5. If a recovery is run at time *T21,* DBRC has no CA information about DBDS D. DBRC selects IC1 and SLDSs, RLDS, and DSLOGs created after IC1 as input to the database recovery.

This last example provides the following general guideline:

Perform an IMAGE COPY just before adding or deleting a member to a CAGRP for a given database. When this is done, any subsequent recovery is not complicated by the inclusion or exclusion of log data sets.

3.10.3 Change Accumulation Group GENJCL Command

The default for the maximum sequence length field on the ID statement is 10 bytes. Normally this is not as large as the root segment sequence field. The value of this field must be at least as large as the largest root segment key. The correct value can be found by looking at the KEYS parameter in the AMS DEFINE CLUSTER command used to create the VSAM data sets. Figure 22 shows an example of the sequence length set to 12.

```
GENJCL.CA GRPNAME(DBGCA) USERKEYS((%CAKYLG,'12'))
```

Figure 22. DBRC Command for CA with USERKEY

Figure 23 on page 38 shows an example of the skeletal JCL used by the GENJCL.CA command.

```
//CA%STPNO EXEC PGM=DfSUCUMO,PARH='CORE=l00000,DBRC=Y',REGION=4096K
//
//       JCL FOR CHANGE ACCUMULATION.
//
//STEPLIB        DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT       DD SYSOUT=
//IMS   DD DSN=STIHS22X.DBDLIB,DISP=SHR
//SYSOUTDD SYSOUT=
//SORTLIB        DD DSN=SYSl.SORTLIB,DISP=SHR
//SORTWKOl DD UNIT=SYSDA,SPACE=(CYL,(2) "CONTIG)
//SORTWKO2 DD UNIT=SYSDA,SPACE=(CYL,(21 "CONTIG)
//SORTHKO3 DD UNIT=SYSDA,SPACE=(CYL,(2) "CONTIG)
                //SORTHKO4 DD UNIT=SYSDA,SPACE=(CYL,(2) "CONTIG)
//SORTWKO5 DD UNIT=SYSDA,SPACE=(CYL,(2) "CONTIG)
//SORTWKO6 DD UNIT=SYSDA,SPACE=(CYL,(2) "CONTIG7
%DELETE (%CAODSN EQ " )
//DFSUCUMO DD DSN=%CAODSN,DISP=SHR
%ENDDEL
%DELETE (%CAODSN NE " )
//DFSUCUMO DD DUMMY,DCB=BLKSIZE=100
%ENDDEL
//DFSUCUMN DD DSN=STIMS220.DBG.CALOG(+1),
//        DISP=(NEW,CATLG,DELETE),
//        UNIT=3380,V0L=SER=(STIMS57,SPACE=(TRK,(l0,ll,RLSE),
//        DCB=(STIMS22X.DCB,RECFH=VB,LRECL=23468,BLKSIZE=234727
%SELECT RLDS((%CAGRPl,(FROM(%DSLLGTM7))
//DFSULOG       DD DSN=%LOGDSN,DISP=SHR
%ENDSEL
%DELETE (%LOGSEL EQ 'YES')
//DFSULOG       DD DUMMY,DCB=BLKSIZE=100
%ENDDEL
//DFSUDD1       DD DUMMY
//SYSIN DD
ID                  %CAKYLG
%CADBOI
```

Figure 23. Skeletal JCL for Change Accumulation

Figure 24 on page 39 shows the generated CA JCL from Figure 22 on page 37 and Figure 23 on page 38.

```
//CA00001 EXEC PGM=DFSUCUMO,PARM='CORE=100000,D8RC=Y',REGION=4096K
//
//       JCL FOR CHANGE ACCUMULATION.
//
//STEPLIB DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SVSPRINT DD SYSOUT=
//IMS   DD DSN=STIMS22X.DBDLIB,DISP=SHR
//SYSOUTDD SYSOUT=
//SORTLIB    DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTWKO1 DD UNIT=SYSDA,SPACE=(CYL,(2) "CONTIG)
//SORTWKO2 DD UNIT=SYSDA,SPACE=(CYL,(2l "CONTIGl
//SORTWKO3 DD UNIT=SYSDA,SPACE=(CYL,(2) "CONTIG7
//DFSUCUMO DD DUMMY,DCB=BlKSIZE=100
//DFSUCUMN DD DSN=STIMS220.DBG.CALOG(+1),
//        DISP=(NEW,CATLG,DELETE),
//        UNIT=3380,V0L=SER=(STIMS5),SPACE=(TRK,(10,1),RLSEl,
//        DCB=(STIHS22O.DCB,RECFM=VB,LRECL=23468,8LKSIZE=23472l
//DFSULOG    DD DSN=STIMS220.DBG.BOlLOG.GOOOlVOO,DISP=SHR
//      DD DSN=STIMS220.DBG.BOlLOG.GOOO2VOO,DISP=SHR
//DFSUDD1    DD DUMMY
//SYSIN DD
ID                  12
DBODBGAMAP 883131845DBGAMAP
DBODBGAMAX 883131845DBGAMAX
DBODBGAMBP 883131845DBGAMBP
DBODBGAMBX 883131846DBGAMBX
DBODBGAMBY 883131846DBGAMBY
```

Figure 24. Generated Change Accumulation JCL

3.11 DFHSM Considerations

From the DBRC and DFHSM points of view, attention must be given to:

- Log, change accumulation, and image copy data sets
- Databases

3.11.1 Log, Change Accumulation, and Image Copy Data Sets

Some customers use DFHSM to manage the following:

- Image copy (IC)
- Change accumulation (CA)
- Log data sets

When these recovery data sets are migrated back online by DFHSM, they retain their original data set names as recorded in the RECON, but frequently are placed on a VOLUME different from the VOLUME recorded in the RECON at the time of their creation.

Because DBRC ensures that the correct data sets are being used, the recovery JCL generated by the GENJCL process using the "old" data registered in RECON fails.

Figure 25 shows the messages produced by DBRC when it detects a discrepancy between the information registered in RECON and the supplied JCL.

```
DFS391I    D A T A   B A S E   D A T A   S E T   R E C O V E R Y      U T I L I T Y
DFS391I   S DBGAMAP  DBGAMAP DFSUDUMP       DBGAMAP    RESTORE CONTROL
DFS391I   ** RECOVER DATA BASE DBGAMAP DDNAME DBGAMAP
DSPO351I  DFSUDUMP DD INFORMATION IS INCONSISTENT WITH RECON
DSPO351I  DSNNAME=STIMS220.DBG.DBGAMAP.IC1            **  ERROR **
DSPO351I    FILE SEQ=0001 VOL SEQ=0001 NUMBER VOLS=0001
DSPO351I    VOLS=STIMS5
DFS339I   FUNCTION RV HAS COMPLETED ABNORMALLY RC=16
```

Figure 25. DBRC Information Messages

To solve this problem, a new option has been added to the CHANGE.RECON and INIT.RECON commands that allow the user to specify that all IC, CA, and Log data sets are catalog-managed.

Setting this option (*CATDS*) causes DBRC to bypass VOLSER, SEQUENCE checking, and FILE SEQUENCE verification of cataloged recovery data sets. The option CATDS is recorded in the RECON header record and becomes effective immediately. The option can be reversed by the use of another option provided (*NOCATDS*) which is also the DEFAULT.

A user choosing to notify DBRC of catalog-managed recovery data sets must first ensure that the following conditions are met:

- VOLUME and UNIT type information is removed from skeletal JCL members for utility job streams which use these data sets.

  All IC, CA, and Log data sets are cataloged, since VOLUME/UNIT information is not present in the generated JCL.

  If the user provides one or more DD cards with VOLUME/UNIT information present, DBRC performs the normal verification of that data set, even if CATDS has been specified as an option.

- Only SINGLE volume log data sets are used.

  Users whose log data sets are being managed by DFHSM must already have this restriction in force, since DFHSM currently does not handle multi-volume data sets.

If the previous conditions have been met, the user can enter the following command:

```
CHANGE.RECON CATDS
```

To revert to the normal verification process, the user should issue the command:

```
CHANGE.RECON NOCATDS
```

The same options are available on the INIT.RECON command with the same meaning.

To archive these changes, a PTF must be applied:

- For IMS 2.1, the PTF number is UL40257.
- For IMS 2.2, the PTF number is UL40258.

Note: The changes added to DBRC for IMS Version 2 are not available for IMS 1.3.

IMS 1.3 installation may be able to avoid this problem by making sure that DFHSM recalls the data set to the same volume from where it was migrated. This can be done by ensuring that the unit and volume parameters are used on the JCL for all IMS utilities. A problem can then occur only when there is not enough room on that volume to recall the data sets.

3.11.2 Database Data Sets

DBRC does not hold unit and volume serial information about any DBDSs; therefore, DFHSM can migrate and recall databases without affecting DBRC.

However, it should be recognized that when a database is allocated with dynamic allocation, the DLISAS region waits while DFHSM recalls the database. The response times could be severely affected.

In most cases this is acceptable in a test environment, but not in a production one. It is also true that for production databases, opened and used every day, the probability of a database being automatically migrated or recalled is very low.

Chapter 4. RECON Record Types

This chapter lists the record types that can be found in the RECON data sets, and for each record, explains its purpose and its relationship with other record types.

The relationship is never imbedded in the records like a direct pointer, but can be built by DBRC using the information registered in each record type. This allows constant acccss of the related records through their physical keys.

4.1 RECON Records

There are six general classes of RECON record types:

1. Control records
2. Log records
3. Change accumulation records
4. Database data set group records
5. Subsystem records
6. Database records

4.1.1 Control Records

Control records are used for controlling the RECON data set and the default values used by DBRC. This class of records includes:

- RECON Header record
- RECON header extension record

4.1.2 Log Records

Log records are used for tracking the log data sets used by all subsystems. This class of records includes:

- PRILOG and SECLOG records (including interim log records)
- LOGALL record
- PRIOLD and SECOLD records (inclucling interim OLDS records)
- PRISLDS and SECSLDS records (including interim SLDS records)

4.1.3 Change Accumulation Records

Change accumulation records are used for tracking information about change accumulation groups. This class of records includes:

- Change accumulation group records
- Change accumulation execution records
- Available change accumulation data set records

4.1.4 DBDS Group Records

Database Data Set Group (DBDSGRP) records are used to define the members of a DBDS group. The only record type in this class is a DBDS group record.

4.1.5 Subsystem Records

Subsystem records are used to track the state of IMS in CICS subsystems. The only record type in this class is a SUBSYS record.

4.1.6 Database Records

Database records are used to track the state of:

- Databases
- DBDSs
- Resources required for recovery of DBDSs

This class of records includes:

- Database record
- AREA authorization record
- DBDS record
- Merge-needed record
- Allocation record
- Image copy record
- Reorganization record
- Recovery record

4.2 RECON Header Record

The header is the first record created in the RECON data set by the INIT.RECON command as shown in Figure 26.

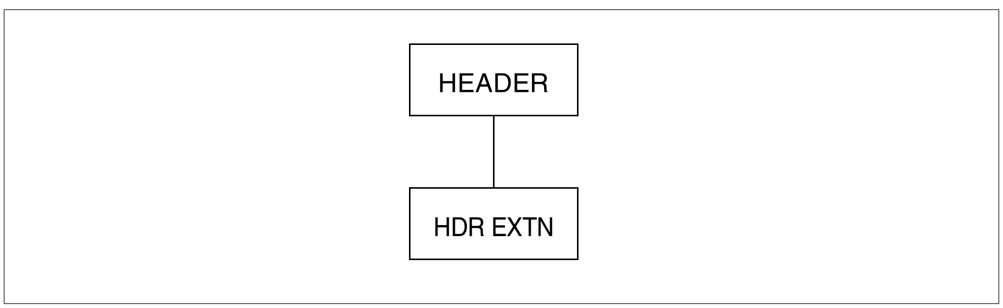


Figure 26. The RECON Header

The header identifies the data set as a RECON data set and keeps information related to the whole DBRC system. It also controls the concurrent update of the RECON data set by several subsystems. The information kept in this record is read when the RECON is opened and the values are placed in various control blocks. Hence, the default values are accessible to other DBRC routines without additional I/O operations to the RECON data set.

The RECON header record is related to the RECON header extension record.

4.3 RECON Header Extension Record

The RECON header extension record identifies the individual RECON data sets. It is also used in the synchronization process of the two primary RECON data sets. It is created by the INIT.RECON command, together with the RECON header record.

The RECON header extension record is related to the RECON header record, as shown in Figure 27.

```
RECON
RECOVERY CONTROL DATA SET, RELEASE 4
DMB#=25          HIGHEST DBDS SEQ=25      INIT TIMESTAMP=88.293 16:46:53.6
NOFORCER         LOG DSN CHECK=CHECK17    STARTNEW=NO      CONTROL=SHARE
TAPE UNIT=3400             DASD UNIT=3400     TRACEOFF    SSID=I220


-DDNAME-      -STATUS-              -DATA SET NAHE-
RECON1         COPY1                 STIHS220.RECON1
RECON2         COPY2                 STIHS220.RECON2
RECON3         SPARE                 STIMS220.RECON3
```

Figure 27. RECON Information

4.4 DB Record

The Database (DB) record describes a database. See Figure 28.

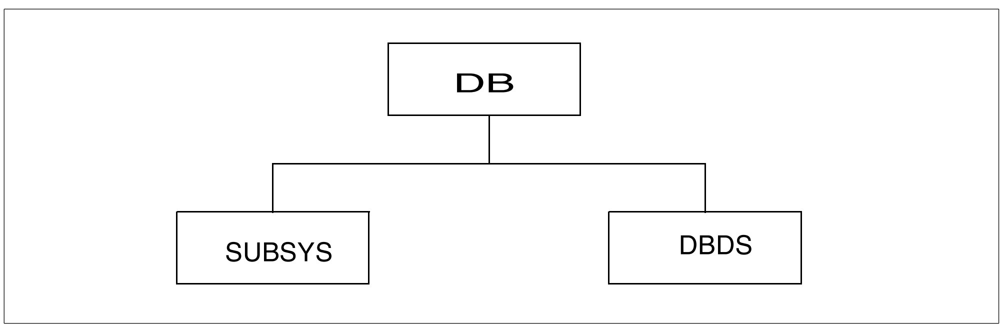


Figure 28. The DB Record

There is one DB record in the RECON data set for each database that has been registered to DBRC through the use of the INIT.DB command.

A DB record is deleted when the DELETE.DB command is used. After use of DELETE.DB, all DBDS records related to the particular DB record are also deleted.

A DB record includes:

- Name of the DBDS for the database
- Share level specified for the database
- Database status flags
- Current authorization usage

A DB record is symbolically related to:

- The DBDS record for each database data set
- The SUBSYS record for each subsystem currently authorized to use the database.

See Figure 29 below.

```
DB                              IRLMID=NULL         DMB#=21   TYPE=IMS
 DBD=DBGAMAP
 SHARE LEVEL=2
 FLAGS:                               COUNTERS:
   BACKOUT NEEDED        =OFF           RECOVERY NEEDED COUNT   =0
   READ ONLY             =OFF           IMAGE COPY NEEDED COUNT =0
   PROHIBIT AUTHORIZATION=OFF           AUTHORIZED SUBSYSTEMS   =0
                                        HELD AUTHORIZATION STATE=0
                                        EQE COUNT               =0
```

Figure 29. DB Information

4.5 DBDS Record

The Database Data Set (DBDS) record describes a database data set in Figure 30. There is a DBDS record in the RECON data set for each database data set that has been defined to the DBRC using the INIT.DBDS command.

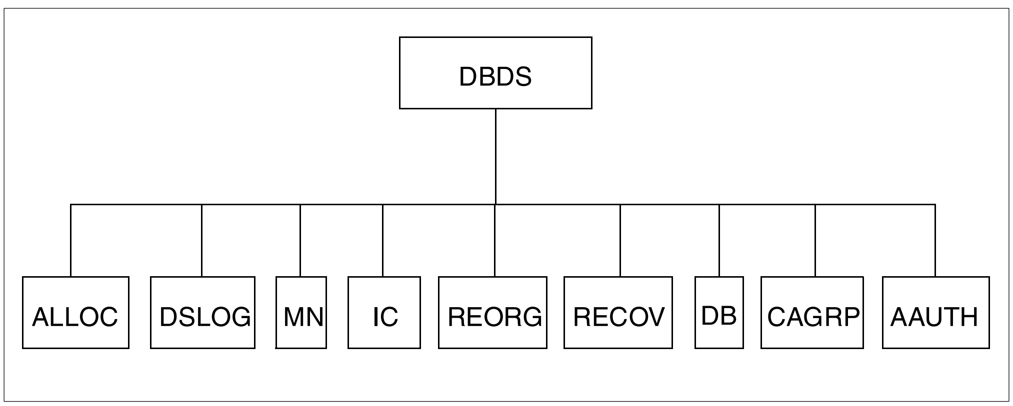


Figure 30. The DBDS Record

A DBDS record is deleted from RECON with the DELETE.DBDS or the DELETE.DB command.

The DBDS record includes:

- Data set name
- DD name for the data set
- DBD name of the database
- Data set, data base organization
- Status flags for the data set
- Information related to image copy or change accumulation
- Name of the JCL member to be used for GENJCL.IC or GENJCL.RECOV.

A DBDS record has the following relationship to other records:

- DB record for the database to which the data set belongs
- CAGRP record for the change accumulation group to which the database data set belongs (when a change accumulation group has been defined)
- DSLOG, ALLOC, MN, IC, REORG, RECOV, AAUTH records.

See Figure 31 below.

```
DBDS
 DSN=STIMS220.DBG.DBGAMAP                          DBDS SEQ=21   TYPE=IMS
 DBD=DBGAMAP   DDN=DBGAMAP  DSID=001 DBORG=HIDAM  DSORG=VSAM DSLG SEQ=00
 CAGRP=DBGCA     GENMAX=3     IC AVAIL=1     IC USED=2     DSSN=00000006
 REUSE           RECOVPD=1
 DEFLTJCL=**NULL**  ICJCL=DBGIC     OICJCL=DBGOIC    RECOVJCL=DBGRECOV
 FLAGS:                             COUNTERS:
   IC NEEDED      =OFF
   RECOV NEEDED   =OFF
   RECOV EXECUTION=OFF              EQE COUNT                 =0
```

Figure 31. DBDS Information

4.6 SUBSYS Record

The Subsystem (SUBSYS) record informs DBRC that a subsystem is currently active, as shown in Figure 32.

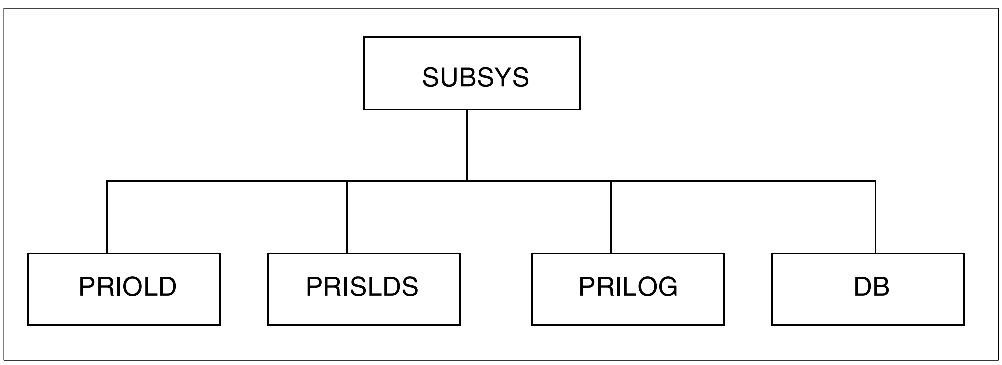


Figure 32. The SUBSYS Record

A SUBSYS record is created any time a subsystem signs on to DBRCA.

A SUBSYS record is deleted when:

- The subsystem terminates normally
- The subsystem terminates abnormally, but without any database updates
- DBRC is notified of the successful completion of the subsystem recovery process (IMS or CICS emergency restart or batch backout).

The SUBSYS record includes:

- ID of the subsystem
- Start time of the log
- Subsystem status flags

- DBDS name for each database currently authorized to the subsystem.

A symbolic relationship exists with the following record types:

- PRILOG record for the log that the subsystem is creating (or PRIOLD or PRISLDS depending on the type of subsystem)
- DB record for each database currently authorized to the subsystem.

See Figure 33 below.

```
SSYS
 SSID=I220      IRLMID=**NULL**      LOG START=88.327   20:27:03.2
 SSTYPE=ONLINE  ABNORMAL  TERM=OFF   RECOVERY STARTED=NO   BACKUP=NO
 IRLM STATUS=NORMAL


 AUTHORIZED DATA BASES/AREAS=5          LEVEL=4
                                                         ENCODED
   -DBD-        -AREA-   -LEVEL-     -ACCESS INTENT-     -STATE-
   DBGAMBX      **NULL**    2           UPDATE              6
   DBGAMBP      **NULL**    2           UPDATE              6
   DBGAMAX      **NULL**    2           UPDATE              6
   DBGAMAP      **NULL**    2           UPDATE              6
   DBGAMBY      **NULL**    2           UPDATE              6
```

Figure 33. SUBSYS Information

4.7 DBDSGRP Record

The Database Data Set Group (DBDSGRP) record describes a DBDS group, as shown in Figure 34.

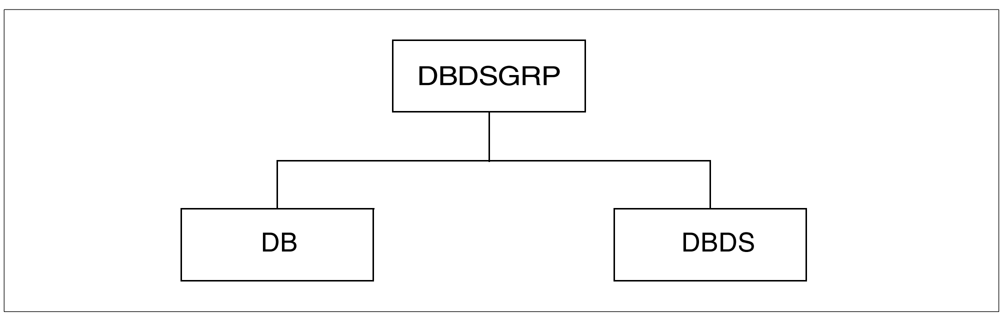


Figure 34. The DBDSGRP Record

The DBDSGRP is created with the use of the INIT.DBDSGRP command.

It is deleted with the use of the DELETE.DBDSGRP command.

The DBDSGRP record includes:

- DBDSGRP name
- Name of all the DBDSs for the databases defined in the group
- DD name for all the DBDS of the databases defined in the group.

The DBDSGRP record is symbolically related to each:

- DB record defined in the group
- DBDS record defined in the group.

See Figure 35 below.

```
DBDSGRP
 GRPNAME=DBGGRP1           #MEMBERS=5      -DBD-      -DDN/AREA-
                                           DBGAMAP    DBGAMAP
                                           DBGAMAX    DBGAMAX
                                           DBGAMBP    DBGAMBP
                                           DBGAMBX    DBGAMBX
                                           DBGAMBY    DBGAMBY
```

Figure 35. DBDSGRP Information

4.8 CAGRP Record

The Change Accumulation Group (CAGRP) record describes a change accumulation group as shown in Figure 36. Each CAGRP record lists up to 1024 DBDSs whose change records are accumulated together during an execution of the CAGRP utility.

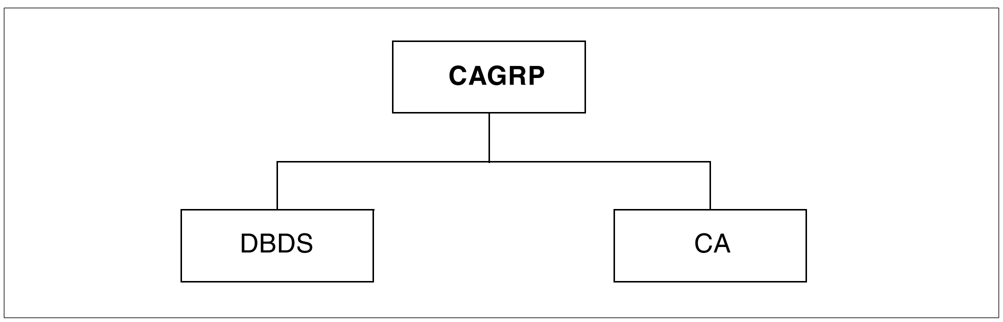


Figure 36. The CAGRP Record

The CAGRP record is created when the INIT.CAGRP command is used to define the CAGRP to DBRC.

The CAGRP record is deleted when the DELETE.CAGRP command is used. This command also deletes all the CA records related to this particular CAGRP record.

The CAGRP record includes:

- CAGRP name
- DBDS name and the DD name for each DBDS belonging to the CAGRP
- Information related to the CA records
- Skeletal JCL name to be used when the GENJCL.CA command is used

The CAGRP record in Figure 37 below is symbolically related to:

- DBDS record for each member of the CAGRP
- CA records each describing a change accumulation output data set (either used or only pre-allocated).

```
CAGRP
 GRPNAME=DBGCA    GRPMAX=3       CA AVAIL=0    CA USED=1
 NOREUSE CAJCL=DBGCA             DEFLTJCL=**NULL**
             #MEMBERS=5                        -DBD-      -DDN-
                                               DBGAMAP   DBGAMAP
                                               DBGAMAX   DBGAMAX
                                               DBGAMBP   DBGAMBP
                                               DBGAMBX   DBGAMBX
                                               DBGAMBY   DBGAMBY
```

Figure 37. CAGRP Information

4.9 CA Record

The Change Accumulation (CA) record in Figure 38, describes a change accumulation data set. There is a CA record for each used change accumulation output data set and one for each change accumulation output data set predefined to DBRC but not yet used.

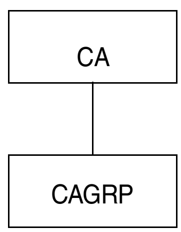


Figure 38. The CA Record

The INIT.CA command is used to predefine a change accumulation data set to DBRC when the REUSE parameter has been chosen on the INIT.CAGRP command.

The DELETE.CA command (dealing with a single CA record) or the DELETE.CAGRP command (dealing with all the CA records related to a given CAGRP record) can delete the CA records.

The CA record includes:

- Name of the CAGRP to which it belongs
- Data set name of the change accumulation output data set and the volume serial information
- List of purge tirnes for each of the logs input to the change accumulation.

The CA record in Figure 39 below, is symbolically related to the CAGRP record to which it belongs.

```
CA
 DSN=STIMS22O.DBG.CALOG.G0001V00                  FILE SEQ=1
 CAGRP=DBGCA     UNIT=3400
 STOP    = 88.327   11:22:48.9      VOLS DEF=1    VOLS USED=1
                                    VOLSER=STIMS5
 RUN     = 88.327   11:24:17.6
    -DBD-    -DDN-      -PURGETIME-     -CHG-CMP-   -LSN-      -DSSN-
   DBGAMAP   DBGAMAP  88.327  11:11:50.5 YES YES 000000000000 00000003
   DBGAMAX   DBGAMAX  88.327  11:12:06.6 NO  YES 000000000000 00000000
   DBGAMBP   DBGAMBP  88.327  11:12:26.2 YES YES 000000000000 00000003
   DBGAMBX   DBGAMBX  88.327  11:12:42.9 YES YES 000000000000 00000003
   DBGAMBY   DBGAMBY  88.327  11:12:58.8 YES YES 000000000000 00000003
```

Figure 39. CA Information

4.10 PRILOG/SECLOG Record

The Primary Recovery Log (PRILOG) record or the Secondary Recovery Log (SECLOG) record in Figure 40, describes a log RLDS created by an IMS DC or CICS/OS/VS online system, a batch DL/I job, or the archive utility.

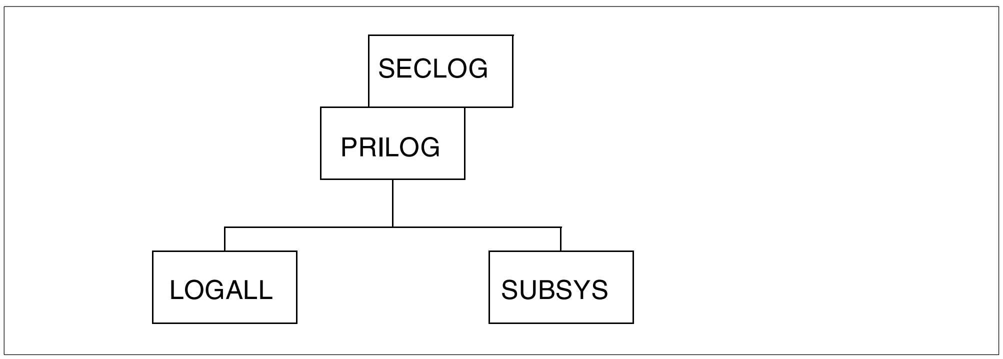


Figure 40. The PRILOG/SECLOG Record

A PRILOG record is created, together with a LOGALL record, whenever a log is opened. If the subsystem is an IMS batch job and dual log is in use, a SECLOG record is also created.

A PRILOG record is deleted in the following cases:

- The command DELETE.LOG INACTIVE deletes all the log records no longer needed for recovery purposes.
- The command DELETE.LOG TOTIME deletes all the inactive log records older than the specified time.
- The command DELETE.LOG STARTIME deletes a particular log record.

The PRILOG (SECLOG) record in Figure 41 below is symbolically related to:

- The LOGALL record for the same log
- The SUBSYS record for the subsystem creating the log (primary or dual) when the subsystem is active

```
PRILOG
 START   = 88.327   11:14:43.9*  STOP    = 88.327    11:19:55.3
 SSID=R82123R3     #DSN=1        IMS


 DSN=STIMS220.DBG.B01LOG.G0003V00                          UNIT=3400
 START   = 88.327   11:14:43.9   STOP    = 88.327     11:19:55.3
 FILE SEQ=0001    #VOLUMES=0001  -VOLSER-         -STOPTIME-
                                  STIMS5    88.327   11:19:55.3
```

Figure 41. PRILOG Information

4.11 PRISLDS/SECSLDS Record

The Primary System Log (*PRISLDS*) record or the Secondary System Log (*SECSLDS*) record in Figure 42 describes a system log SLDS created by an I*MS DC online system*.

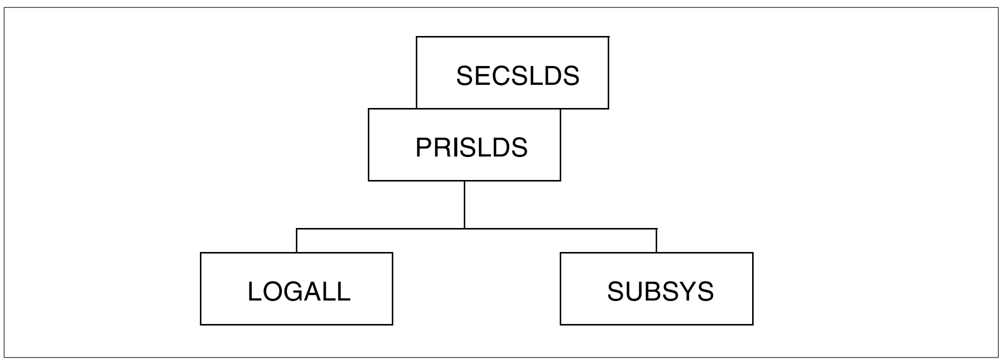


Figure 42. The PRISLDS/SECSLDS Record

A PRISLDS record is created, along with a LOGALL record, whenever a system log is opened. A SECSLDS record can be created at archive time.

A PRISLDS record is deleted in the following cases:

- The command DELETE.LOG INACTIVE deletes all the log records no longer needed for recovery purposes.
- The command DELETE.LOG TOTIME deletes all the inactive log records older than the specified time.
- The command DELETE.LOG STARTIME deletes a particular log record.

The PRISLDS (SECSLDS) record in Figure 43 below is symbolically related to:

- The LOGALL record for the same log
- The SUBSYS record for the subsystem creating the log (primary or dual) when the subsystem is active

```
PRISLD
 START   = 88.327    12:41:15.3*  STOP    = 88.327   13:03:26.1
 SSID=I220         #DSN=1

 DSN=STIMS220.SLDS.G0003V00                                UNIT=3400
 START   = 88.327   12:41:15.3    STOP    = 88.327   13:03:26.1
 FILE SEQ=0001    #VOLUMES=0001 - VOLSER-         -STOPTIME-
                                  STIMS4     88.327   13:03:26.1
```

Figure 43. PRISLDS Information

4.12 PRIOLD/SECOLD Record

The Primary OLDS (PRIOLD) record and the Secondary OLDS (SECOLD) record in Figure 44 describe the IMS DC Online Data Sets (OLDS) defined for use. Whenever an OLDS is defined to IMS DC, the PRIOLD record is updated. If IMS dual logging is in use, the SECOLD record is also updated. The PRIOLD (SECOLD) record is deleted by the DELETE.LOG command.

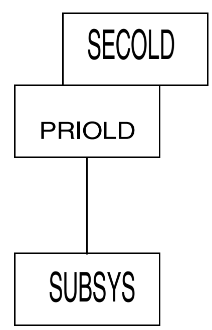


Figure 44. The PRIOLD/SECOLD Record

The PRIOLD (SECOLD) record in Figure 45 on page 54 is symbolically related to the SUBSYS record for the subsystem using the OLDS (primary or dual).

```
PRIOLD
 SSID=I220             # DD ENTRIES=3

 DDNAME=DFSOLP02    DSN=STIMS220.OLP02
 START   = 88.319    18:46:13.6   STOP     = 88.319   19:01:57.0
 STATUS=ARC COMPLT                                  FEOV=NO AVAIL
 PRILOG TIME=88.319    18:39:58.7       ARCHIVE JOB NAME=ST220ARC


 DDNAME=DFSOLP01    DSN=STIMS220.OLP01
 START   = 88.321    17:24:41.8   STOP     = 88.321   17:27:33.3
 STATUS=ARC COMPLT                                  FEOV=NO AVAIL
 PRILOG TIME=88.321    17:15:53.9       ARCHIVE JOB NAME=ST220ARC

 DDNAME=DFSOLP00    DSN=STIMS220.OLP00
 START  = 88.327     12:41:15.3   STOP     = 88.327   13:03:26.1
 STATUS=ARC COMPLT                                  FEOV=NO AVAIL
 PRILOG TIME=88.327    12:41:15.3       ARCHIVE JOB NAME=ST220ARC

SECOLD
 SSID=I220             # DD ENTRIES=3
 DDNAME=DFSOLS02    DSN=STIMS220.OLS02
 START  = 88.319     18:46:13.6   STOP     = 88.319   19:01:57.0
                                                              AVAIL

 PRILOG TIME=88.319    18:39:58.7
  .
 DDNAME=DFSOLS01    DSN=STIMS22O.0LSO1
  START = 88.321 17: 24:41.8      STOP     = 88.321   17:27:33.3
                                                              AVAIL
  PRILOG TIME=88.321   17:15:53.9

  DDNAME=DFSOLS00   DSN=STIMS220.OLS00
  START = 88.327     12:41:15.3   STOP     = 88.327   13:03:26.1
                                                              AVAIL
  PRILOG TIME=88.327   12:41:15.3
```

Figure 45. PRIOLD/SECOLD Information

4.13 LOGALL Record

The Log Allocation (LOGALL) record in Figure 46 lists the DBDSs for which data base change records have been written to a particular log.

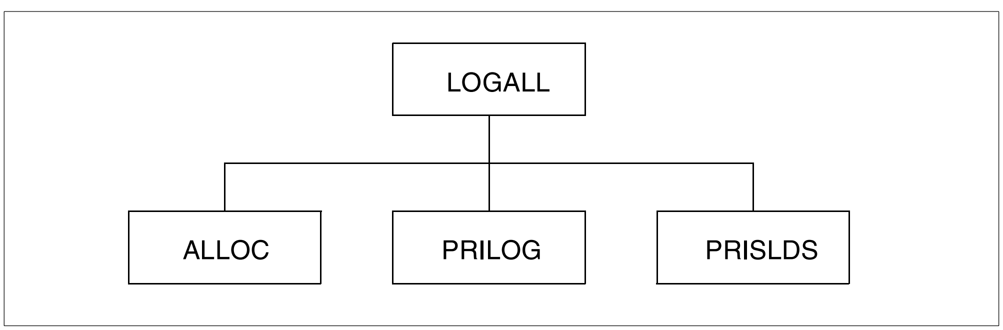


Figure 46. The LOGALL Record

A LOGALL record is created whenever a PRILOG record is created.

A LOGALL record is deleted from RECON whenever its corresponding PRILOG record is deleted.

The LOGALL record contains a list of the names of DBDSs that have change records on the log. There is a one-to-one correspondence between entries in this list and ALLOC records. Entries are added in this list when ALLOC records are created and deleted when ALLOC records are erased. When there are no more ALLOC records, this list is empty and the log is no longer needed for future recovery.

The LOGALL record in Figure 47 below is symbolically related to:

- The ALLOC record for each of the entry in the LOGALL record
- The PRILOG record for the same recovery log
- The PRISLDS record for the same system log

```
LOGALL

 START   = 88.327       11:14:43.9*
 DBDS ALLOC=4                                   -DBD-    -DDN-      -ALLOC-
                                                 DBGAMBP  DBGAMBP    1
                                                 DBGAMBX  DBGAMBX    1
                                                 DBGAMAP  DBGAMAP    1
                                                 DBGAMBY  DBGAMBY    1
```

Figure 47. LOGALL Information

4.14 ALLOC Record

The Allocation (ALLOC) record in Figure 48 shows that a DBDS has been changed, and that database change records have been written to a particular log.

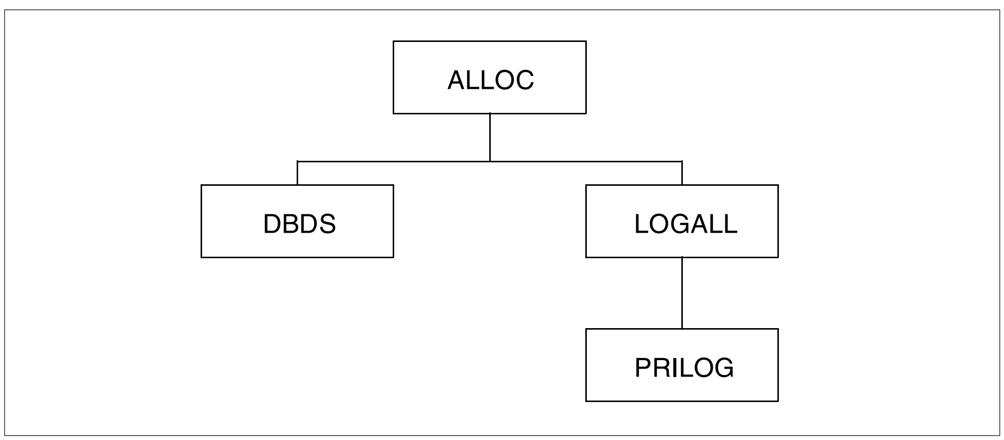


Figure 48. The ALLOC Record

An ALLOC record is created for a DBDS when a subsystem, signed on to DBRC, updates that DBDS for the first time. The ALLOC record, if still active when the need for recovery arises, shows that the related log must be included in the recovery process.

The ALLOC record is deleted when its DEALLOC time-stamp becomes older than the oldest image copy registered to DBRC for the DBDS.

The ALLOC record in Figure 49 on page 56 is symbolically related to:

- The DBDS record for the DBDS to which the ALLOC record belongs

- The LOGALL record for the log that the ALLOC record identifies
- The PRILOG record through the LOGALL record

```
ALLOC
 ALLOC   = 88.327          11:20:48.5     START = 88.327    11:20:40.4
                                          DSSN=00000003


ALLOC
 ALLOC   = 88.327          12:56:53.0     START = 88.327    12:41:15.3
 DEALLOC = 88.327          12:59:10.3     DSSN=00000004
```

Figure 49. ALLOC Information

4.15 MN Record

The merge-needed (MN) record, in past IMS releases, showed that a merge of data base change records recorded in two or more logs for the same DBDS was needed. In IMS Version 2 it still exists, but only for compatibility. See Figure 50 below.

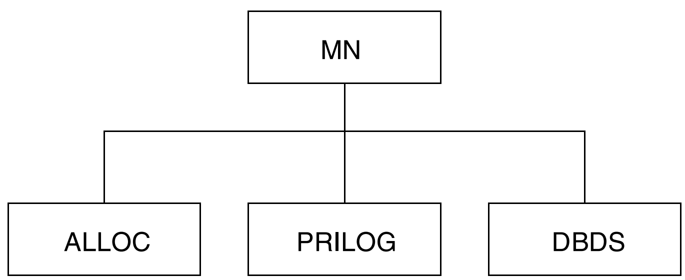


Figure 50. The MN Record

An MN record is created any time an ALLOC record is created for a DBDS, when another ALLOC record has already been created by some other subsystem for the same DBDS.

An MN record is deleted when all the ALLOC records associated with the MN record have been deleted.

The MN record in Figure 51 is symbolically related to:

- The ALLOC record for each of the subsystems listed in the MN records
- The PRILOG record for each of the logs used by the subsystems listed in the MN records
- The DBDS record for the DBDS to which the MN record belongs

```
MRGND
 DBGAMAP       DDN=DBGAMAP            MN TIME = 88.327     13:00:21.6
 DSID=001      DSSN=00000004          SUBSYSTEM COUNT=2

 ASSOCIATED SUBSYSTEM INFORMATION:
  -SSID-       -PRILOG START TIME-       -ALLOCATION TIME STAMP-
 I220          88.327   12:41:15.3         88.327   12:56:53.0
 R82123R5      88.327   13:00:13.5         88.327   13:00:21.6
```

Figure 51. MN Infromation

Note: The MN record is *not used* by any related recovery utility in IMS Version 2.

4.16 IC Record

The Image Copy (IC) record in Figure 52 describes an image copy output data set.

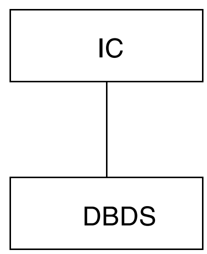


Figure 52. The IC Record

This record can be created:

- Automatically, when the image copy utility is executed to create a standard image copy
- With the NOTIFY.IC command, when a standard image copy has been created with DBRC = NO
- With the NOTIFY.UIC command, when another nonstandard image copy has been created
- In advance, and reserved for future use with the INIT.IC command, when the related DBDS record has the REUSE option
- By the HISAM reload utility, which creates an IC record pointing to the unload data set if the REUSE option is not being used for the DBDS under reload.

This record is deleted when the maximum image copy generation count is exceeded and its time-stamp is beyond the recovery period.

An option available with the image copy utility allows the user to create two copies of the same IC, referred to as image copy-1 and image copy-2. Both copies are described by the IC record.

The IC record is symbolically related to the DBDS record for the DBDS to which it belongs. See Figure 53 below.

```
*IMAGE
 RUN   = 88.327     11:11:50.5     RECORD COUNT =155
 STOP  = 00.000     00:00:00.0     BATCH

IC1
 DSN=STIMS220.DBG.DBGAMAP.IC1                     FILE SEQ=0001
 UNIT=3400                         VOLS DEF=0001 VOLS USED=0001
                                   VOLSER=STIHS5
```

Figure 53. IC Information

4.17 REORG Record

The Reorganization (REORG) record in Figure 54 informs DBRC that a reorganization of a particular DBDS has taken place.

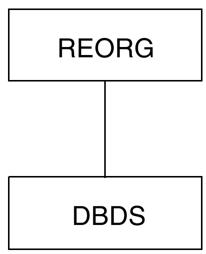


Figure 54. The Reorg Record

With this information, DBRC will not allow recovery operations beyond the time-stamp of this reorganization.

The REORG record is created when:

- A HISAM or HDAM reload utility is successfully executed
- A prefix update utility is executed

The REORG record is deleted when its creation time-stamp is older than the last IC associated with the database data set.

The REORG record in Figure 55 is symbolically related to the DBDS record for the database data set to which it belongs.

```
REORG
RUN  = 88.32711:51:35.1*
```

Figure 55. REORG Information

4.18 RECOV Record

The Recovery (RECOV) record in Figure 56 informs DBRC that the recovery of a particular DBDS has taken place. With this information, DBRC knows when a time-stamp recovery has been performed.

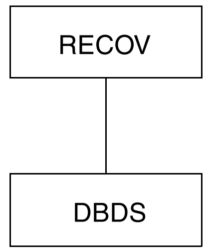


Figure 56. The RECOV Record

The RECOV record is created when the IMS DB recovery utility is successfully executed.

A RECOV record is erased when its creation time-stamp is found to be older than the oldest IC record associated with the DBDS.

The RECOV record in Figure 57 below is symbolically related to the DBDS record for the database data set to which it belongs.

```
RECOV
 RUN     = 88.328       12:24:15.3*
```

Figure 57. RECOV Information

4.19 AAUTH Record

The Area Authorization (AAUTH) record in Figure 58 indicates the sharing status of a Fast Path Database Area.

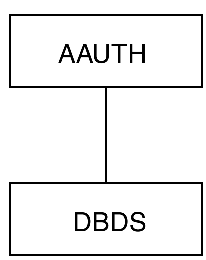


Figure 58. The AAUTH Record

It is symbolically related to the DBDS record for the DBDS to which it belongs.

4.20 Interim Log Records

During the DUP phase of the IMS log recovery utility DFSULTRO, interim log records are created for the RECON data set. All these records are temporary records and are deleted when:

- The REP phase of the utility successfully completes
- The DUP phase resolves all errors

These interim log records are as follows:

- IPRILOG Interim primary recovery log record
- ISECLOG Interim secondary recovery log record
- IPSLDS Interim primary system log record
- ISSLDS Interim secondary system log record
- IPRIOLD Interim primary OLDS record
- ISECOLD Interim secondary OLDS record

Chapter 5. Normal Recovery Situations

This chapter explains the following normal recovery situations:

- Recovery (full or partial) of a database
- Batch backout
- Recovery with a merge needed situation

5.1 Full Recovery of a Database Data Set

A full recovery of a DBDS or DEDB area data set requires the most recent image copy data set. It performs the following tasks:

- The image copy data set is restored to DASD
- Changes made to the DBDS or DEDB area data set since the time-stamp of the image copy data set are applied to the restored DBDS or DEDB area data set. These changes can be contained in:
  - Change accumulation data sets
  - Log data sets (RLDS or SLDS)

5.1.1 Full Recovery—Input

The input to the data base recovery utility consists of the following items:

1. The *control statement* identifying the DBDS for which a full recovery is requested.
2. The *IMS DBD library* providing information about the data set organization of the DBDS to be recovered.
3. The *RECON data set* verifying the correctness of the JCL input for the recovery. The RECON data set is the source for the data set name and volume information for the DSLOG data set that is dynamically allocated durning this recovery.
4. The *image copy data set* used as the base for the recovery. The database recovery utility applies change records from other sources to a copy of this image copy data set to produce the recovered DBDS.
5. The *change accumulation data set* (if present) accumulates the change records that occurred since the image copy data set was made. The change accumulation data set is used as the first source of change records.
6. The *DSLOG data set* (if present) accumulates the change records for the DBDS that have occurred since the change accumulation data set was made. It includes all remaining change records with the exception of those found in most recent log data sets. (This data set will not be supported by future releases of IMS).
7. The *log data set* (if present) for this DBDS contains the change records that have occurred since the DSLOG data set was made. A volume of the log data set contains the final change records available for this DBDS.

5.1.2 Full Recovery—Output

The output from the database recovery utility consists of the following items:

1. The **DBDS**, which is fully recovered.
2. The RECON data set that is updated with the **RECOV** record, to indicate the time-stamp of the recovery of the DBDS.
3. A message, written to a **SYSPRINT** data set, that indicates that the DBDS was successfully recovered; or a diagnostic message indicating the type of failure and the reasons for it.

5.1.3 Full Recovery—Example

Figure 59 shows the GENJCL command used to generate the recovery JCL of a database.

```
GENJCL.RECOV DDN(DBGAMAP) DBD(DBGAMAP)
```

Figure 59. DBRC Command for Recovery

Figure 60 on page 63 shows the skeletal recovery JCL, used by the GENJCL command shown in Figure 59.

Figure 61 on page 64 shows the recovery JCL generated by the GENJCL.RECOV command of Figure 59.

```
//NOAUTH   EXEC PGM=DSPURX00,COND=(O,NE)
//*
//* SET AUTHORIZATION TO NO AUTH TO PREVENT ACCESS DURING RECOVERY
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//JCLPDS   DD DSH=STIMS220.JCLLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 CHANGE.DB DBD(%DBNAME) NOAUTH
/*
//*
//DEFINE EXEC PGM=IDCAMS,COND=(0,LT)
//*
//* DELETE/DEFINE DATABASE DATASET
//*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DSN=STIMS220.DBG.UTIL(%DBNAME),DISP=SHR
//*
//RECOVER EXEC PGH=DFSRRC00,
//        PARM='UDR,DFSURDB0,%DBNAME',REGION=4096K
//*
//* RECOVER %DBNAME DATABASE DATASET
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//%DBDDN   DD DSN=%DBDSN,DISP=SHR,AMP=(,BUFND=30')
//DFSUDUMP DD DSN=%ICDSN,DISP=SHR,DCB=BUFNO=10
%DELETE (%CADSN EQ ‘ ’)
//DFSUCUM  DD DSN=%CADSN,DISP=SHR
%ENDDEL
%DELETE (%CADSN NE ‘ ’)
//DFSUCUM  DD DUMMY
%ENDDEL
%SELECT RLDS((%DBNAME,%DBDDN),ALL)
//DFSULOG  DD DSN=%LOGDSN,DISP=SHR
%ENDSEL
%DELETE (%LOGSEL EQ 'YES')
//DFSULOG  DD DUMMY
%ENDDEL
//DFSVSAMP DD DSN=R82123R.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD *
S %DBNAME %DBDDN DFSUDUMP      %DBNAME RESTORE CONTROL
/*
//*
//AUTH   EXEC PGM=DSPURX00,COND=(O,NE)
//*
//* SET AUTHORIZATION TO AUTH TO ALLOW ACCESS AFTER RECOVERY
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//JCLPDS   DD DSN=STIMS220.JCLLIB,DISP=SHR
//JCLOUT   DD SYSOUT=*,INTRDR)
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 CHANGE.DB DBD(%DBNAME) AUTH
/*
```

Figure 60. Skeletal JCL for Recovery

```
//NOAUTH   EXEC PGM=DSPURX00
//STEPLIB   DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//IMS       DD DSN=SYSC.STIMS220.DBDLIB,DISP=SHR
//JCLPDS    DD DSN=STIMS220.JCLLIB,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//SYSINDD
CHANGE.DB DBD(DBGAMAP) NOAUTH
/*
//*
//DEFINE   EXEC PGM=IDCAMS,
//          COND=(O,LT)
//*
//* DELETE/DEFINE DATABASE DATASET
//*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DSN=STIMS220.DBG.UTIL(DBGAMAP),DISP=SHR
//*
//* RECOVER DATABASE DATASET
//*
//RECOVER  EXEC PGM=DFSRRC00,
//         PARM='UDR,DFSURDBO,DBGAMAP',REGION=1800K
//*
//* RECOVER DBGAMAP DATABASE DATASET
//*
//STEPLIB   DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//IMS       DD DSN=SYSC.STIMS220.DBDLIB,DISP=SHR
//DBGAMAP   DD DSN=STIMS220.DBG.DBGAMAP,DISP=OLD
//DFSUDUMP  DD DSN=STIMS220.DBG.DBGAMAP.IC4,DISP=SHR,AMP=('BUFND=30')
//DFSUCUM   DD DUMMY
//DFSULOG   DD DUMMY
//DFSVSAMP  DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN     DD *
S DBGAMAP DBGAMAP DFSUDUMP    DBGAMAP RESTORE CONTROL
/*
//AUTH     EXEC PGM=DSPURX00,COND=(O,NE)
//STEPLIB   DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//IMS       DD DSN=STIMS22X.DBDLIB,DISP=SHR
//JCLPDS    DD DSN=STIMS220.JCLLIB,DISP=SHR
//JCLOUT    DD SYSOUT=(S,INTRDR)
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
CHANGE.DB DBD(DBGAMAP) AUTH
/*
```

Figure 61. Generated JCL for Recovery

5.2 Time-Stamp Recovery

A time-stamp recovery of a DBDS creates a DBDS at the same level as it was at the specified time.

The time-stamp for a DBDS recovery can specify any tme when the DBDS was not being updated (that is, any time for which no ALLOC record for the DBDS exists in the RECON).

Time-stamp recovery is typically used in a batch environment to restore the databases to the state they were at the beginning of a batch run. For example, a time-stamp recovery can be used when the next step of a job aborted or had the wrong input, and a rerun of the entire job stream, without batch backout for all the steps, is required.

5.2.1 Example of a Time-Stamp Recovery

Figure 62 shows three examples of a time-stamp recovery.

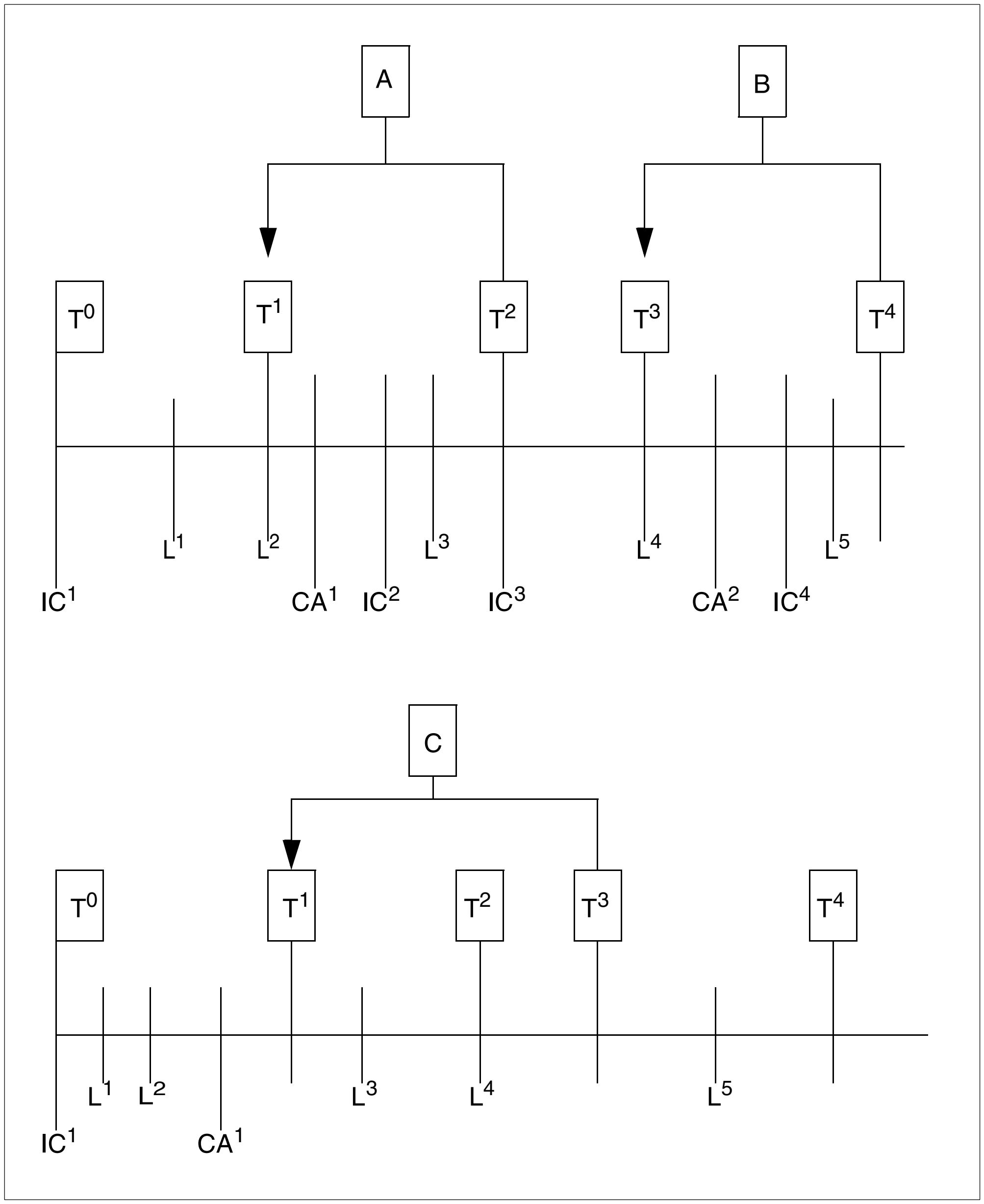


Figure 62. Time-Stamp Recovery of a DBDS

1. **EXAMPLE A**

   At time *T2*. a GENJCL.RECOV for the DBDS is run. The time-stamp specified for the recovery is the time *T1*, the stop time of a log volume.

   The Recovery Control utility generates a job for the database recovery utility that includes ddnames for all the recovery data sets needed in the recovery to time T1. The requested data sets are:

   - The image copy (*IC1*) data set created at time T0.
   - All log data sets created between time TO and time T1 (*L1 + L2*)

   After the recovery, a database image copy utility for the DBDS is run (IC3). At time T2, the DBDS is as it was at time T1, with an image copy data set as a backup.

2. **EXAMPLE B**:

   The time-stamp recovery from time *T4* to time *T3* is similar to that of example A, because an image copy data set has been created after the previous time-stamp recovery (at time T2).

3. **EXAMPLE C**:

   Performing a time-stamp recovery from time T3 to time T1, the DBDS is as it was at time T1, and the log data sets L3 and L4 are not used in any future recovery.

   The result of a GENJCL.RECOV command for a full recovery to be run at *T4* of example *C* now includes:

   - The image copy data set created at time T0 (*IC1*)
   - The change accumulation data set created between time TO and time T1 (*CA1*)
   - The log data set with stop time between T3 and T4 (*L5*).

5.2.2 Recommendations after Doing a Time-stamp Recovery

The database image copy utility should be run after the time-stamp recovery of a DBDS.

This image copy should be run before allowing IMS to do any further processing on the DBDS.

Future recoveries of this DBDS are greatly simplified by the presence of an image copy data set that reflects the time-stamp recovery.

5.3 BMP Recovery

Recovery from BMP failures are more complicated than recovery from batch failures. The BMP runs in the same environment with all the online transactions. The dynamic transaction backout should have returned the databases to the last checkpoint automatically unless a message has been produced to indicate that batch backout is required. The first step in any BMP recovery is to detemine if batch backout is required.

In most situations it is easier to restart the failed BMP than to try to rerun it. Consider the following recommendations as standard for all BMP programs:

- · All BMP programs should issue checkpoints except in the following cases:
  - The highest PROCOPT is GO. A processing intent of RD or higher may cause enqueue and affect online transaction performance. For more details about enqueue, see 1.3.6, “Program Isolation and Enqueue” on page 4.
  - The programs are message-driven and have a MODE of single (SNGL). In this case there is an automatic sync point any time the prograrn reads a new message from the queue (GU). For more details about SYNC points, see 1.3.12, “Commit Points” on page 5 Each program should have a unique checkpoint ID prefix to ensure unique checkpoint IDs.
- The frequency of checkpoints (number of updates between checkpoints) should be made variable without requiring program code changes (parameter driven).

When restarting a BMP, the following information applies:

- The checkpoint of LAST is valid while the OLDS is available.
- The OLDS should be available until it is reused, not just archived.
- The checkpoint must be unique, or unpredictable results can occur.
- When restarting from an RLDS/SLDS, the correct checkpoint ID must be used (LAST is only valid from an OLDS).
- Backout to any checkpoint other than the last should never be done unless only one PSB was updating all the affected databases.

Note: If a program cannot provide a unique checkpoint ID, IMS always provides (together with the program checkpoint ID) a 12 byte checkpoint ID.

This system-provided ID consists of the time/date stamp and has the form IIDDDHHMMSST. In this ID the IIDDD is the region ID and HHMMSST is the actual time of the checkpoint expressed in hours (HH), minutes (MM), seconds (SS), and tenths of a second (T). These 12 bytes are enough to provide uniqueness.

5.4 Batch Backout

The batch backout utility is used to recover databases to a point before a program start time or to a given checkpoint.

The batch backout utility operates as a normal IMS batch job using the PSB of the program whose effects are to be backed out.

5.4.1 Batch Backout: Input

The input to the batch backout utility consists of:

- Log data sets (RLDS, SLDS) containing the database updates to be backed out:
  - When the updates to be backed out are from a batch job, the complete log from a single run of that job must be provided.
  - When the database updates to be backed out were done in an online IMS system, all the log data sets must be included, from the last SYNC point of the program being backed out, to the point where all the databases being backed out have been stopped.
- The databases whose updates are to be backed out. All the data sets related to the databases in the PCBs used for the updates must be provided.

To determine what is to be backed out, the following optional control statements may be provided:

- **CHKPT** statement

  The CHKPT statement identifies an earlier checkpoint to back out to. This control statement is valid ONLY if the input log is from an IMS batch job that did not use IRLM.

  Do not use the CHKPT statement when backing out a BMP. Batch backout used for backing out BMP always uses the first checkpoint that is encountered when reading backwards from the end of the log data set.

- **COLDSTART** statement

  The COLDSTART statement is used to back out all databases with incomplete online updates (MPP or BMP), using a chosen PSB, to the last SYNC point. When this control statement is specified, the last OLDS allocated to the online IMS system must be used as input.

- **DATABASE** statement

  The DATABASE statement is used for a backout that requires more than 11 log data sets, and it is only valid for IMS online regions.

5.4.2 Batch Backout: Output

The output from the batch backout utility consists of:

- The input databases with the changes made by the incomplete transactions or jobs backed out to the last SYNC point (or, if the CHKPT statement was used, to the specified checkpoint).
- An output log that must be kept as future input to the database recovery utility, in case a forward recovery has to be performed on any of the backed out data bases.

5.4.3 Batch Backout of a Normally Terminated Batch Job

A batch job, using DBRC but without IRLM, terminates normally. When this normally terminated job has to be completely backed out, the DBRC parameter on the JCL must have the value *C*.

This is the only case where the batch backout utility can back out updates of a normally terminated job.

5.4.4 Batch Backout: Log Example

Batch logging writes a "before" image and an "after" image of any updated segment into the log data set. This enables backout or forward recovery to replace the correct image of the segment on the database.

During forward recovery the "after" image is used, and during backout the "before" image is used.

Batch backout replaces the segment on the data base with the "before" image from the input log, but also follows the logic of writing the “after image” (not the "before image") to the new log data set.

This log created by the run of the batch backout must be treated like any log created by an updating subsystem. It must be retained until the next image copy of all the affected databases is taken.

Figure 63 shows an example of the batch backout logs being used as input to a recovery.

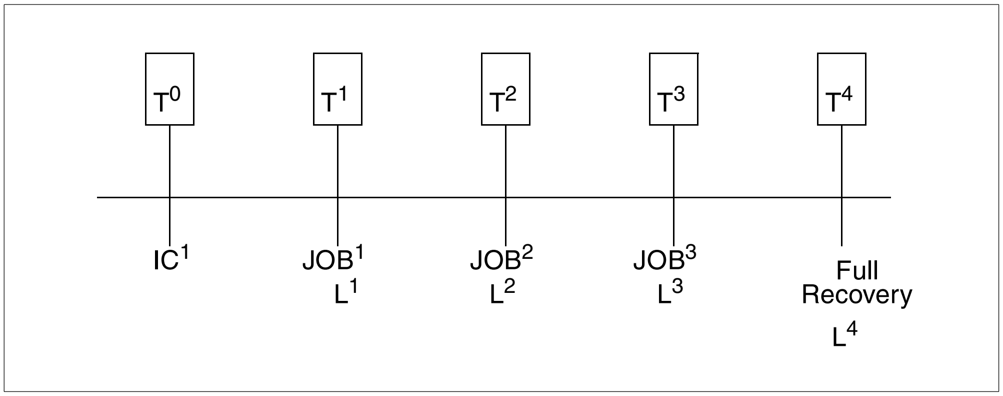


Figure 63. Batch Backout Logs

The following events occur:

1. At time *T0* an *IC1* is taken.
2. At time *T1,* the batch job, JOB1, performs the following actions:
    1. Updates the segments A and B
    2. Issues the checkpoint A7000001
    3. Updates the segment D
    4. Fails for some reason

    The log data set (L1) contains the following change records:

    - Segment A before image (*A*)
    - Segment A after image (*A'*)
    - Segment B before image (*B*)
    - Segment B after image (*B'*)
    - Checkpoint A7000001
    - Segment D before image (*D*)
    - Segment D after image (*D'*)
3. At time *T2,* a batch backout (JOB2) is run using the checkpoint A700001.

    All updates after this checkpoint are backed out. Only segment D is affected by the backout.

    The log data set (*L2*) contains the following change records:

    - Segment D after image (*D*)
4. At time *T3,* a batch job (JOB3) performs the following actions:
    1. Updates the segment E
    2. Issues a checkpoint A7000002
    3. Ends normally

    The log data set (*L3*) contains the following change records:

    - Segment E before image (*E*)
    - Segment E after image (*E'*)

5. At time *T4* a full recovery is run.

   The required inputs are:

   - IC1 data set
   - L1 data set
   - L2 data set
   - L3 data set

6. Conclusion: If the batch backout log (*L2*) is not used, then segment D is updated by the record on log L1 (only). Thus, it becomes D' and not the correct segment D.

5.5 Recovery from an Online Image Copy

Conisider the following scenario:

1. A BMP job is running and updating databases.
2. Concurrently, an online image copy (OIC) is running using the same databases.
3. After having finished the previous job, a recovery for those databases is required.

The following steps must be performed:

1. Deallocate the data bases.
2. Archive any OLDS that was online during the execution of the BMP and the OIC jobs. This can be accomplished by issuing a /DBR command (without the NOFEOV parameter) for the databases. The result is an OLDS switch and an automatic archive.

   Note: If automatic archiving is not used, a manual archive is needed.

3. Issue the GENJCL.RECOV. The results of the GENJCL are shown in the Figure 64 on page 71
4. Run the recovery using the proposed JCL.

Note: Performing a recovery without archiving the OLDS (/DBR) produces the message shown in Figure 65.

```
//NOAUTH EXEC PGM=DSPURX00,COND=(O,NE)
//*
//* SET AUTHORIZATION TO NOAUTH TO PREVENT ACCESS DURING RECOVERY
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//JCLPDS   DD DSN=STIMS220.JCLLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD
 CHANGE.DB DBD(DBGAMAPDBGAMAP) NOAUTH
/*
//*
//DEFINE EXEC PGM=IDCAMS,COND=(0,LT)
//*
//* DELETE/DEFINE DATABASE DATASET
//*
//SYSPRINT   DD SYSOUT=
//SYSIN      DD DSN=STIMS220.DBG.UTIL(DBGAMAP),DISP=SHR
//*
//RECOVER EXEC PGM=DFSRRC00,
//        PARM='UDR,DFSURDBO,DBGAMAP',REGION=4096K
//*
//* RECOVER DBGAMAP DATABASE DATASET
//*
//STEPLIB    DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT   DD SYSOUT=*
//IMS        DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DBGAMAP    DD DSN=STIMS220.DBG.DBGAMAP,DISP=SHR AMP=('BUFND=30')
//DFSUDUMP   DD DSN=STIMS220.DBG.DBGAMAP.IC3,DISP=SHR,DCB=BUFNO=10)
//DFSUCUM    DD DUMMY
//DFSULOG    DD DSN=STIMS220.RLDS.G00l1V00,DISP=SHR
//DFSVSAMP   DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN      DD *
S DBGAMAP DBGAMAP DFSUDUMP     DBGAMAP RESTORE CONTROL
/*
//*
//AUTH EXEC PGM=DSPURX00,COND=(O,NE)
//*
//* SET AUTHORIZATION TO AUTH TO ALLOW ACCESS AFTER RECOVERY
//*
//STEPLIB DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//IMS       DD DSN=STIMS22X.DBDLIB,DISP=SHR
//JCLPDS    DD DSN=STIMS220.JCLLIB,DISP=SHR
//JCLOUT    DD SYSOUT=(*,INTRDR)
//SYSPRINT  DD SYSOUT=*
//SYSIN DD
 CHANGE.DB DBD(DBGAMAP) AUTH
/*
```

Note: In this example JCL for only one Data Base is shown.

Figure 64. Generated JCL for Recovery Example

```
         IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4
  GENJCL.RECOV GROUP(DBGGRP1) JOB(DBGJOBR) ONEJOB
DSPO274I UTILITY REQUESTING AN UNARCHIVED ONLINE LOG
DSPO2O9I PROCESSING TERMINATED WITH CONDITION CODE = 12
```

Figure 65. Error Messages for Recovery without /DBR

5.6 Recovery with a Merge-Needed Situation

When DBRC detects a situation where more than one log may have been used in a given period of time, DBRC creates a merge-needed (MN) record in the RECON. This situation can arise for the following reasons:

- The database was involved in block level sharing with two subsystems active at the same time.
- The database was deallocated from an online subsystem *with* the NOFEOV parameter to allow a batch update job to run and then was reallocated to the online subsystem.

  In this case a merge-needed record is placed in the RECON.
- The database was deallocated from an online subsystem *without* the NOFEOV parameter to allow a batch update job to run and then was re-allocated to the online subsystem.

  In this case a merge-needed record is not placed in the RECON. When a recovery is attempted, DBRC is able to determine that a merging of log data sets is required before a recovery could be run.

A change accumulation is *required* to merge the logs and remove the merge-needed situation before the recovery JCL can be generated.

Once the CA has been run, the normal GENJCL.RECOV can be used to create the required recovery JCL.

5.6.1 Merge-Needed Record Created

In the following example, the database has been deallocated with the NOFEOV parameter and then reallocated to the IMS system after a batch update job has been run.

The same RLDS data set (STIMS220.RLDS.G0003V00) has one allocation time before and one after the batch log (STIMS220.DBG.B01LOG.G0005V00) allocation time.

This could happen regardless of the number of OLDS switches which may have occurred while the databases were off-line, depending on the archive frequency.

Figure 66 on page 73 and Figure 67 on page 74 shows RECON listings created by LIST.HISTORY commands. These listings also include the CA record created by the change accumulation to merge the two logs.

Note: The MN record exists for compatibility purpose only. Recovery related utilities no longer use this record.

```
DB
 DSN=STIMS220.DBG.DBGAMAP

 ALLOC
 ALLOC   = 88.327  12:56:53.0*      START   = 88.327     12:41:15.3
 DEALLOC = 88.327  12:59:10.3       DSSN=00000004

PRILOG
 START   = 88.327  12:41:15.3    STOP    = 88.327      13:03:26.1
 SSID=I220        #DSN=1         IMS

DSN=STIMS220.RLDS.G0003V00                                UNIT=3400
 START   = 88.327  12:41:15.3    STOP    = 88.327      13:03:26.1
 FILE SEQ=0001    #VOLUMES=0001 -VOLSER-         -STOPTIME-
                                STIMS5    88.327      13:03:26.1
MRGND
 DBD=DBGAMAP DDN=DBGAMAP           MN TIME = 88.327   13:00:21.6*
 DSID=001    DSSN=00000004         SUBSYSTEM COUNT=2

ASSOCIATED SUBSYSTEM INFORMATION:
 -SSID-       -PRILOG START TIME-        -ALLOCATION TIME STAMP-
 I220         88.327   12:41:15.3           88.327  12:56:53.0
 R82123R5     88.327   13:00:13.5           88.327  13:00:21.6

ALLOC
 ALLOC   = 88.327  13:00:21.6       START   = 88.327     13:00:13.5
                                    DSSN=00000004
PRILOG
 START   = 88.327  13:00:13.5*   STOP    = 88.327     13:00:38.5
 SSID=R82123R5    #DSN=1         IMS

DSN=STIMS220.DBG.B01LOG.G0003VO00                         UNIT=3400
 START   = 88.327  13:00:13.5    STOP    = 88.327      13:00:38.5
 FILE SEQ=0001    #VOLUMES=0001  -VOLSER-        -STOPTIME-
                                  STIMS5    88.327   13:00:38.5
ALlOC
 ALLOC  = 88.327   13:02:34.9*      START   = 88.327     12:41:15.3
                                    DSSN=00000004
PRILOG
 START  = 88.327   12:41:15.3*   STOP   = 88.327      13:03:26.1
 SSID=I220        #DSN=1         IMS

 DSN=STIMS22O.RLDS.GOOO3VOO                               UNIT=3400
 START  = 88.327   12:41:15.3    STOP   = 88.327      13:03:26.1
 FILE SEQ=0001   #VOLUMES=0001   -VOLSER-        -STOPTIME-
                                  STIMS5   88.327    13:03:26.1
CA
 DSN=STIMS22O.DBG.CALOG.GOOO2VOO                      FILE SEQ=1
 CAGRP=DBGCA     UNIT=3400
 STOP  = 88.327    13:03:26.1*    VOLS DEF=1      VOLS USED=1
                                  VOLSER=STIMS5
RUN    = 88.327    15:49:25.9
  - DBD-   -DDN-      -PURGETIME- -CHG -CMP-  -LSN-       -DSSN-
  DBGAMAP DBGAMAP 88.327 12:54:46.4 YES YES 000000000000 00000006
  DBGAMAX DBGAMAX 88.327 12:55:01.6 NO  YES 000000000000 00000000
  DBGAMBP DBGAMBP 88.327 12:55:16.6 YES YES 000000000000 00000006
  DBGAMBX DBGAMBX 88.327 12:55:32.5 YES YES 000000000000 00000006
  DBGAMBY DBGAMBY 88.327 12:55:48.6 YES YES 000000000000 00000004
```

Figure 66. DB History RECON Listing showing a merge-needed situation

```
      HISTORY OF STIMS22O.DBG.DBGAMAP    DBD:DBGAMAP       DDN:DBGAMAP
ALLOC ---> 88327     12:56:53.0  SESSION      : START-88327,12:41:15.3   STOP-88327,13:03:26.1
           22.11.88  TUESDAY     DEALLOC      : 88327,12:59:10.3
                                 JOBNAME/SSID : I220
                                 RLDS-DSN     : STIMS22O.RLDS.G0003V00
                                 VOLSER(S)    : STIMSS ---> DSLOG=N CHANGES=Y

ALLOC ---> 88327     13:00:21.6  SESSION      : START-88327,13:00:13.5   STOP-88327,13:00:38.5
           22.11.88  TUESDAY     DEALLOC      : -
                                 JOBNAME/SSID : R82123R5
                                 RLDS-DSN     : STIMS220.DBG.B0lLOG.G0005V00
                                 VOLSER(S)    : STIMS5 ---> DSLOG=N CHANGES=Y

ALLOC ---> 88327     13:02:34.9  SESSION      : START-88327,12:41:15.3   STOP-88327,13:03:26.1
           22.11.88  TUESDAY     DEALLOC      : -
                                 JOBNAME/SSID : I220
                                 RLDS-DSN     : STIMS220.RlDS.G0003V00
                                 VOLSER(S)    : STIMS5 ---> DSLOG=N CHANGES=Y
```

Figure 67. Merge-Needed RECON Listing (Created by DBICF-Partial)

If a GENJCL.RECOV command is attempted while a merge-needed situation exists, then the message in Figure 68 is issued.

```
         IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4
  GENJCL.RECOV GROUP(DBGGRPI) JOB(DBGJOBR) ONEJOB
DSPO2BlI LOG MERGE REQUIRED FOR RECOVERY REQUEST
DSPO2O9I PROCESSING TERMINATED WITH CONDITION CODE = 12
```

Figure 68. Merge Needed RECON Listing (Termination Message)

5.6.2 Merge-Needed Record Not Created

In this example, a database has been deallocated without the NOFEOV parameter and then reallocated to the IMS system after a batch update job has been run.

In this case a merge-needed record has not been created. The PRILOG record for the online system has multiple RLDS records:

- The first RLDS (STIMS220.RLDS.G0007V00) starts and stops before the batch log (STIMS220.DBG.BO1LOG.G000BV00).
- The second RLDS (STIMS220.RLDS.G000BV00) starts before the batch log and stops after it.
- The third RLDS (STIMS220.RLDS.G000V009V00) starts and stops after the batch log.

The start and stop time of the batch subsystems PRILOG record contains a time within the time period of the PRILOG record of the online systerm.

Because of this, DBRC needs to ensure that the log change records are processed in the correct time sequence, and the only way to do that is to merge the log files with a change accumulation run. Figure 69 on page 75 shows a RECON listing taken after all the RLDS have been created.

```
LIST.HISTORY DBD(DBGAMAP)

AlLOC
  ALLOC   = 88.328        12:49:39.5    START   = 88.328    12:46:46.0
  DEALLOC = 88.328        12:50:35.4    DSSN=00000010

PRILOG
 START    = 88.328        12:46:46.0 STOP =    88.328    12:54:06.4
 SSID=I220               #DSN=3      IMS

 DSN=STIMS220.RLDS.G0007V00                                 UNIT=3400
 START    = 88.328        12:46:46.0    STOP = 88.328    12:50:42.6
 FILE SEQ=0001    #VOLUMES=0001 -VOLSER-            -STOPTIME-
                                 STIMS5        88.328    12:50:42.6

 DSN=STIMS220.RLDS.G0008V00                                 UNIT=3400
 START    = 88.328        12:50:42.6    STOP = 88.328    12:52:24.5
 fILE SEQ=0001    #VOLUMES=0001 -VOlSER-            -STOPTIME-
                                 STIMS5        88.328    12:.52:24.5

 DSN=STIMS220.RLDS.G0009V00                                 UNIT=3400
 START    = 88.328        12:52:24.5     STOP =88.328    12:54:06.4
 FILE SEQ=0001    #VOLUMES=0001  -VOLSER-            -STOPTIME-
                                 STIMS5        88.328    12:54:06.4

ALLOC
 ALLOC    = 88.328         12:51:28.2*   START  = 88.328  12:51:19.3
                                         DSSN=00000011
PRILOG
 START    = 88.328         12:51:19.3    STOP   = 88.328  12:51:45.2
 SSID=R82123RK            #DSN=1         IMS

 DSN=STIMS220.DBG.BOlLOG.G0008V00                           UNIT=3400
 START    = 88.328         12:51:19.3    STOP   = 88.328  12:51:45.2
  FILE SEQ=0001   #VOLUHES=0001  -VOLSER-            -STOPTIME-
                                  STIMS5         88.328  12:51:45.2

ALLOC              ,
 ALLOC    = 88.328         12:53:42.8*   START  = 88.328  12:46:46.0
                                         DSSN=00000012
PRILOG
 START    = 88.328         12:46:46.0    STOP   = 88.328  12:54:06.4
 SSID=I220                #DSN=3         IMS

 DSN=STIMS220.RLDS.G0007V00                                 UNIT=3400
 START    = 88.328         12:46:46.0    STOP   = 88.328  12:50:42.6
 FILE SEQ=0001    #VOLUMES=0001  -VOLSER-        -STOPTIME-
                                  STIMS5         88.328  12:50:42.6

DSN=STIMS220.RLDS.G0008V00                                 UNIT=3400
 START    = 88.328         12:50:42.6    STOP   = 88.328  12:52:24.5
 FILE SEQ=0001    #VOLUMES=0001 -VOLSER-         -STOPTIME-
                                 STIMS5          88.328  12:52:24.5

DSN=STIMS220.RLDS.G0009V00                                 UNIT=3400
 START    = 88.328         12:52:24.5    STOP   = 88.328  12:54:06.4
 FILE SEQ=0001    #VOLUMES=0001 -VOLSER-         -STOPTIME-
                                 STIMS5          88.328  12:54:06.4
```

Figure 69. Merge-Needed RECON Listing (Partial Printout)

Figure 70 uses the DBICF DB history function to list the same time period. It clearly shows the sequence of allocations.

```
                             HISTORY OF STIMS22O.DBG.DBGAMAP
ALLOC ---> 88328    12:49:39.5 SESSION      : START-88328,12:46:46.0  STOP-88328,12:54:06.4
           23.11.88 WEDNESDAY  DEALLOC      : 88328,12:50:35.4
                               JOBNAME/SSID : I220
                               RLDS-DSN     : MULTIPLE RLDS-DSN'S (ISSUE A LIST.LOG-CMD.)
                               VOLSER(S)    : -

ALLOC ---> 88328    12:51:28.2 SESSION      : START-88328,12:51:19.3  STOP-88328,12:51:45.2
           23.11.88 WEDNESDAY  DEALLOC      : -
                               JOBNAME/SSID : R82123RK
                               RLDS-DSN     : STIMS22O.DBG.B0lLOG.G000BV00
                               VOLSER(S)    : STIMSS ---> DSLOG=N CHANGES=Y

ALLOC ---> 88328    12:53:42.8 SESSION      : START-88328,12:46:46.0  STOP-88328,12:54:06.4
           23.11.88 WEDNESDAY  DEAlLOC
                               JOBNAME/SSID : I220
                               RLDS-DSN     : MULTIPLE RLDS-DSN'S (ISSUE A LIST.LOG-CMD.)
                               VOLSER(S)    : -
```

Figure 70. Data Base History Listing (Produced by DBICF)

5.7 Daylight Savings Time for IMS Users

On a Sunday in autumn, most states change the official time from daylight savings time to standard time. At some time on that Sunday, the MVS machine clock is set back one hour. Normally, this occurs at 2:00 AM.

This resetting of the MVS machine clock has serious implications for an IMS subsystem that is running during this period. Database recovery depends greatly upon the log record time-stamps, and the IMS time-stamp cannot decrease its own value.

5.7.1 IMS without the Use of DBRC

A number of options exist for the IMS subsystem that is not using DBRC:

1. Do not run any IMS subsystems during the "duplicate" time period (the one hour period between the original 2:00 AM and the new 2:00 AM). This option is still the easiest to implement.
2. Bring all IMS subsystems down immediately prior to resetting the clocks. Reset the clocks. Backup all databases updated in the preceding hour. Restart the IMS subsystems.
3. If no time is available for backing up the databases between the time the subsystems were brought down, the clocks were reset, and the subsystems were restarted, the operators must remember which logs were created under the daylight savings time and which logs were created under the standard time.
4. Any database recovery using a backup created under daylight savings time must be performed in two steps:
   - Recover the database up to the time change and use the image copy and the log data sets created prior to the time change.
   - Perform a second recovery using the logs created after the time change.
5. All IMS subsystems must be terminated prior to the time change, for the following reasons:
   - IMS subsystems generally use the hardware clock (STCK instruction), rather than the MVS TIME macro, to obtain the time.

- At subsystem initialization time, IMS computes an adjustment factor to convert the STCK value to a date and time. This adjustment factor is recomputed every midnight for IMS.
- Resetting the MVS clock while an IMS subsystem is running means that the subsystem does not recognize the time immediately.
- Other subsystems started after the change will use the new time; these new subsystems are running one hour behind the old subsystems.
- The IMS subsystems running prior to the time change continue their executions past midnight (old time), then they reset their internal clocks back one hour, causing negative time breaks in the logs.

5.7.2 IMS and DBRC with Registered Databases

When the IMS subsystem is using DBRC with registered databases (Recovery or Sharing control environment), IMS does not have many options, because:

- Almost all of the record types in the RECON data sets have time-stamps. Some of these time-stamps are provided by the IMS modules that invoke DBRC, and some of these time-stamps are obtained by the DBRC code (with the TIME macro).
- DBRC assumes that the time never goes backwards, and it is coded with this basic assumption.

Therfore, the DBRC user has only the following choices:

- Terminate all IMS subsystems prior to resetting the MVS clock.
- Do not run any new IMS subsystems for one hour.

Note: There are no other options available if the user expects DBRC to perform correct recovery automatically.

5.7.3 IMS, DBRC and Log Control

The IMS subsystem that is only using DBRC to manage the online logs (that is, no databases are registered, and the RECON data sets were initialized with RECOVCTL) has the same options as the IMS subsystem without DBRC.

When the online system is brought down, the operator must switch to a clean set of RECON data sets (that is, no PRIOLDs or PRISLDS records registered for the subsystem) and then bring the online system back up with a COLD START.

Chapter 6. Abnormal Recovery Situations

This chapter explains non-normal operations, such as when a recovery operation cannot be performed using the primary copy (or the last version) of the required data sets (image copy, log or other).

6.1 Recovery with an Invalid IC Data Set

What happens to the log data set required for recovery if an IC data set becomes unavailable? When change accumulation is used, log data sets that were used as input to the change accumulation may still be required.

The use of the change accumulation does not remove the need to retain all log data sets until an image copy of all affected databases has been taken.

6.1.1 Example of an Invalid IC Data Set

In Figure 71, only a single IC data set is produced by each backup job.

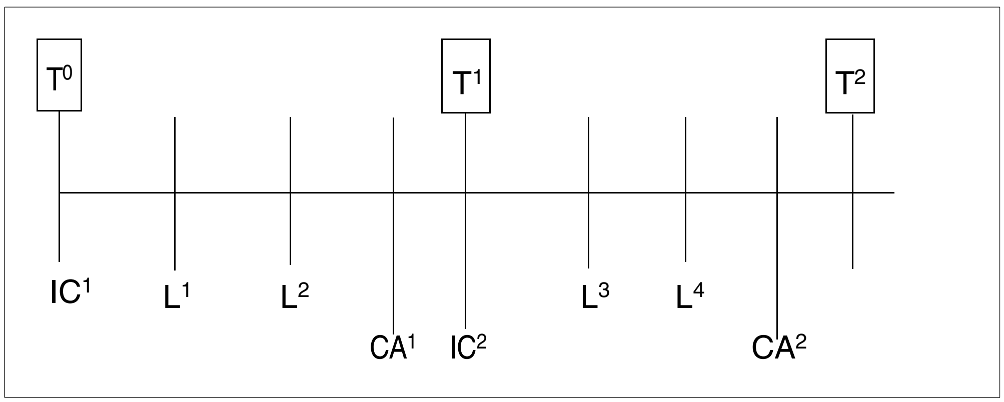


Figure 71. Full Recovery Using the IC-1 Version

This example shows:

1. At time T0, an image copy (IC1) is taken.
2. Between T0 and T1, two log data sets (L1 and L2) are produced and a change accumulation is run to merge the two logs (CA1).
3. At time T1, a second image copy (IC2) is taken.
4. Between T2 and T3, two log data sets (L3 and L4) are produced and then a second change accumulation is run to merge the two logs (CA2).
5. At time T2, a full recovery is required. The IC2 data set taken at T1 has been marked invalid. DBRC generates the recovery JCL with the following input:
   - IC1 data set
   - CA1 data set
   - L3 data set
   - L4 data set

6.1.2 Conclusions

The following conclusions can be extracted from the example shown in Figure 71 on page 79:

1. The CA2 data set is not used.

   When DBRC detects the error on the IC2, this data set has a given time-stamp. from that time-stamp on, only log data sets are used.

2. Producing the change accumulation data set CA2 does not eliminate the need to retain all log data sets until the next IC is taken.

   It is the NOTIFY.IC command (generated by the IC job) that marks the log data sets as no-longer-required.

Note: Retain all log data sets until the DBRC determines that they are no longer required. When DASD logs are used and space considerations are a problem, the logs can be archived to tape with the log archive utility run with DBRC active.

6.2 Removing Image Copy Information from RECON

The following example presents a situation where there is a need to remove the information about the last IC data set.

6.2.1 Example Scenario

Consider the example scenario depicted in Figure 72.

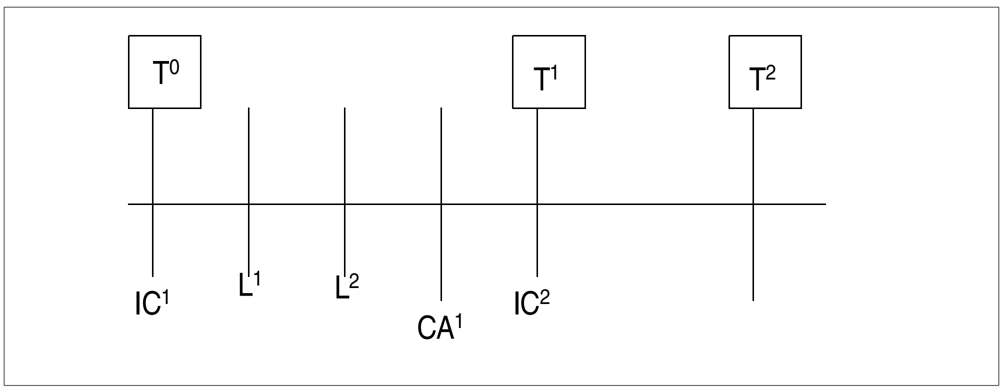


Figure 72. Removing the Last IC Information

This example shows:

1. At time T0, an image copy (IC1) is taken.
2. Between times T0 and T1, two batch logs are produced, and a change accumulation is run (CA1) to merge the two logs.
3. At time T1, a second image copy (IC2) is taken.
4. At time T2, (for some valid reason) it is decided to remove the image copy information taken at time T1.
5. Figure 73 shows the RECON information after the second IC has been run and before any change has been applied.

```
HISTORY OF STIMS220. DBG.DBGAMAP DBD:DBGAMAP DDN:DBGAMAP
IC1-> 88313 18:45:17.7 DSN : STIMS220.DBG.DBGAMAP.IC1
08.11.88 TUESDAY FILE-SEQ: 1
NBR RCDS: 155
VOLSER : STIHS5
ALLOC:88313 18:53:54.1 SESSION : START-88313,18:53:46.4 STOP-88313,18:54:05.9
08.11.88 TUESDAY DEALLOC : -
JOB/SSID: R82123R9
RLDS-DSN: STIMS220.DBG.BOlLOG.GOOOlVOO
  VOLSER : STIHS6 ---> DSLOG=N CHANGES=Y
ALLOC:88313 18:57:14.3 SESSION : START-88313,18:57:05.6 STOP-88313,18:57:25.8

  08.11.88 TUESDAY DEALLOC : -
  JOB/SSID: R82123RX
  RLDS-DSN: STIMS220.DBG.B01LOG.G0002V00O
  VOLSER : STIMS6 ---> DSLOG=N CHANGES=Y
CA -> 8314 12:56:41.5 DSN : STIMS220.DBG.CALOG.G000lV00

  09.11.88 WED'DAY FILE-SEQ: 1
  VOLSER : STIMS5


IC1-> 88314 18:46:22.5 DSN : STIMS220.DBG.DBGAHAP.IC3

  09.11.88 WED'DAY FILE-SEQ: 1
  NBR RCDS: 155
  VOLSER : STIMS5
  VOLSER : STIMS6 ---> DSLOG=N CHANGES=Y
  VOLSER(S) : STIMS5
```

Figure 73. RECON Listing Produced by DBICF (Condensed)

The task of updating the RECON can be performed through a direct DBRC command or through the use of the GENJCL.USER function.

6.2.2 DELETE.IC DBRC Command

The DELETE.IC command removes the IC information. To execute the DELETE.IC command, supply the DBDNAME, DDNAME and the start time of the IC data set. This can be accomplished by listing the RECON information and entering the RECTIME into each DELETE.IC command required. See Figure 74 for an example of the command.

```
DELETE.IC DBD(DBDAMAP) DDN(DBGAMAP) RECTIHE(883141846225)
```

Figure 74. DELETE.IC Command

6.2.3 GENJCL.USER Function

The normal solution (through the use of DELETE.IC commands) can be a tedious task if more than one database is involved, for instance, if the IC was requested and run at DBDSGRP level.

To provide a better solution, Figure 75 shows a GENJCL.USER function to produce the same result for an entire DBDSGRP and the related skeletal JCL.

```
 DBRC Command:

      GENJCL.USER MEMBER(DLETIC) GROUP(DBGGRP1)


Member DLETIC:


//DBRC    EXEC PGM=DSPURX00
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//IHS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
%SELECT IC t(%DBNAME,%DDNAMEl,LAST1
DELETE.IC DBD(%DBNAME) DDN(%DDNAME) RECTIHE(%ICTIME)
%ENDSEL
/*
```

Figure 75. GENJCL.USER Command and Skeletal JCL Member

Figure 76 shows the generated JCL output obtained when using the GEN.ICL.USER command.

```
/DBRC EXEC PGM=DSPURXOO
//STEPLIB DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//IMS   DD DSN=STIMS22X.DBDLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD
DELETE.IC DBD(DBGAMAP7 DDN(DBGAMAP) RECTIHE(883141846225l
/
//DBRC EXEC PGM=DSPURXOO
//STEPLIB DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//IHS   DD DSN=STIMS22X.DBDlIB,DISP=SHR
//SYSPRINT DD SYSOUT=
//SYSIN DD
DELETE.IC DBD(DBGAMAX) DDN(DBGAMAX) RECTIME(883141846385)
/
//DBRC EXEC PGM=DSPURXOO
//STEPLIB DD DSN=SYSC.STIMS220.RESLIB,DTSP=SHR
//IMS   DD DSN=STIMS22X.DBDLIB,DISP=SHR
//SYSPRINT DD SYSOUT=
//SYSIN DD
ELETE.IC DBD(DBGAMBP) DDN(DBGAMBP) RECTIME(883141846545)
/
//
//DBRC EXEC PGM=DSPURXOO
//STEPLIB DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//IHS   DD DSN=STIMS22X.DBDLIB,DISP=SHR
//SYSPRINT DD SYSOUT=
//SYSIN DD
DELETE.IC DBD(DBGAMBX) DDN(DBGAMBX) RECTIME(883141847105l
/*
//*
//DBRC EXEC PGM=DSPURXOO
//STEPLIB DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//IHS   DD DSN=STIMS220.DBDLIB,DISP=SHR
//SYSPRINT DD SYSOUT=
//SYSIN DD
DELETE.IC DBD(DBGAMBY) DDN(DBGAHBY) RECTIHE(883141847263)
/*
```

Figure 76. JCL to Delete IC RECON Information

6.2.4 Additional Details

The commands proposed in this example work successfully. Other points to be considered are:

- If a DELETE.LOG INACTIVE command was issued after time T1, the two RLDS and the CA data sets would have been marked no-longer-needed by the second image copy and therefore deleted from the RECON.
- If the IC was done at a DBDSGRP level, then the deleting of RECON information should also be done using that DBDSGRP. In this way, information for related databases is maintained at the same level.
- Removing IC information from RECON must be performed with a good understanding of the effects that it will have on future recovery processing.
- An IC backup for any database effected by the removal of IC information must be taken as soon as possible. This simplifies any future recovery situations and helps in ensuring database integrity.

6.3 Recovery from a Secondary IC Data Set

This example describes a situation where a secondary image copy data set is used because the first image copy data set is marked as invalid.

6.3.1 Image Copies Listed in RECON

A partial listing of RECON, in Figure 77, shows that there are two image copy data sets for the database DBGAMAP (time-stamp 88.31318:29:37.9):

- STIMS220.DBG.DBGAMAP.IC1
- STIMS220.DBG.DBGAMAP.IC21

```
88.313 18:31:12.2 PARTIAL LISTING OF RECON


DB
 DBD=DBGAMAP


IMAGE
 RUN = 88.313 18:29:37.9a RECORD COUNT =155
 STOP = 00.000 00:00:00.0 BATCH

IC1
  DSN=STIMS220.DBG.DBGAMAP.IC1
  UNIT=3400

  FILE SEQ=0001
VOLS DEF=0001 VOLS USED=0001
VOLSER=STIMS5


  IC2
  DSN=STIMS220.DBG.DBGAMAP.IC21 FILE SEQ=0001
  UNIT=3400 VOLS DEF=0001 VOLS USED=0001
  VOLSER=STIMS5
```

Figure 77. RECON Listing for One DB

6.3.2 Invalid Image Copy Record

For some reason (media failures, for instance), the image copy data set (STIMS220.DBG.DBGAMAP.IC1) is no longer available or valid.

In this case DBRC must be informed that the IC1 data set is not available. This can be done with the CHANGE.IC command.

Figure 78 shows the result of using the CHANGE.IC command with the parameter INVALID for the given time-stamp (RECTIME). This command uses the skeletal JCL that was shown in Figure 60 on page 63.

```
IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4
CHANGE.IC -
INVALID -
DDN(DBGAMAP) -
DBD(DBGAMAP) RECTIME(883131829379)
DSPO2O3I COMMAND COHPLETED WITH CONDITION CODE 00
DSPO22OI COMMAND COMPlETION TIME 88.313 18:50:10.2
IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4
DSPO2llI COMMAND PROCESSING COMPLETE
DSPO2llI HIGHEST CONDITION CODE = 00
```

Figure 78. Change IC Record

6.3.3 Recovery Using the Secondary IC Data Set

Figure 79 on page 84 shows that a recovery of the DBDS DBGAMAP now successfully uses as input the secondary image copy data set.

- STIMS220.DBG.DBGAMAP.IC21

```
//DEFINE  EXEC PGM=IDCAHS,
//        COND=(O,LT)
//*
//* DELETE/DEFINE DATABASE DATASET
//*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DSN=STIMS220.DBG.UTIL(DBGAHAPl,DISP=SNR
//*
//RECOVER    EXEC PGM=DFSRRC00,
//            PARM='UDR,DFSURDB0,DBGAMAP',REGION=4096K
//*
//* RECOVER DBGAHAP DATABASE DATASET
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESlIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DBGAMAP  DD DSN=STIMS220.DBG.DBGAMAP,DISP=SHR,AMP=('BUFND=30')
//DFSUDUMP DD DSN=STIMS220.DBG.DBGAMAP.IC2l,DISP=SHR,DCB=BUFNO=10
//DFSUCUM  DD DUMMY
//DFSULOG  DD DUMMY
//DFSVSAMP DD DSN=R82123R.DBG.UTIL(ISSVBUFl,DISP=SHR
/SYSIN     DD *
S DBGAMAP DBGAMAP DFSUDUMP DBGAHAP RESTORE CONTROL
/*
```

Figure 79. Recover DB with Invalid First IC

6.4 Recovery Using Secondary RLDS

This example describes a situation where DUAL LOGGING is used and the primary log is no longer valid or usable.

6.4.1 Log Records Listed in RECON

A partial listing of RECON in Figure 80 on page 85 shows PRILOG and SECLOG information:

- PRILOG = STIMS220.DBG.B01LOG.G000IV00
- SECLOG = STIMS220.DBG.B01LOG2.G0001V00

```
PARTIAL LISTING OF RECON


LIST.HISTORY DBD(DBGAMBP)
88.315   11:51:34.1   LISTING OF RECON               PAGE 0002
DB
DBD=DBGAMBP           IRLMID=sNULL DMBit=13          TYPE=IMS
IMAGE
88.315   11:51:34.1   LISTING OF RECON               PAGE 0004
RUN      = 88.313     18:30:11.4s RECORD COUNT =90
STOP     = 00.000     00:00:00.0 BATCH
IC1
DSN=STIMS220.DBG.DBGAMBP.IC1  FILE SEQ=0001
UNIT=3400             VOLS DEf=0001 VOLS USED=0001
                      VOLSER=STIHS5


IC2
DSN=STIMS220.DBG.DBGAMBP.IC21 FILE SEQ=0001
UNIT=3400 VOLS DEF=0001 VOLS USED=0001
  VOLSER=STIMS5


ALLOC
ALLOC = 88.315 11:46:25.5 START = 88.315 11:46:19.2
  DSSN=00000001


PRILOG
START = 88.315 11:46:19.2 STOP = 88.315 11:47:13.2
SSID=R82123H1  #DSN=1 IMS

DSN=STIMS220.DBG.BOlLOG.GOOOlVOO UNIT=3400
START = 88.315 11:46:19.2 STOP = 88.315 11:47:13.2
FILE SEQ=0001 #VOLUMES=0001 -VOLSER- -STOPTIME-
  STIMS5 88.315 11:47:13.2


  SECLOG
  START = 88.315 11:46:19.2 STOP = 88.315 11:47:13.2
  SSID=R82123H1 #DSN=1 IMS

  DSN=STIMS220.DBG.BOlLOG2.G000lV00 UNIT=3400
  START = 88.315 11:46:19.2 STOP = 88.315 11:47:13.2
  FILE SEQ=0001 #VOLUMES=0001 -VOLSER- -STOPTIME-
```

Figure 80. RECON Listing of Log Records

6.4.2 PRILOG Record Flag Marked in ERROR

For some reason (such as media failures) the PRILOG record (STIMS220.DBG.B01LOG.G0001V00) is no longer available or valid.

In this case DBRC must be informed that the PRILOG record is not available. This can be done with the CHANGE.PRILOG command.

Figure 81 shows the result of the CHANGE.PRILOG command with the ERROR parameter for the given time-stamp (STARTIME).

```
 IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4 PAGE 0001
CHANGE.PRILOG -
 IMS -
 ERROR -
 STARTIME(883151146192)
DSPO2O3I COMMAND COMPLETED MITH CONDITION CODE 00

DSPO22OI COHMAND COHPLETION TIME 88.315 14:02:42.1

  IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4 PAGE 0002
DSPO2llI COMHAND PROCESSING COMPLETE
DSPO2llI HIGHEST CONDITION CODE = 00
```

Figure 81. Change PRILOG Record

6.4.3 Recovery without the PRILOG Record

Figure 82 shows that a recovery of the DBDS DBGAMBP now successfully uses the secondary log data set as input:

- STIMS220.DBG.BOILOG2.G0001V00

```
//DEFINE EXEC PGM=IDCAMS,
//     COND=(0,LT)
//*
//* DELETE/DEFINE DATABASE DATASET
//*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DSN=STIMS220.DBG.UTIL(DBGAMBP),DISP=SHR
//*
//RECOVER EXEC PGM=DFSRRC00,
// PARM='UDR,DFSURDB0,DBGAMBP',REGION=4096K
//*
//* RECOVER DBGAMBP DATABASE DATASET
//*
//SEPLIB   DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DBGAMBP  DD DSN=STIMS220.DBG.DBGAMBP,DISP=SHR,AMP=(,BUFND=30')
//DFSUDUMP DD DSN=STIMS220.DBG.DBGAMBP.ICl,DISP=SHR,DCB=BUfNO=10
//DFSUCUM  DD DUMMY
//DFSULOG  DD DSN=STIMS220.DBG.B0lLOG2.G000lV00,DISP=SHR
//DFSVSAMP DD DSN=R82123R.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DDSTIMS22
S DBGAMBP DBGAMBP DFSUDUHP DBGAMBP RESTORE CONTROL
/*
```

Figure 82. DB Recovery without PRILOG Record

6.5 Recovery from SLDS Instead of a Bad RLDS

The Recovery Log Data Set (RLDS) contains all the log records needed for database recovery. When the RLDS is known to DBRC, the RLDS is used instead of the SLDS, by the GENJCL mechanism to create the JCL for DB recovery or change accumulation.

6.5.1 Write Error During an Archive

If a write error occurs on an RLDS during an archive, the simplest solution is to rerun the archive operation.

6.5.2 RLDS is Unavailable

When a read error occurs on an RLDS, or when the RLDS is unavailable and there is no secondary RLDS, SLDS must be used in its place.

In a DBRC Recovery or Share control environment, the RECON must be updated to reflect the error of the RLDS. The actions required are either:

- A copy of the SLDS using IEBGENER or DFSUARC0 to a new data set with the same name and VOLSER as the bad RLDS. In this case no update to the RECON is needed. The only problem could be space, because the SLDS is larger than the RLDS.
- An update of the PRILOG record in the RECON, with DSN and VOLSER information, to point at the SLDS, using the CHANGE.PRILOG command.

6.5.3 Update of the RECON

Figure 83 on page 88 shows a partial listing of the RECON with information about RLDS and SLDS before any change:

- PRILOG
  - DSN=STIMS220.RLDS.GOOO1VOO
  - DSN=STIMS220.RLDS.GOOO2VOO
- PRISLDS
  - DSN=STIMS220.SLDS.GOOO1VOO
  - DSN=STIMS220.SLDS.GOOO2VOO

Figure 84 on page 88 shows the CHANGE.PRILOG command where the parameter DSN points to the SLDS.

The correct SLDS to choose is shown in the listing of the RECON, where the start time of SLDS and RLDS is the same.

The parameter DSSTART is required if the PRILOG has multiple data set entries. When the SLDS is on a different volume, the parameters NEWVOL and OLDVOL must be specified.

```
Partial Listing of RECON

PRILOG
 START    = 88.321   17:15:53.9      STOP    = 88.321      17:27:33.3
 SSID=I220          #DSN=2           IMS

 DSN=STIMS220.SLDS.G0001V00                                UNIT=3400
 START    = 88.321   17:15:53.9      STOP    = 88.321     17:24:41.8
 FILE SEQ=0001    #VOLUMES=0001     -VOLSER-          -STOPTIME-
                                     STIMS4     88.321     17:24:41.8
 DSN=STIMS220.RLDS.G0002V00 UNIT=3400
 START    = 88.321   17:24:41.8      STOP    = 88.321      17:27:33.3
 FILE SEQ=0001    #VOLUMES=0001    -VOLSER-           -STOPTIME-
                                     STIMS5      88.321    17:27:33.3

LOGALl
 START    = 88.321   17:15:53.9
 DBDS ALLOC=4                        - DBD-   -DDN-   -ALLOC-
                                     DBGAMBP  DBGAMBP    1
                                     DBGAMBX  DBGAMBX    1
                                     DBGAMAP  DBGAMAP    1
                                     DBGAMBY  DBGAMBY    1

PRISLD
START    = 88.321  17:15:53.9      STOP    = 88.321    17:27:33.3
  SSID=I220         #DSN=2

  DSN=STIMS220.SLDS.G0001V00 UNIT=3400
  START   = 88.321   17:15:53.9      STOP    = 88.321      17:24:41.8
  FILE SEQ=0001   #VOLUMES=0001    -VOLSER-           -STOPTIME-
                                    STIMS4      88.321     17:24:41.8
  DSN=STIMS220.SLDS.G0002V00                                 UNIT=3400
  START   = 88.321   17:24:41.8      STOP    = 88.321      17:27:33.3
  FILE SEQ=0001   #VOLUMES=0001    -VOLSER- -          -STOPTIME-
                                    STIMS4      88.321      17:27:33.3
PRIOLD
SSID=I220             # DD ENTRIES=3

 DDNAME=DFSOLP00   DSN=STIMS220.0LP00
 START = 88.321     17:15:53.9       STOP     = 88.321    17:24:41.8
 STATUS=ARC COMPLT                              FEOV=YES    AVAIL
PRILOG TIME=88:321    17:15:53.9           ARCHIVE JOB NAME=ST220ARC
```

Figure 83. Logs Listed in RECON after CHANGE.PRILOG

```
   IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4   PAGE 0001
CHANG.PRILOG RLDS STARTIME(883211715539) DSN(STIMS22O.SLDS.G0001V00)-
  OLDVOL(STIMS5) NEWVOL(STIMS4) DSSTART(883211715539)
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I COMMAND COMPLETION TIME 88.322 12:23:59.4
   IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4 PAGE 0002
DSP02l1I COMMAND PROCESSING COMPLETE
DSP02l1I HIGHEST CONDITION CODE = 00
```

Figure 84. Change PRILOG Record Pointing at the SLDS

Figure 85 on page 89 shows the partial listing of RECON after execution of a CHANGE.PRILOG command.

Figure 86 on page 89 shows that after the change, a recovery of the DBDS DBGAMAP successfully uses the following SLDS as input:

- STIMS220.SLDS.G0015V00

```
Partial Listing of RECON
PRILOG
 START    = 88.321   17:15:53.9*   STOP    = 88.321   17:27:33.3
 SSID=I220           #DSN=2        IMS

 DSN=STIMS220.SLDS.G0001V00                               UNIT=3400
 START    = 88.321   17:15:53.9    STOP    = 88.321   17:24:41.8
 FILE SEQ=0001      #VOLUMES=0001  -VOLSER-         -STOPTIME-
                                    STIMS4    88.321  17:24:41.8
 DSN=STIMS220.RLDS.G0002V00                                UNIT=3400
 START    = 88.321   17:24:41.8    STOP    = 88.321   17:27:33.3
 FILE SEQ=0001      #VOLUMES=0001  -VOLSER-         -STOPTIME-
                                    STIMS5    88.321  17:27:33.3

LOGALL
 START    = 88.321   17:15:53.9*
 DBDS ALLOC=4                        -DBD-   -DDN-     -ALLOC-
                                     DBGAMBP DBGAMBP     1
                                     DBGAMBX DBGAMBX     1
                                     DBGAMAP DBGAMAP     1
                                     DBGAMBY DBGAMBY     1

PRISLD
  START   = 88.321   17:15:53.9    STOP    = 88.321   17:27:33.3
  SSID=I220         #DSN=2

  DSN=STIMS22O.SLDS.GOOOlVOO                                UNIT=3400
  START   = 88.321   17:15:53.9    STOP    = 88.321   17:24:41.8
  FILE SEQ=0001     #VOLUMES=0001  -VOLSER-         -STOPTIME-
                                    STIMS4    88.321  17:24:41.8

 DSN=STIMS22O.SLDS.GOOO2VOO                               UNIT=3400
  START    = 88.321  17:24:41.8    STOP    = 88.321   17:27:33.3
  FILE SEQ=0001     #VOLUMES=0001  -VOLSER-         -STOPTIME-
                                    STIMS4     88.321 17:27:33.3
88.322 12:27:29.5 LISTING OF RECON PAGE 0010

PRIOLD
 SSID=I220             # DD ENTRIES=3
 DDNAME=DFSOLP00    DSN=STIMS22O.0LPOO
 START = 88.321      17:15:53.9   STOP      = 88.321 17:24:41.8
 STATUS=ARC COMPLT                                 FEOV=YES AVAIL
 PRILOG TIME=88.321  17:15:53.9         ARCHIVE JOB NAME=ST22OARC
```

Figure 85. Logs Listed in RECON after CHANGE.PRILOG

```
//DEFINE EXEC PGM=IDCAMS,
// COND=(O,LT)
//*
//* DELETE/DEFINE DATABASE DATASET
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DSN=STIMS220.DBG.UTIL(DBGAMAP),DISP=SHR
//*
//RECOVER EXEC PGM=DFSRRC00,
//        PARM='UDR,DFSURDB0,DBGAMAP',REGION=1800K
//*
//* RECOVER DBGAMAP DATABASE DATASET
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DBGAHAP  DD USN=STIMS220.DBG.DBGAMAP,DISP=SHR,AMP=('BUFND=30')
//DFSUDUMP DD DSN=STIMS220.DBG.DBGAMAP.IC2,DISP=SHR,DCB=BUFNO=10
//DFSUCUH  DD DUMMY
//DFSULOG  DD DSN=STIMS220.SLDS.GOOOlVOO,DISP=SHR
//DFSVSAHP DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD *
S DBGAMAP DBGAMAP DFSUDUMP DBGAMAP RESTORE CONTROL
/*
```

Figure 86. DB Recovery Using an SLDS Instead of an RLDS

6.6 Unload Utility Authorization Details

The unload utility performs authorization processing and requests a level sharing of RD (read with integrity). This provides maximum availability of the database and allows parallel scheduling of the utility in read-only mode.

Part of the database status flags checking is bypassed, but the unload utility does not execute if the database is authorized for update to some other subsystem. One of the status flags not checked by the utility is the IC-needed flag.

Consider the following scenario in this case:

1. After a reorganization has been completed, the IC-needed flag is set to ON unless the HISAM Reload Utility was used and REUSE is not being used for the image copies.
2. GENJCL.RECOV does not use an IC data set with a time-stamp older than The reorganization time.
3. GENJCL.RECOV does not create any JCL if no valid IC data set is available.
4. During reorganization processing, the database can be unloaded but not reloaded, while the IC-needed flag is ON. This could cause the following scenario:
    1. The database is unloaded to a temporary disk file.
    2. The database is deleted and redefined with AMS.
    3. The reload step fails because it cannot obtain the authorization from DBRC because the IC-needed flag is set to ON.
    4. The temporary disk is deleted.
    5. There could be no valid IC data set to be used for a recovery.
    6. A time-stamp recovery could be required to produce a valid IC data set.

In the previous scenario an easy solution is to recover the database outside of the DBRC control.

To avoid this problem permanently, the database authorization can be checked for all the databases under reorganization, before proceeding with the reorganization itself. This check can be done in several ways:

- Execute a job step, prior to the unload step, with a dummy program that uses a PSB naming all the required databases with PROCOPT=GE.

  This forces DBRC to go through the authorization checking logic before the reorganization procedure. If the authorization checking of this program fails, with a user abend forced by DBRC, the following unload step is never executed and all the previous negative effects are eliminated.

- Use the authorization checking program of the DBICF program product

  When the authorization required through the DBICF program fails, the program writes a message to the MVS console and waits for a reply. The operator or other automatic solution (CLIST or program) can respond to the outstanding reply (retry the request or abort the job).

6.7 Removing a Subsystem Record

The removal of subsystem records is automatically done by DBRC in the following cases:

- The subsystem ends normally.
- The subsystem ends abnormally, but without any database allocated.
- The subsystem ends abnormally, but batch backout runs successfully.
- The subsystem ends abnormally, but dynamic backout was already evoked by the failed subsystem (BKO=Y on DFSRRC00 parm for DLI).

In some situations it is necessary to manually remove the subsystem record. Use the DBRC DELETE.SUBSYS command.

Before DBRC can permit the use of the DELETE command, all the allocations must also be removed. The easiest way to perform this task is to:

1. List the subsystem records with the command:

   ```
   LIST.SUBSYS ALL
   ```

2. If RECOVERY STARTED = NO then use the command:

   ```
   CHANGE.SUBSYS SSID(x) STARTRCV
   ```

3. Use the command:

   ```
   CHANGE.SUBSYS SSID(x) ENDRECOV
   ```

4. Use the command:

   ```
   DELETE.SUBSYS SSID(x)
   ```

Having deleted the subsystem record, DBRC assumes that all the databases have been recovered to a valid point.

This recovery of the databases must be performed manually.

Once the subsystem record has been deleted, DBRC allows normal authorizations to occur. Therefore, the recovery should be done before the subsystem record is deleted, or the PROHIBIT AUTHORIZATION flag should be turned on until the recovery is complete.

Note: Use the DELETE.SUBSYS command with great care. Understand fully the consequences to the overall environment before using it.

Chapter 7. User Function Guidelines

This chapter provides examples of GENJCL.USER commands for automating processes that otherwise would have to be done manually. The chapter's main goal is not to provide a complete set of examples, but rather to:

- Offer some hints on how the parameter substitution works
- Give some useful examples
- Stimulate ideas on further functions than can be automated

7.1 Hints for Coding User Functions

Coding most of the user functions is a straightforward task. Nonetheless, some of the following hints can be helpful:

- **Time qualifier:**
  - All time-stamps are validated for a valid date and time.
  - The time-stamp must be a 12 digit number.
  - If the time-stamp is found invalid, an INVALID KEYWORD in SELECT CONTROL statement message is issued (sometimes it can be misleading).
- **Batch Logs:**
  - When using the archive utility to archive batch log data sets:
    - Use the DD name of DFSSLDSP.
    - Use the SLDS control card.
  - When using the skeletal JCL commands to select batch logs:
    - Use the RLDS select statement.
    - The DSN keyword is %LOGDSN.
    - The DDN keyword is %LOGDDN.
- **Default members:**
  - Use default members to set user keys. This reduces the number of USERKEYS statements in the GENJCL commands and simplifies the task.
- Use of different skeletal members for different applications allows the coding of DSNs to suit each application.
- SELECT statements cannot be embedded in other SELECT statements.
- When using GROUP, the member is used for each member within the group and the keywords are substituted each time.

7.2 Generation of Batch Backout JCL for Failed Subsystems

The batch backout utility requires the PSB name and the DSN name of the log data set to be used as input to the backout. These are variables and cannot be recoded in the JCL. With the use of the GENJCL.USER function and the skeletal JCL, this process can be automated.

7.2.1 Generating the JCL

Here, a batch job (R82123R8), has abended, and batch backout is required. This job logs to a DASD SLDS (STIMS220.DBG.B01LOG.G0001V00). See Figure 87 for the RECON information after the batch job failure. See Figure 88 for the batch backout JCL generated using the DBRC command and the skeletal JCL.

```
Partial listing of RECON

PRILOG
 START   = 88.316   17:13:09.8*  STOP      = 88.316   17:14:00.4
 SSID=R82123RB     #DSN=1        IMS

 DSN=STIMS22X.DBG.B01LOG.G0001V00                           UNIT=3400
 START   = 88.316   17:13:09.8   STOP      = 88.316   17:14:00.4
 FILE SEQ=0001    #VOLUMES=0001  -VOLSER-          -STOPTIME-
                                  STIMS6       88.316   17:14:00.4

LOGALL
 START   = 88.316   17:13:09.5s
 DBDS ALLOC=5                        -DBD-    -DDN-    -ALLOC-
                                     DBGAMBP  DBGAMBP   1
                                     DBGAMBX  DBGAMBX   1
                                     DBGAMAP  DBGAMAP   1
                                     DBGAMAX  DBGAMAX   1
...
SSYS
 SSID=R82123R8  IRLMID=**NULL**     LOG START=88.316   17:13:09.8
 TYPE=BATCH     ABNORMAL TERM=ON    RECOVERY STARTED=NO   BACKUP=NO
 IRLM STATUS=NORMAL

 AUTHORIZED DATA BASES/AREAS=5        LEVEL=4

                                                      ENCODED
-DBD-        -AREA-     -LEVEL-   -ACCESS INTENT-    -STATE-
DBGAMBX      **NULL**      2         UPDATE             6
DBGAMBP      **NULL**      2         UPDATE             6
DBGAMAX      **NULL**      2         UPDATE             6
DBGAMAP      **NULL**      2         UPDATE             6
DBGAMBY      **NULL**      2         UPDATE             6
```

Figure 87. RECON Listing after Batch Job Failure (Partial)

```
DBRC Command:

 GENJCL.USER MEMBER(DBGBO) -
   USERKEYS((%USYSID,'R82123RB'),(%UPSB,'DBGBO1'),(%UTIME,'LAST'))

Member: DBGBO

//* I M S - D A T A B A S E          B A C K 0 U T
//STEPO1 EXEC PGM=DFSRRC00,REGION=4069K,
//        PARM=('DLI,DFSBB000,%UPSB,,0000,,0,,N,0,T,,N,Y,N,,Y')
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//DFSRESLB DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
%SELECT RLDS (%USYSID,%UTIME)
//IMSLOGR  DD DSN=%LOGDSN,DISP=SHR
%ENDSEL
//IEFRDER  DD DSN=STIMS220.DBG.B01LOG(+1),DISP=(NEW,CATLG,CATLG),
//            UNIT=3380,SPACE=(TRK,(100,100),RLSE),VOL=SER=STIMS6,
//            DCB=(STIMS22O.DCB,RECFM=VB,LRECL=23468,BLKSIZE=23472)
//IMS      DD DSN=STIMS22X.PSBLIB,DISP=SHR
//         DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DFSVSAMP DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD *
/*
//
```

Figure 88. DBRC Command and Skeletal JCL for Batch Backout

Note: Dynamic allocation is used to allocate all required databases.

The generated JCL is shown in Figure 89.

```
//*
//* I M S - D A T A B A S E   B A C K 0 U T
//*
//STEPO1 EXEC PGM=DFSRRC00,REGION=4096K,
//     PARM=(DLI,DFSBBO00,DBGB01,0000,,0,,N,0,T,,N,Y,N,,Y)
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//DFSRESLB DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//IMSLOGR  DD DSN=STIMS220.DBG.B01LOG.G0001V00,DISP=SHR
//IEFRDER  DD DSN=STIMS220.DBG.B01LOG(+1),DISP=(NEW,CATLG,CATLG),
//            UNIT=3380,SPACE=(TRK,(100,100),RLSE),VOL=SER=STIMS6,
//            DCB=(STIMS220.DCB,RECFM=VB,LRECL=23468,BLKSIZE=23472)
//IMS      DD DSN=STIMS22X.PSBLIB,DISP=SHR
//         DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DFSVSAMP DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD *
/*
```

Figure 89. Generated Batch Backout JCL

7.2.2 Adding Checkpoint Information

For entirely backing out a job, it is possible to use the checkpoint control card supplying the ID of the first checkpoint issued by the program. This returns the databases to the situation before the program was run. Figure 90 shows the required changes to the DBRC command and the skeletal JCL.

```
DBRC Command:

GENJCL.USER MEHBER(DBGBO) -
  USERKEYS((%USYSID,'R82123R8'),(%UPSB,'DBGB01'),(%UTIME,'LAST'),-
  (%UCPT,'CHKPT'),(%UCKPTID,('A7000172'))

Member DBGBO:

//*
//* I M S - D A T A B A S E   B A C K 0 U T
//*
//STEPO1 EXEC PGM=DFSRRC00,REGION=4069K,
//      PARM=(DLI,DFSBBO00,%UPSB,,0000,,0,,N,0,T,,N,Y,N,,Y)
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//DFSRESLB DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
%SELECT RLDS (%USYSID,%UTIME)
//IMSLOGR  DD DSN=%LOGDSN,DISP=SHR
%ENDSEL
//IEFRDER  DD DSN=STIMS220.DBG.B01LOG(+1),DISP=(NEW,CATLG,CATLGl,
//            UNIT=3380,SPACE=(TRK,(100,100),RLSE),VOL=SER=STIMS6,
//            DCB=(STIMS220.DCB,RECFM=VB,LRECL=23468,BLKSIZE=23472)
//IMS      DD DSN=STIMS22X.PSBLIB,DISP=SHR
//         DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DFSVSAMP DD DSN=STIMS220.DBG.UTIL(ISSVBUFl,DISP=SHR
//SYSIN    DD *
%UCPT %UCKPTID
/*
```

Figure 90. DBRC Command and Skeletal JCL for Batch Backout

Figure 91 shows the generated JCL with the added checkpoint information.

```
//*
//* I M S - D A T A B A S E   B A C K 0 U T
//*
//STEPO1 EXEC PGH=DFSRRC00,REGION=4096K,
//            PARM=(DLI,DFSBBO00,DBGB01,,000,,0,,N,O,T,,N,Y,N,,Y)
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//DFSRESLB DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//IMSLOGR  DD DSN=STIMS220.DBG.B0lLOG.G000lV00,DISP=SHR
//IEFRDER  DD DSN=STIMS220.DBG.B0lLOG(+1),DISP=(NEW,CATLG,CATLG),
//            UNIT=3380,SPACE=(TRK,(100,100),RLSE),VOL=SER=STIMS6,
//            DCB=(STIMS220.DCB,RECFM=VB,LRECL=23468,8LKSIZE=23472)
//IMS      DD DSN=STIMS22X.PSBLIB,DISP=SHR
//         DD DSN=STIMS22X.DBDLIB,DISP=SHR
//DFSVSAMP DD DSN=STIMS220.DBG.UTIL(ISSVBUF),DISP=SHR
//SYSIN    DD
CHKPT A7000172
/*
```

Figure 91. Generated Batch Backout JCL with Checkpoint Information

7.2.3 Additional Details

Consider the following suggestions when handling a complicated situation:
the log data set must be closed before being used as input to the batch backout utility. The batch backout utility DOES NOT verify the input.

- When more then one SLDS data set has been created before the batch backout execution (multiple restarts and corresponding failures), ensure that the correct log data set is used before submitting the batch backout.
- When a batch backout has been run successfully for one log and another run is required, the DBRC parameter on the EXEC card must be set to "C".
- When multiple log data sets must be used in input to a batch backout utility, they must be used in reverse order, those created last, used first. This can be done in single or multiple runs.

7.3 Selecting SLDS Data Sets

The following example deals with the selection of all the SLDS data sets created during the day for an online system. The data sets can be used, for instance, as input to jobs for:

- Performance reporting
- Tuning requirements
- Disaster recovery data set copies
- Archiving to tape

These jobs require as input only selected data sets. The manual effort to create the JCL can be significant.

In the following example, all the SLDS data sets for one day of IMS must be selected. They serve as input to a GPAR program to produce BMP usage statistics. Only one IMS session is run in the example, but the results would be the same with several sessions.

Figure 92 shows the DBRC command and the skeletal JCL used.

```
DBRC Command:

GENJCL.USER MEMBER(IMSTUNE) -

  USERKEYS((ZZTIME1)'883220000000'),(ZTIME2,'883222359599,),-
  (%ZSYSID,'I22X'))

Member IMSTUNE:
//*
//* I M S - PERFORMACE STATICS
//*
//A      EXEC PGM=GPAR
//STEPLIB  DD DISP=SHR,DSN=STIM22X.GPAR.LOAD
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//BMPPRINT DD SYSOUT=*,
//            DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSIN    DD DSN=STIMS22X.DBG.UTIL(IMSX22P2),DISP=SHR
//*
%SELECT SLDS(%ZSYSID,(FROM(%ZTIME1),TO(%ZTIME2))
//LOGIN DD DSN=SLDSDSN,DISP=SHR
%ENDSEL
//
```

Figure 92. Select SLDS Records (DBRC Command and Skeletal JCL)

A partial RECON listing in Figure 93 on page 98 shows the relevant PRILOG and SECLOG records.

```
Partial Listing of RECON
PRILOG
 START    = 88.322    15:07:42.7*   STOP   = 88.322     17:23:32.2
 SSID=I22X           #DSN=5         IMS

 DSN=STIMS22X.RLDS.G0050V00                              UNIT=3400
 START    = 88.322    15:07:42.7    STOP    = 88.322    17:17:25.1
 FILE SEQ=0001       #VOLUMES=0001 -VOLSER-       -STOPTIME-
                                    STIMS5    88.322    17:17:25.1

 DSN=STIMS22X.RLDS.G0051V00                             UNIT=3400
 START     = 88.322    17:17:25.1   STOP    = 88.322    17:20:06.8
 FILE SEQ=0001       #VOLUMES=0001 -VOLSER-        STOPTIME-
                                    STIMS5    88.322    17:20:06.8

 DSN=STIMS22X.RLDS.G0052V00                             UNIT=3400
 START    = 88.322    17:20:06.8     STOP    = 88.322   17:21:23.8
FILE SEQ=0001        #VOLUHES=0001  -VOLSER-      -STOPTIME-
                                     STIMS5    88.322   17:21:23.8

 DSN=STIMS22X.RLDS.G00G0053V00                              UNIT=3400
 START    = 88.322    17:21:23.8     STOP    = 88.322   17:22:14.5
 FILE SEQ=0001       #VOLVMES=0001  -VOLSER-        -STOPTIME-
                                     STIMS5    88.322   17:22:14.5

 DSN=STIMS22X.RLDS.G0054V00                             UNIT=3400
 START    = 88.322    17:22:14.5     STOP    = 88.322   17:23:32.2
 FIlE SEQ=0001       #VOLUMES=0001  -VOLSER-       -STOPTIME-
                                     STIMS5    88.322   17:23:32.2

LOGALL
 START    = 88.322    15:07:42.7
 DBDS ALLOC=4                              -DBD- -DDN-     -ALLOC-
                                          DBGAMBP DBGAMBP     1
                                          DBGAMBX DBGAMBX     1
                                          DBGAMAP DBGAMAP     5
                                          DBGAMBY DBGAMBY     1
PRISLD
 START     = 88.322   15:07:42.7     STOP    = 88.322   17:23:32.2
 SSID=I22X           #DSN=5

 DSN=STIMS22X.SLDS.G0050V00                             UNIT=3400
 START    = 88.322    15:07:42.7     STOP    = 88.322   17:17:25.1
 FILE SEQ=0001       #VOLUMES=0001  -VOLSER-        -STOPTIHE-
                                     STIMS6    88.322   17:17:25.1

 DSN=STIMS22X.SLDS.G0051V00                             UNIT=3400
 START    = 88.322    17:17:25.1     STOP    = 88.322   17:20:06.8
 FILE SEQ=0001       #VOLUMES=0001  -VOLSER-         -STOPTIME-
                                     STIMS6    88.322   17:20:06.8

 DSN=STIMS22X.SLDS.G0052V00                            UNIT=3400
 START = 88.322       17:20:06.8     STOP    = 88.322   17:21:23.8
 FILE SEQ=0001       #VOLUMES=0001  -VOLSER-         -STOPTIME-
                                     STIMS6    88.322   17:21:23.8

 DSN=STIMS22X.SLDS.G0053V00                            UNIT=3400
 START = 88.322       17:21:23.8     STOP    = 88.322   17:22:14.5
 FILE SEQ=0001       #VOLUMES=0001  -VOLSER-         -STOPTIME-
                                     STIMS6    88.322   17:22:14.5

 DSN=STIMS22X.SLDS.G0054V00                            UNIT=3400
 START = 88.322       17:22:14.5     STOP    = 88.322   17:23:32.2
 FILE SEQ=0001       #VOLUMES=0001  -VOLSER-         -STOPTIME-
                                     STIMS6    88.322   17:23:32.2
```

Figure 93. Select SLDS Records—RECON Listing

The resulting JCL in shown in Figure 94 on page 99.

```
//*
//* I M S -PERFORMANCE STATICS
//*
//A      EXEC PGM=GPAR
//STEPLIB  DD DISP=SHR,DSN=STIM22X.GPAR.LOAD
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//BMPPRINT DD SYSOUT=*,
//            DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSIN    DD DSN=STIMS22X.DBG.UTIL(IMSX22P2),DISP=SHR
//*
//LOGIN    DD DSN=STIMS22X.SLDS.G0050V00,DISP=SHR
//         DD DSN=STIMS22X.SLDS.G005lV00,DISP=SHR
//         DD DSN=STIMS22X.SLDS.G0052V00,DISP=SHR
//         DD DSN=STIMS22X.SLDS.G0053V00,DISP=SHR
//         DD DSN=STIMS22X.SLDS.G0054V00,DISP=SHR
//*
```

Figure 94. Select SLDS Records (Generated JCL)

7.4 Selecting IC Data Sets

The following example is related to the task of selecting the first created IC data sets to be copied on tapes for an off-site storage. In the example, all the IC data sets for one DBDSGRP are selected and copied to tapes with IEBGENER.

Figure 95 on page 99 shows the DBRC command and skeletal JCL used. A partial RECON listing in Figure 96 on page 100 shows the relevant DBDS and IC records.The resulting JCL is shown in Figure 97 on page 101.

```
DBRC Comnand:

GENJCL.USER MEMBER(ICCOPY) GROUP(DBGGRP1) ONEJOB

Member ICCOPY:

//*
//C%DBNAME EXEC PGM=IEBGENER,REGION=4096K
//*
//* COPY IC DATASETS TO TAPE
//*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
%SELECt IC ((%DBNAME,%DDNAME),LAST)
//SYSUT1   DD DSN=%ICDSN,DISP=SHR
%ENDSEL
//%DDNAME  DD DSN=STIMS22X.%DBNAME.VITAL,DISP=SHR
//SYSUT2   DD DSN=ICDSN.VITAL,DISP=(,CATLG),
//            UNIT=TAPE,VOL=REF=%DDNAME
```

Figure 95. Select IC Data Sets (DBRC Command and Skeletal JCL)

```
**** HISTORY OF STIMS22X.DBG.DBGAMAP    DBD:DBGAMAP DDN:DBGAMAP

IC1-> 88319      19:50:06.4 DSN       : STIMS22X.DBG.DBGAMAP.IC1
      14.11.88 MONDAY       FILE-SEQ# : 1
                            NBR RCDS  : 155
                            VOLSER    : STIMS5


IC1-> 88323      12:01:05.0 DSN       : STIMS22X.DBG.DBGAMAP.IC3
      18.11.88 FRIDAY       FILE-SEQN : 1
                            NBR RCDS  : 155
                            VOLSER    : STIMS5

**** HISTORY Of STIHS22X.DBG.DBGAMAX    DBD:DBGAMAX DDN:DBGAMAX

IC1->88319       19:50:21.2 DSN : STIMS22X.DBG.DBGAMAX.IC1
     14.11.88  MONDAY       FILE-SEQ# : 1
                            NBR RCDS  : 2
                            VOLSER    : STIMS5

IC1->88323       12:01:19.4 DSN : STIMS22X.DBG.DBGAMAX.IC3
     18.11.88  FRIDAY       FILE-SEQN : 1
                            NBR OF RECORDS :      24
                            VOLSER(S) : STIMS5

**** HISTORY OF STIHS22X.DBG.DBGAMBP     DBD:DBGAMBP DDN:DBGAMBP

IC1-> 88319      19:50:35.7 DSN : STIMS22X.DBG.DBGAMBP.IC1
      14.11.88  MONDAY      FILE-SEQ# : 1
                            NBR RCDS  : 90
                            VOLSER : STIMS5

IC1-> 88323      12:01:33.6 DSN : STIMS22X.DBG.DBGAHBP.iC3
      18.11.88  FRIDAY      FILE-SEQ#: 1
                            NBR RCDS : 90
                            VOLSER   : STIMS5

**** HISTORY OF STIMS22X.DBG.DBGAMBX     DBD:DBGAMBX DDN:DBGAMBX

IC1-> 88319      19:50:50.2 DSN : STIMS22X.DBG.DBGAMBX.IC1
      14.11.88  MONDAY      FILE-SEQ# : 1
                            NBR RCDS  : 2
                            VOLSER    : STIMS5

IC1-> 88323      12:01:47.8 DSN : STIMS22X.DBG.DBGAMBX.IC3
      18.11.88  FRIDAY      FILE-SEQ# : 1
                            NBR RCDS  : 3
                            VOLSER : STIMS5


**** HISTORY OF STIMS22X.DBG.DBGAMBY     DBD:DBGAMBY DDN:DBGAMBY

IC1-> 88319      19:51:05.1 DSN : STIMS22X.DBG.DBGAMBY.IC2
      14.11.88  MONDAY      FILE-SEQ# : 1
                            NBR RCDS  : 1
                            VOLSER    : STIMS5


IC1-> 88323      12:02:03.7 DSN: STIMS22X.DBG.DBGAMBY.IC3
      18.11.88  FRIDAY      FILE-SEQ# : 1
                            NBR RCDS  : 4
                            VOLSER    : STIMS5
```

Figure 96. Select IC Data Sets (RECON Listing generated with DBICF)

```
//CDBGAMAP EXEC PGM=IEBGENER,REGION=4096K
//*
//* COPY IC DATASETS TO TAPE
//*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=STIMS22X.DBG.DBGAMAP.IC3,DISP=SHR
//DBGAMAP  DD DSN=STIMS22X.DBGAMAP.VITAL,DISP=SHR
//SYSUT2   DD DSN=STIMS22X.DBG.DBGAMAP.IC3.VITAL,DISP=(,CATLG),
//            UNIT=TAPE,VOL=REF=DBGAMAP
//*
//CDBGAMAX EXEC PGM=IEBGENER,REGION=4096K
//*
//* COPY IC DATASETS TO TAPE
//*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=STIMS22X.DBG.DBGAMAX.IC3 DISP=SHR
//DBGAMAX  DD DSN=STIMS22X.DBGAMAX.VITAL,DISP=SHR
//SYSUT2   DD DSN=STIMS22X.DBG.DBGAMAX.IC3.VITAL,DISP=(,CATLG),
//            UNIT=TAPE,VOL=REF=DBGAMAX
//*
//CDBGAMBP EXEC PGM=IEBGENER,REGION=4096K
//*
//* COPY IC DATASETS TO TAPE
//*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=STIMS22X.DBG.DBGAMBP.IC3,DISP=SHR
//DBGAMBP  DD DSN=STIMS22X.DBGAMBP.VITAL;DISP=SHR
//SYSUT2   DD DSN=STIMS22X.DBG.DBGAMBP.IC3.VITAL,DISP=(,CATLG),
//            UNIT=TAPE,VOL=REF=DBGAMBP
//*
//CDBGAMBX EXEC PGM=IEBGENER,REGION=4096K
//*
//* COPY IC DATASETS TO TAPE
//*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=STIMS22X.DBG.DBGAMBX.IC3,DISP=SHR
//DBGAMBX  DD DSN=STIMS22X.DBGAMBX.VITAL,DISP=SHR
//SYSUT2   DD DSN=STIMS22X.DBG.DBGAMBX.IC3.VITAL,DISP=(,CATLG),
//            UNIT=TAPE,VOL=REf=DBGAMBX
//*
//CDBGAMBY EXEC PGM=IEBGENER,REGION=4096K
//*
//* COPY IC DATASETS TO TAPE
//*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=STIMS22X.DBG.DBGAMBY.IC3,DISP=SHR
//DBGAMBY  DD DSN=STIMS22X.DBGAMBY.VITAL,DISP=SHR
//SYSUT2   DD DSN=STIMS22X.DBG.DBGAMBY.IC3.VITAL,DISP=(,CATLG),
//            UNIT=TAPE,VOL=REF=DBGAMBY
```

Figure 97. Select IC Data Sets (Generated JCL)

Note: The VOL=REF parameter points to a cataloged data set to mount the correct volume serial for each IC data set.

7.5 Selecting PRILOG Allocation Records

The following example shows how to select the ALLOC records to keep track of the database updates.

In the example, all the PRILOG allocation records for one day are selected (88322).

Figure 98 shows a partial RECON listing with the relevant DBDS and IC records.

```
PRILOG
 START    = 88.322    15:07:42.7     STOP    = 88.322   17:23:32.2
 SSID=I22X           #DSN=5          IMS

 DSN=STIMS22X.RLDS.G0050V00                                UNIT=3400
 START    = 88.322    15:07:42.7     STOP    = 88.322   17:17:25.1
 FILE SEQ=0001       #VOlUMES=0001  -VOLSER-        -STOPTIME-
                                     STIMS5    88.322   17:17:25.1

 DSN=STIMS22X.RLDS.G005lV00                                UNIT=3400
 START    = 88.322    17:17:25.1     STOP    = 88.322   17:20:06.8
  FILE SEQ=0001       #VOLUMES=0001 -VOLSER-        -STOPTIME-
                                     STIMS5    88.322   17:20:06.8

 DSN=STIMS22X.RLOS.G0052V00                                UNIT=3400
 START    = 88.322    17:20:06.8     STOP    = 88.322   17:21:23.8
 FILE SEQ=0001       #VOLUMES=0001  -VOLSER-        -STOPTIME-
                                 STIMS5    88.322   17:21:23.8

 DSN=STIMS22X.RLDS.G0053V00                                UNIT=3400
 START    = 88.322   17:21:23.8      STOP    = 88.322   17:22:14.5
 FILE SEQ=0001      #VOlUMES=0001   -VOLSER-        -STOPTIME-
                                     STIMS5    88.322   17:22:14.5

 DSN=STIMS22X.RLDS.G0054V00                                UNIT=3400
 START    = 88.322   17:22:14.5      STOP    = 88.322   17:23:32.2
 FILE SEQ=0001      #VOLUMES=0001   -VOLSER-        -STOPTIME-
                                     STIMS5    88.322   17:23:32.2

lOGALL
 START    = 88.322   15:07:42.7*
 DBDS ALLOC=4                            -DBD-      -DDN- -ALLOC-
                                         DBGAMBP    DBGAMBP  1
                                         DBGAMBX    DBGAMBX  1
                                         DBGAMAP    DBGAMAP  5
                                         DBGAMBY    DBGAMBY  1
PRISLD
 START    = 88.322   15:07:42.7*       STOP    88.322   17:23:32.2
 SSID=I22X          #DSN=5
 DSN=STIMS22X.SLDS.G005OV00                                UNIT=3400
 START    = 88.322   15:07:42.7      STOP    = 88.322   17:17:25.1
 FILE SEQ=0001      #VOLUMES=0001   -VOLSER-        -STOPTIME-
                                     STIMS6    88.322   17:17:25.1

 DSN=STIMS22X.SLDS.G005lV00                                UNIT=3400
 START    = 88.322   17:17:25.1      STOP    = 88.322   17:20:06.8
 FILE SEQ=0001      #VOLUMES=0001   -VOLSER-        -STOPTIME-
                                     STIMS6    88.322   17:20:06.8

 DSN=STIMS22X.SLDS.G0052V00                                UNIT=3400
 START    = 88.322   17:20:06.8       STOP   = 88.322   17:21:23.8
 FILE SEQ=0001      #VOLUMES=0001   -VOLSER-        -STOPTIME-
                                     STIMS6    88.322   17:21:23.8

 DSN=STIMS22X.SLDS.G0053V00                                UNIT=3400
 START    = 88.322   17:21:23.8      STOP    = 88.322   17:22:14.5
 FILE SEQ=0001      #VOLUHES=0001   -VOLSER-        -STOPTIME-
                                     STIMS6    88.322   17:22:14.5

DSN=STIMS22X.SLDS.G0054V00                                 UNIT=3400
 START    = 88.322   17:22:14.5      STOP    = 88.322   17:23:32.2
 FILE SEQ=0001      #VOLUMES=0001   -VOLSER-        -STOPTIME-
                                     STIMS6    88.322   17:23:32.2
```

Figure 98. Select SLDS Records - RECON Listing

Figure 99 shows the DBRC command and the skeletal JCL used.

```
DBRC Command:

GENJCL.USER MEMBER(PRIALLOC) NOJOB-
 USERKEYS((%ZTIME1,'883220000000'1,(%ZTIME2,'883222359599'))

Member PRIALLOC:

//*
//*      REPORT ON ALL DBDS UPDATED BY ONLINE SUBSYSTEN SESSION
//*
REPORT #F0000     DATA BASE      START TIME       STOPTIME
%SELECT ALLOC(PRILOG,(FROM(%ZTIME1),TO(%ZTIME2)))
                   %DBNAME    %ALLTIME      %DALTIME

%ENDSEL
```

Figure 99. Selecting PRILOG Records (DBRC Command and Skeletal JCL)

The resulting report is shown in Figure 100.

```
//*
//*      REPORT ON ALL DBDS UPDATED BY ONLINE SUBSYSTEM SESSION
//*
REPORT #F0000   DATA BASE     START TINE      STOPTIME
                 DBGAMBP    883221715316       000000000000
                 DBGAMBX    883221715323       000000000000
                 DBGAMAP    883221715333       883221717238
                 DBGAMAP    883221717519       883221720055
                 DBGAMAP    883221720340       883221721225
                 DBGAMAP    883221721536       883221722133
                 DBGAMAP    883221722472       000000000000
                 DBGAMBY    883221715411       000000000000
```

Figure 100. Select PRILOG Records (Generated Output)

7.6 Default Members

When using the DBRC user function, many parameters must be coded to set all the required keywords to the correct values.

The use of default members can be beneficial in reducing the amount of coding required each time the command is used. In the following example, the JCL to archive and then delete the RLDS data sets for a database must be generated:

- The skeletal JCL has been set up to use a date range supplied with the USERKEYS parameter.
- When the USERKEYS parameter is added to the GENJCL.USER command with the other required parameters, a very long and complicated command is created.
- Since the start time (0000000) and the stop time (2359599) are always the same, they need not to be re-entered on each command.
- The default member in this example has set the start time to '0000000' and the stop time to '2359595'.

- The DBDNAME and DDNAME keywords must also be entered.
- If a keyword is set, both in the default member and on the GENJCL.USER command, the value from the GENJCL command overrides the default values.

Figure 101 shows the DBRC command with and without the use of the default member.

Command with all key words coded:

```
GENJCL.USER MEMBER(SLDSARCH) DBD(DBGAMAP) DDN(DBGAMAP)-
  USERKEYS(%ZD1,'88328'),(%ZT1,'0000000'),(%ZD2,'88328'),-
  (%ZT2,'2359599'))
```

Command using default member SLDSBO1:

```
GENJCL.USER MEMBER(SLDSARCH) DEFAULTS(SLDSBO1) -
  USERKEYS(%ZD1,'88328'),(%ZD2,'88328'))
```

Command using default member SLDSBO1 and overriding some keywords:

```
GENJCL.USER MEMBER(SLDSARCH) DEFAULTS(SLDSB01) GROUP(DBGGRP1)-
        USERKEYS(%ZD1,'88328'),(%ZD2,'88328'))
```

Figure 101. Default Member—DBRC Command

The third command in Figure 101 overrides the DBNAME and DDNAME keywords in the default member and replaces them with the DBNAME and DBDDN of each database data set in the group. Figure 102 shows the skeletal JCL members.

Member SLDSARCH:

```
//AR00001 EXEC PGM=DFSUARCO,PARM='DBRC=YES',REGION=4096K
//*
//* JCL FOR ARCHIVE UTILITY
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
%SELECT RLDS((%DBNAME,%DDNAME1,%FROM(%ZDATE1%ZT11,TO(%ZDATE2%ZT2)),1)
//DFSSLDSP DD DSN=%LOGDSN,DISP=(OLD,DELETE)
%ENDSEL
//DFSSLOGP DD DSN=%LOGDSN.ARCHIVED,
//            DISP=(NEW,CATLG),
//            DCB=(LRECL=22524,8LKSIZE=22528,RECFM=VB,BUFNO=30),
//            UNIT=TAPE
//SYSIN    DD *
SLDS FEOV(08000)
/*
//
```

Member SLDSBO1:

```
%DBNAME = 'DBGAMAP'
%DDNAME = 'DBGAMAP'
%ZT1    = '0000000'
%ZT2    = '2359599'
```

Figure 102. Default Members—Skeletal JCL Members

Figure 103 shows the generated JCL.

```
//AR00001 EXEC PGM=DFSUARC0,PARM='DBRC=YES',REGION=4096K
//*
//* JCL FOR ARCHIVE UTILITY
//*
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//DFSSLDSP DD DSN=STIMS22O.RLDS.G0007V00,DISP=(OLD,DELETE)
//DFSSLOGP DD DSN=STIHS22O.RLDS.G0007V00.ARCHIVED,
//            DISP=(NEW,CATLG),
//            DCB=(LRECL=22524,8LKSIZE=22528,RECFM=VB,BUFNO=30),
//            UNIT=TAPE
//SYSIN    DD *
SLDS FEOV(08000)
/
//
```

Figure 103. Default Members—Generated JCL

Chapter 8. Example of a DBRC Front-End

The Data Base Integrity Control Facility (DBICF) is designed to protect databases against most types of operational error in an IMS online, CICS/OS with DL/I Data Bases, and IMS batch environment.

The main benefits of DBICF are to automate, simplify, and accelerate the DB repair-oriented actions by providing the user with a set of interrelated programs and procedures.

Note: DBICF may not be available in all countries. It is used in this book only as an example. In their previous experience with customers, the authors of the book had excellent feed-back using this product, but any other functionally equivalent program may be used instead.

8.1 General Product Information

The “home” panel of DBICF is shown in Figure 104.

```
-------------------ENVIRONMENT SELECTION MENU ------------------
SELECT OPTION ==>

ENVIRONMENT SELECTION MENU                                  USERID          USER
                                                            DATE            DATE


              DDD     BBBB       I       CC   FFFFF         TIME            TIME
              D  D    B   B      I     C      F
              D   D   BBBB       I    C       FFF
              D  D    B   B      I     C      F
              DDD     BBBB       I       CC   F

              P - PRODUCTION IMS
              T - TEST IMS
              X - EXIT

  - FOR SPECIAL RECON, ENTER SECOND QUALIFIER OF RECON =-
                             (E.G. IMSVS.QUAL.RECONOX)
     OR FULLY QUALIFIED RECON DSNAME:
      - RECON1 ==>
      - RECON2 ==>
      - RECON3 ==>
      -RECON-LOG ==>
  - DBICF LOAD LIBRARY (DSNAME) ==>

PRESS END KEY TO TERMINATE DBICF
```

Figure 104. DBICF Environment Selection Panel

8.1.1 DBICF Organization

DBICF consists of two parts:

- **ONLINE**

  When running under TSO/ISPF, DBICF is a direct interface between the user and DBRC. Over 200 panels and 100 CLIST procedures help the user to enter correct DBRC commands. This part of DBICF can be easily incorporated into an ISPF work panel for use by various users:

  - IMS MTO operators (for monitoring the systems)
  - DBA administrators (for recovery processing)
  - Programming staff (for testing system operations)

- **BATCH**

  DBICF contains other program functions that can be started as batch jobs, with procedures, or if applicable, under TSO foreground. All these functions work independently from each other, and a user can choose the functions that best fit a specific requirement.

8.1.2 DBICF Access to RECON

DBICF accesses the RECON in two ways:

- **READ**

  The RECON data sets are accessed directly through standard VSAM macros (ACB, GET, POINT, and so on).

- **UPDATE**

  DBICF never updates a RECON directly. To perform the updates, the following tasks are generated:

  - DBRC commands are generated.
  - DBRC is called.
  - DBRC executes the generated commands and updates the RECON.

Note: All DBRC commands that update the RECON may optionally be recorded on a DBICF log (VSAM KSDS). This log provides a useful audit trail for actions with DBRC.

8.1.3 DBICF Primary Menu Panel

Figure 105 shows the primary menu panel with the available user functions.

```
---------------------- DBRC- CMD Primary Menu ----------------------------------
SELECT OPTION ==>

ENVIRONMENT SELECTION MENU                                 USERID        USER
                                                           DATE          DATE


                                                           TIME          TIME
SELECT OPTION ===>

  1  CHANGE    - CHANGE RECON INFORMATION
  2  COMPARE   - RECORD BY RECORD COMPARISON (DBRC REL.2)
  3  DELETE    - REMOVE RECON RECORDS
  4  GENJCL    - GENERATE JOB CONTORL
  5  INIT      - CREATE RECON RECORDS
  6  LIST      - LIST RECON INFORMATION
  7  NOTIFY    - ADD RECON INFORMATION
  8  DBICF     - DBICF-FUNCTIONS
  9  DBRC      - ENTER DBRC-COMMANDS IN DBRC SYNTAX
  A  ISPF      - ISPF PRIMARY MENU

PRESS END KEY TO RETURN TO THE PREVIOUS MENU
________________________________________________________________________________
```

Figure 105. DBICF Primary Menu Panel

8.2 DBICF as a DBRC Front-End

DBICF allows the user to automatically take preventive and corrective actions to enhance the integrity of the databases. It also enables the user to better organize and supervise the daily IMS online, CICS/OS, and IMS batch operations.

8.2.1 GENJCL Functions

Figure 106 shows the DBICF DBRC-Command panel for GENJCL.

```
--------------------------------  DBRC-CMD GENJCL ----------------------------
SELECT OPTION ===>
                                                              USERID         USER
0   GENJC1.ARCHIVE - GENERATE JOB FOR LOG ARCHIVE             DATE           DATE
1   GENJC1.CA      - GENERATE CHANGE ACCUMULATION JCL
2   GENJCL.CLOSE   - GENERATE JOB fOR LOG RECOVERY            TIME
3   GENJCL.DSLOG   - GENERATE DSLOG JCL
4   GENJCL.IC      - GENERATE IMAGE COPY JCL
5   GENJCL.0IC     - GENERATE ONLINE IMAGE COPY JCL
6   GENJCL.RECOV   - GENERATE RECOVERY JCL
7   GENJCL.USER    - GENERATE USER OUTPUT (DBRC REL.4)


PRESS END KEY TO RETURN TO PREVIOUS MENU
______________________________________________________________
```

Figure 106. DBICF DBRC-CMD Panel for GENJCL

DBRC provides partitioned data set (PDS) members containing skeletal JCL statements. These PDS members are called skeletal JCL execution members. DBRC installation procedures are designed to place these skeletal JCL execution members in an IMS distribution library (IMSVS.JCLLIB). This library is also used by DBICF.

8.2.1.1 GENJCL.IC

Figure 107 and Figure 108 show examples of the panels for GENJCL.IC.

DBICF provides a set of batch functions which can generate the jobs necessary for:

- The image copy of:
  - All databases
  - Individual databases
  - Specific databases; for example:
    - The DBs that have been updated since the last image copy
    - The DBs that must be saved because they have never been saved (after a registration to DBRC)
    - The DBs with the latest image copy flagged invalid.
- The checking of the successful execution of the generated image copy jobs (IC checklist).

```
--------------------------- DBRC-CMD GENJCL.IC     -----------------------
  ENTER REQUIRED PARAMETERS :                                USERID -  USER
                                                             DATE   -  DATE

                                                             TIME   -  TIME
     WHICH DBD                           DBD-NAHE ====>              REQ.(R3)
     WHICH DDN (IMS)             OR      WHICH AREA-NAME (DEDB)     REQ.(R3)
     DDN =-==>              <-------------> AREA-NAME
     HOW HANY COPIES                     ENTER 1 OR 2 ==>
     SPECIFY THE VOLUMES TO CONTAIN THE IC DATA SET (UP TO 9)
     IN FORM: VOL1,VO12,....VOL9
  ==>
     SPECIFY THE VOLUMES TO CONTAIN THE DUPLICATE IC DATA SET (UP TO 9)
     IN FORM: VOL1,VOL2,....VOL9
  ==>

     IF NO JOB CARD IS TO BE PRODUCED: ENTER 'NOJOB' ==>
    IF YOU WANT TO SPECIFY ADDITIONAL PARAMETERS SUCH AS:
     DEFAULTS  JCLOUT  JCLPDS  JOB  LIST|NOLIST  MEMBER  NODEFLT  GROUP
     MULTIJOB|ONEJOB  UNIT1  UNIT2       ------           ENTER 'Y' ==>
     --------

PRESS END KEY TO RETURN TO THE PREVIOUS MENU
____________________________________________________________________________
```

Figure 107. GENJCL.IC Panel (Part 1)

```
------------------  DBRC-CMD GENJCL.IC    (CONTINTUED)--------------------------
                                                          USERID         USER
                                                          DATE           DATE
  LIST OF GENERATED JCL DESIRED ? (Y/N) ==>
                                                         TIME      -    TIME
  *** FOR IMS/VS V.2 (DBRC R4) ONLY:
  SPECIAL DEFAULT MEMBERS TO BE USED ?  ENTER NAME(S)  (FORM: MBR1,MBR2..MBR6)
  ==>
  IS THE DEFAULT MEMBER TO BE USED ?    ENTER 'Y' OR 'N' ==>
  SPECIAL 'JCLOUT' DD-STATEMENT ?       ENTER DD-NAME ==> ( SEE
  SPECIAL 'JCLPDS' DD-STATEMENT ?       ENTER DD-NAME ==> NOTE
  SPECIAL EXECUTION JCL SKELETAL ?      ENTER MEMBER ==>
  SPECIAL JOB-CARD SKELETAL ?           ENTER MEMBER ==>
  GENJCL FOR AN ENTIRE DBDS-GROUP ?     ENT£R GRPNAME ==>
  MULTIPLE JOBS DESIRED ?               ENTER 'Y' OR 'N' ==>
  SPECIAL UNIT-TYPE FOR COPY1 ?         ENTER UNIT1-TYPE ==>
  SPECIAL UNIT-TYPE FOR COPY2 ?         ENTER UNIT2-TYPE ==>


PRESS END KEY TO RETURN TO THE FIRST PART OF THE MENU

NOTE: THE JCL STATEMENT WITH THIS DD-NAME MUST BE PRESENT IN THE PROC. 'DBRC'
_____________________________________________________________________________
```

Figure 108. GENJCL.IC Panel (Part 2)

The user can also specify additional parameters such as:

- Defaults
- JCLOUT
- JCLPDS
- Job

The second part of the GENJCL.IC menu panel shown in Figure 108 is used for these choices.

8.2.2 NOTIFY Functions

Figure 109 shows the DBICF panel for the NOTIFY commands.

```
-------------------------------  DBRC-CMD   NOTIFY ----------------------------
SELECT OPTION ===>
                                                            USERID    -    USER
   0  NOTIFY AllOC   - ADD ALLOC/DEALLOC RECORD            DATE      -    DATE
   1  NOTIFY.CA      - ADD CA EXECUTION RECORD
   2  NOTIFY.IC      - ADD IMAGE COPY RECORD               TIME      -    TIME
   3  NOTIFY.MN      - ADD MN RECORD (DBRC R2 AND R3)
   4  NOTIFY.PRILOG  - ADD PRILOG RECORD (DBRC R2 / IMS-BATCH / CICS)
   5  NOTIFY.PRILOG  - ADD PRILOG RECORD (OLDS)
   6  NOTIFY.PRILOG  - ADD PRILOG RECORD (RLDS)
   7  NOTIFY.PRILOG  - ADD PRILOG RECORD (SLDS)
   8  NOTIFY.RECOV   - ADD RECOVERY RECORD
   9  NOTIFY.REORG   - ADD REORG RECORD
   A  NOTIfY.SEClOG  - ADD SECLOG RECORD (DBRC R2 / IMS-BATCH / CICS)
   B  NOTIFY.SECLOG  - ADD SECLOG RECORD (OLDS)
   C  NOTIFY.SECLOG  - ADD SECLOG RECORD (RLDS)
   D  NOTIFY.SECLOG  - ADD SECLOG RECORD (SLDS)
   E  NOTIFY.SUBSYS  - ADD SUBSYSTEM RECORD
   F  NOTIFY.UIC     - ADD USER IMAGE COPY RECORD

PRESS END KEY TO RETURN TO PREVIOUS MENU
______________________________________________________________
```

Figure 109. DBICF DBRC-CMD Panel for NOTIFY

If the standard database image copy or database recovery utilities are not used, the NOTIFY.UIC and NOTIFY.RECOV commands must be used to update the RECON with the needed information.

The NOTIFY command can also be used to update the RECON with information about other activities, such as change accumulations, log data sets, and so on.

8.3 DBICF Online Functions

Figure 110 and Figure 111 show the first and second parts of the DBICF functions panel.

```
SELECT OPTION ===>

                                                                    USERID -  USER
                                                                    DATE   -  DATE
  0  ICFDBS          - RECOVERY ORIENTED DB OVERVIEW
  1  DBHIST          - DATABASE HISTORY TIME - TIME
  2  OVERVIEW        - DBH ORIENTED DB OVERVIEW
  3  OIC-PSBGEN      - PSB-GENERATOR FOR OIC-PSB
  4  RUPDLOG-PRINT   - PRINT OF THE RECON UPDATES LOG-DATASET
  5  DBDCOMP         - COMPARE DBDLIB WITH RECON
  6  USICVOLS        - LIST OF VOLSER'S CONTAINING VALID IMAGE COPY DATA
  7  SSSTATUS        - LIST OF THE 'ACTIVE' SUBSYSTEMS
  8  CR.CAGRP        - CHANGE ACCUMULATION GROUP CREATION
  9  UP.CAGRP        - CHANGE ACCUHULATION ENVIRONMENT UPDATE
  A  RUPDLOG-MAINT.  - MAINTENANCE OF THE RECON UPDATES LOG-DATASET
  SECOND PART:
  CON                - B ADS-OVERVIEN       - C SESSION-REPORT
   OR                - D CR.DBGRP           - E UP.DBGRP
  B-H                - F MDACHK             - G NHICHGRP
                     - H DATABASE CHECK

PRESS END KEY TO RETURN TO THE PRIMARY MENU
__________________________________________________________________________________________
```

Figure 110. DBICF Functions Panel (Part 1)

```
---------------------DBICF-FUNCTIONS MENU    SECOND PART  --------------------
SELECT OPTION ===>

                                                                USERID -  USER
                                                                DATE   -  DATE

   B ADS-OVERVIEW      - DEDB/ADS OVERVIEW (R3/R4 ONLY)
   C SESSION-REPORT    - LOGGING AND DB ALLOCATION REPORT (R3/R4 ONLY)
   D CR.DBGRP          - DATABASE DATASET GROUP CREATION (R4 ONLY)
   E UP.DBGRP          - DATABASE DATASET GROUP UPDATE (R4 ONLY)
   F MDACHK            - CONSISTENCY CHECK BETWEEN MDA-MEMBERS AND RECON
   G HHICHGRP          - TO WHIGH CA- OR DB-GROUP(S1 BELONGS A SPECIFIG DB
   H DB CHECK          - CHECK WHICH DB'S ARE RECOV-/IC-/BACKOUT-NEEDED, ETC..



PRESS END KEY TO RETURN TO THE FIRST PART
```

Figure 111. DBICF Functions Panel (Part 2)

8.3.1 Report Functions

Reports obtained through DBICF are:

- Recovery oriented DB overview
- Database history
- DBM oriented DB overview
- List of VOLSERs containing valid image copy data
- List of the ACTIVE subsystems
- DEDB/ADS overview
- Logging and DB allocation report

8.3.1.1 Recovery Oriented DB Overview

Figure 112 shows an example of a recovery oriented database overview.

```
MONDAY          ,21.11.88(88326)D B I C F  D A T A B A S E S              0 V E R V I E N  PAGE 1
TIME            17/31/25
RECON1-DSN : STIMS22O.RECON1
RECON2-DSN : STIMS22O.RECON2
DBD-NAME DDN/AREA DBORG ACCESS OLDEST IC NEHEST IC-- NBR.0F  LAST UPDATE DSL CAGRP MEDIUM SHRLVL RECOVJCL
                                 DD.MM.YY   DD.HM.YY RECORDS  DD.HM.YY                   MEMBER
STIMS22O.DBG.DBGAMAP                        GENMAX= 3   ALLOCATED   ON VOLUME(S1 : STIMS4
DBGAMAP DBGAMAP HIDAM VSAM   88313/1829379 88319/1842197 155 88321/1719046 NO DBGCA  N.DEF.  2 DBGRECOV
                                 08.11.88INV 14.11.88 16.11.88
                                 TUESDAY    MONDAY      WEDNESDAY

STIMS22O.DBG.DBGAMAX                         GENMAX= 3   ALLOCATED   ON VOLUME(S) : STIMS6
DBGAMAX   DBGAMAX   INDEX VSAM   88313/182954988319/1842354  24 NONENO DBGCA   N.DEF.   2 DBGRECOV
                                 08.11.88   14.11.88
                                 TUESDAY    MONDAY

STIMS22O.DBG.DBGAMBP                          GENMAX= 3   ALLOCATED          ON VOLUME(S) : STIMS6
DBGAMBP   DBGAMBP   HIDAM VSAH   88313/183011488319/1842508  9088321/1719027 NO DBGCA  N.DEF.  2 DBGRECOV
                                 08.11.88   14.11.88    16.11.88
                                 TUESDAY    MONDAY      WEDNESDAY
STIMS22O.DBG.DBGAMBX                        GENMAX= 3   ALLOCATED          ON VOLUME(Sl : STIMS4
DBGAMBX   DBGAMBX   INDEX VSAM   88313/183028788319/1843067  9288321/1719034 NO DBGCA  N.DEF.  2 DBGRECOV
                                 08.11.88   14.11.88    16.11.88
                                 TUESDAY    MONDAY      WEDNESDAY
STIMS22O.DBG.DBGAMBY                        GENMAXMONDAY= 3   ALLOCATED          ON VOLUME(S) : STIMS5
DBGAMBY   DBGAMBY   INDEX VSAM   88313/183045688319/184321915188321/1719217 NO DBGCA  N.DEF.  2 DBGRECOV
                                 08.11.88   14.11.88    16.11.88
                                 TUESDAY    MONDAY      WEDNESDAY
NOTES

DSL     A DSL-DATA SET IS DEFINED FOR THIS DB
CAGRP   NAME OF THE CA-GROUP TO WHICH THIS DB BELONGS
MEDIUM  DEVICE ON WHICH THE CA-DS RESIDES (TAPE OR DASD)
SHRLVL  SHARE-LEVEL OF THIS DB
IF      IN FORCE FLAG (I.E. A CA-RUN HAS BEEN DONE FOR THIS DB SINCE IT'S LAST IMAGE-COPY)
UIC     USER-IMAGE-COPY (NOT USABLE FOR RECOVERY)
INV     INVALID FLAG (NOT USABLE FOR RECOVERY)
```

Figure 112. Recovery Oriented DB Overview (Condensed Listing)

This report gives a quick summary of recovery information:

- DBDS data set names
- Date and time-stamp of the oldest IC
- Date and time-stamp of the latest IC
- Date of the last allocation
- Default recovery JCL member name
- Share level
- CA group name

8.3.1.2 Data Base History

```
IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4           PAGE 0001
LIST.HISTORY -
GROUP(DBGGRP1) -
/* LIST.HISTORY COMMAND    */
88.327 12:10:44.8                 LISTING OF RECON              PAGE 0002
-------------------------------------------------------------------------
DB
 DBD=DBGAMAP                  IRLMID=NULL        DMBN=21   TYPE=IMS
 SHARE LEVEL=2
 FLAGS:                             COUNTERS:
   BACKOUT NEEDED        =OFF         RECOVERY NEEDED COUNT   =0
   READ ONLY             =OFF         IMAGE COPY NEEDED COUNT =0
   PROHIBIT AUTHORIZATION=OFF         AUTHORIZED SUBSYSTEMS   =0
                                      HELD AUTHORIZATION STATE=0
                                      EQE COUNT               =0
DBDS
 DSN=STIMS22O.DBG.DBGAMAP                            DBDS SEQ=21  TYPE=IMS
 DBD=DBGAMAP   DDN=DBGAMAP  DSID=001 DBORG=HIDAM  DSORG=VSAM DSLG SEQ=00
 CAGRP=DBGCA     GENMAX=3     IC AVAIL=2     IC USED=1     DSSN=00000003
 REUSE           RECOVPD=1
 DEFLTJCL=**NULL**  ICJCL=DBGIC     OICJCL=DBGOIC    RECOVJCL=DBGRECOV
 FLAGS:                             COUNTERS:
 IC NEEDED      =OFF
 RECOV NEEDED   =OfF
 RECOV EXECUTION=OFF                EQE COUNT                 =0
 CAGRP
 GRPNAME=DBGCA GRPMAX=3            CA AVAIL=0 CA USED=1
 NOREUSE CAJCL=DBGCA               DEFLTJCL=**NULL**
                                  #MEMBERS=5 -DBD-     -DDN-
                                             DBGAMAP   DBGAMAP
                                             DBGAMAX   DBGAMAX
                                             DBGAMBP   DBGAMBP
                                             DBGAMBX   DBGAMBX
                                             DBGAMBY   DBGAMBY
---------------------------AVAILABLE DATA SET ------------------------------
                        DBD=DBGAMAP   DDN=DBGAMAP
IMAGE
* CREATE  = 88.327    11:04:45.7*
IC1
 DSN=STIMS22O.DBG.DBGAMAP.IC2                       FILE SEQ=0001
 UNIT=3400  VOLS DEF=0000 VOLS USED=0000
                           AVAILABLE DATA SET
                        DBD=DBGAMAP   DDN=DBGAMAP
IMAGE
* CREATE = 88.327     11:04:51.8*
IC1
 DSN=STIMS22O.DBG.DBGAMAP.IC3                       FILE SEQ=0001
 UNIT=3400                         VOLS DEF=0000 VOLS USED=0000
```

Figure 113. Data Base History without DBICF (Part 1 of 2)

Figure 113 and Figure 114 show examples of database HISTORY produced using the DBRC LIST.HISTORY command.

```
88.327     12:10:44.8                 LISTING OF RECON PAGE        0003
----------------------------------------------------------------------
IMAGE
 RUN       = 88.327     11:11:50.5*    RECORD COUNT = 155
 STOP      = 00.000     00:00:00.0     BATCH
IC1
 DSN=STIMS220.DBG.DBGAMAP.IC1                         FILE SEQ=0001
 UNIT=3400                             VOLS DEF=0001  VOLS USED=0001
                                       VOLSER=STIMS5

ALLOC
 ALLOC     = 88.327     11:14:51.5*    START   = 88.327   11:14:43.9
                                       DSSN=00000002
PRILOG
  START    = 88.327     11:14:43.9     STOP = 88.327    11:19:55.3
  SSID=R82123R3        #DSN=1          IMS

  DSN=STIMS220.DBG.B01LOG.G0003V00                             UNIT=3400
  START    = 88.327     11:14:43.9     STOP = 88.327    11:19:55.3
  FILE SEQ=0001    #VOLUHES=0001   -VOLSER-         -STOPTIME-
                                    STIMS5     88.327    11:19:55.3
ALLOC
  ALLOC    = 88.327     11:20:48.5     START = 88.327     11:20:40.4
                                       DSSN=00000003

PRILOG
  START    = 88.327      11:20:40.4    STOP = 88.327    11:22:48.9
  SSID=R82123R4         #DSN=1         IMS

  DSN=STIMS220.DBG.B01LOG.G0004V00                             UNIT=3400
  START    = 88.327     11:20:40.4     STOP = 88.327    11:22:48.9
  FILE SEQ=0001    #VOLUMES=0001   -VOLSER-         -STOPTIME-
                                    STIMS5    88.327    11:22:48.9
CA
  DSN=STIMS220.DBG.CALOG.G000V00                      FILE SEQ=1
  CAGRP=DBGCA     UNIT=3400
  STOP    = 88.327       11:22:48.9*    VOLS DEF=1   VOLS USED=1
                                        VOLSER=STIMS5
  RUN     = 88.327       11:24:17.6
  -DBD-    -DDN-     -PURGETIME- -  CHG-CMP-    -LSN-     -DSSN-
  DBGAMAP DBGAMAP 88.327 11:11:50.5 YES YES 000000000000 00000003
  DBGAMAX DBGAMAX 88.327 11:12:06.6 NO  YES 000000000000 00000000
  DBGAMBP DBGAMBP 88.327 11:12:26.2 YES YES 000000000000 00000003
  DBGAMBX DBGAMBX 88.327 11:12:42.9 YES YES 000000000000 00000003
  DBGAMBY DBGAMBY 88.327 11:12:58.8 YES YES 000000000000 00000003
RECOV
  RUN      = 88.327      11:51:35.1

RECOV PAGE 0004
 RUN       = 88.327      12:06:16.2
DSP01B1I NO AVAIL CA RECORD FOUND
DSP01B1I NO AVAIL DSLOG RECORD FOUND
DSP01B1I NO DSLOG RECORD FOUND
DSP01B1I NO MRGND RECORD FOUND
DSP01B1I NO REORG RECORD FOUND
```

Figure 114. Data Base History without DBICF (Part 2 of 2)

Figure 115 on page 115 shows a DBICF report of DATA BASE HISTORY for the same database DBGAMAP.

```
                              HISTORY OF STIMS22O.DBG.DBGAMAP

RECON-ID: I220                  DATE: 22.11.88 (88327)     TIME: 12/13/19
RECON1-DSN : STIMS220.RECON1
RECON2-DSN : STIMS220.RECON2

IC1   ---> 88327     11:11:50.5  DSN              : STIMS220.DBG.DBGAMAP.IC1
           22.11.88  TUESDAY     FILE-SEQUENCE#   : 1
                                 NBR OF RECORDS   :      155
                                 VOLSER(S)        : STIMS5

ALLOC ---> 88327     11:14:51.5  SESSION          : START-88327,11:14:43.9 STOP-88327,11:19:55.3
           22.11.88  TUESDAY     DEALLOC          : -
                                 JOBNAME/SSID     : R82123R3
                                 RLDS-DSN         : STIMS220.DBG.B0llOG.G0003V00
                                 VOLSER(S)        : STIMS5 ---> DSLOG=N CHANGES=Y

ALLOC ---> 88327     11:20:48.5  SESSION           : START-88327,11:20:40.4 STOP-88327,11:22:48.9
           22.11.88  TUESDAY    DEALLOC           . -
                                JOBNAME/SSID      : R82123R4
                                RLDS-DSN          : STIMS220.DBG.B0lLOG.G0004V00 DBGAM
                                VOLSER(5)         : STIMS5 ---> DSLOG=N CHANGES=Y

CA   ---> 88327      11:24:17.6 DSN               : STIMS220.DBG.CALOG.G000lV00
          22.11.88   TUESDAY    FILE-SEQUENCE#    :   1
                                VOLSER(S)         : STIMS5

RECOV ---> 88327     11:51:35.1 FULL-RECOVERY
           22.11.88  TUESDAY

RECOV ---> 88327     12:06:16.2 FULL-RECOVERY
           22.11.88  TUESDAY
```

Figure 115. Data Base History with DBICF

8.3.1.3 DB Check Report

The DB check report is a complete listing of all databases that present any of the following conditions:

- Recovery needed
- Image copy needed
- Batch backout needed
- (Online) backout needed
- Never been saved (image copy taken)
- Last image copy invalid
- Reorganized but without a subsequent image copy
- Time-stamp recovered but without a subsequent image copy
- In use (PROCOPT > GO) by another batch subsystem
- Further authorization prohibited
- Read only flag set

This function is useful to quickly monitor the status of all the databases. The MTO or the DBA can use this function to identify abnormal situations in the system. Figure 116 shows an example of the produced report.

```
DBICF-FUNCTION: 'DATABASE CHECK'


RECON-ID . I220
DATE    : TUESDAY ,29.11.88 (88334)
TIME    . 14/19/29
RECON1-DSN : STIMS22O.RECON1
RECON2-DSN : STIMS22O.RECON2
THE FOLLOWING 'INCONSISTENCIES' HAVE BEEN FOUND IN THE RECON:
ICFl92I DB: DBGAMAP DDN: DBGAMAP RECOVERY NEEDED. DSN: STIMS22O.DBG.DBGAMAP
ICFl92I DB: DBGAMAX DDN: DBGAMAX RECOVERY NEEDED. DSN: STIMS22O.DBG.DBGAMAX
ICFl92I DB: DBGAMBP DDN: DBGAMBP RECOVERY NEEDED. DSN: STIMS22O.DBG.DBGAMBP
ICFl95I LAST IC FOR DB: DBGAMBP DDN: DBGAMBP INVALID. IC TIMESTAMP: (88.333/20.00.08.9)
ICFl92I DB: DBGAMBX DDN: DBGAMBX RECOVERY NEEDED. DSN: STIMS22O.DBG.DBGAMBX
ICFl92I DB: DBGAMBY DDN: DBGAMBY RECOVERY NEEDED. DSN: STIMS22O.DBG.DBGAMBY
ICFl94I DB: IMSAMRP DDN: IMSAMRP HAS NEVER BEEN SAVED. DSN: STIMS22O.DBG.IMSAMRP
ICFl9OI FURTHER AUTHORIZATION PROHIBITED FOR DB: DBGAMAP
ICFl9lI READ ONLY FLAG SET FOR DB: DBGAMBY
```

Figure 116. DB Check Function Report

8.3.2 OIC PSBGEN Function

Figure 117 shows the panel for OIC PSB generation.

```
--------------------      DBICF-OIC-PSBGEN     --------------------------------


                                                                USERID -  USER
                                                                DATE   -  DATE


   WHICH PSB     ENTER PSB-HAME
                 ST1- GENERATION OF PSB-STATEMENTS (PX021038)
                 ST2- COMPILATION OF PSB
                 ST3- LINK INTO "IMSVS.PSBLIB"
                 ST4- ACBGEN INTO "IMSVS.ACBLIB1"




PRESS END KEY TO RETURN TO THE PREVIOUS MENU
____________________________________________________________________________________
```

Figure 117. DBICF Panel for Online Image Copy PSBGEN

To guarantee that an OIC of a specific database data set recorded in RECON can be performed under IMS, the PSB source generator creates an OIC PSB' for all DBDS that are registered in RECON and have an OIC JCL member name with the prefix OIC in the appropriate DBDS record.

The user also receives a report containing the names of the DBDs building, the OIC PSB.

This function could be run after DB migrations (INIT.DB or DELETE.DB) and IMS distributions which require an ACBGEN run.

8.4 DBICF Batch Functions

A second way of working with DBICF is to submit the provided batch jobs to perform specific tasks in normal and emergency situations. Three of approximately 50 functions are:

- Authorization checking
- RECON backup

• RECON reorganization

An example of each of these is included in the following sections.

8.4.1 Authorization Checking

Figure 118 shows the authorization checking function that can be placed as a preliminary step in every IMS batch job or executed as a separate job.

```
// RUN DBICF AUTHOCHK PROGRAM
//*
//AUTHCHK PROC PSB=
//*
//*
//* JCL EXAMPLE FOR THE DBICF-FUNCTION:
//
//s  ' P S B        A U T H 0 R I Z A T I 0 N       C H E C K I N G '
//***
//AUTHOCK1 EXEC PGM=PX021063,PARM='&PSB',REGION=2048K
//STEPLIB   DD DSN=SYSC.STIMS22X.RESLIB,DISP=SHR
//DFSRESLB  DD DSN=SYSC.STIMS22X.RESLIB,DISP=SHR
//IMS       DD DSN=SYSC.STIMS22X.CTLLIB,DISP=SHR
//RECON3    DD DUMMY
//IEFRDER   DD DSN=&&AUTHOCHK,UNIT=SYSDA,SPACE=(TRK,1),DISP=(,DELETE)
//DFSVSAMP  DD DSN=STIMS22X.DBG.UTIL(ISSVBUF),DISP=SHR
//REPORT    DD SYSOUT=*,DCB=(LRECL=133,BLKSIZE=133,RECFM=FBA)
//RUPDLOG   DD DSN=STIMS22X.RECON.LOG,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//SYSIN DD DSN=DDSYSIN,DISP=(,PASS),SPACE=(TRK,1),DCB=BLKSIZE=80)
// UNIT=SYSDA
//DBDYNALL  DD DUMMY (IF THE DB'S ARE TO BE ALLOCATED
//             DYNAMICALLY (IMS/VS V.2 ONLY))
//NOREQ DD DUMMY (IF 'REQUEUE' IS NOT ALLOWED IN YOUR ENVIRONMENT)
//NOCANCEL  DD DUMMY (IF 'CANCEL' IS NOT ALLOWED IN YOUR ENVIRONMENT)
//******************* END OF JOB AUTHOCHK ************************
// PEND
//A    EXEC AUTHCHK,PSB=DBGC01
```

Figure 118. Authorization Checking

The authorization checking function can only have two possible cases:

1. In case of a signon/authorization failure, a message is sent to the operator with a pending reply number. The operator, a CLIST, or any other automation program can choose between four types of action:
   - **RETRY**, repeat the signon/authorization check (for example: after image copy when a database was IC-needed, or after the deletion of an erroneously remaining subsystem record).
   - **REQUEUE**, the whole job is queued for re-execution.
   - **CANCEL**, the whole job is cancelled.
   - **ABORT**, the user acknowledges the failure and the job continues.
2. In case of successful DBRC signon/authorization, the user receives a report about the DB authorizations, access intents, processing states, and so on, as shown in Figure 119.

```
D B I C F   P S B   A U T H O R I Z A T I 0 N   C H E C K I N G
------------------------------------------------------------

THE FOLLOWING DB-DATASET(S) HAS/HAVE BEEN AUTHORIZED BY PSB: DBGC01


DBD-NAME  DB-DDNAME  DB-DSNAME                   ACCESS-INTENT + SHARE-LEVEL = PROCESSING-STATE

--------  --------  ---------

DBGAMAP   DBGAMAP   STIMS220.DBG.DBGAMAP        READ           BLS (INTRA)    READ SHARE (2)
DBGAMAX   DBGAMAX   STIMS220.DBG.DBGAMAX        READ           BLS (INTRA)    READ SHARE (2)
DBGAMBX   DBGAMBX   STIMS220.DBG.DBGAMBX        READ           BLS (INTRA)    READ SHARE (2)
DBGAMBP   DBGAMBP   STIMS220.DBG.DBGAMBP        READ           BLS (INTRA)    READ SHARE (2)
DBGAMBY   DBGAMBY   STIMS220.DBG.DBGAMBY        READ           BLS (INTRA)    READ SHARE (2)

(BATCH) LOGGING NOT REQUIRED.


N.B. THE DISPLAYED PROCESSING-STATE(S) = COMPOSITE USE OF THE DATABASE BY ALL
  CURRENTLY AUTHORIZED SUBSYSTEM
```

Figure 119. Output of Authorization Checking

Note: Since only physical databases are registered in the RECON, only PSBs that use physical DBDs are checked and listed.

8.4.2 RECON Backup

Figure 120 shows the JCL needed for creating a portable RECON BACKUP data set.

```
//* *************************************************************************
//* * JCL EXAMPLE FOR THE CREATION OF A PORTABLE RECON-BACKUP               *
//* *           **** FOR DBRC R3/4 USER'S ****                              *
//* *************************************************************************
//DELDEFB EXEC PGM=IDCAMS,REGION=4096K
//* *************************************************************************
//* * DELETE/DEFINE CLUSTER STEP OF A RECON DATA SET                        *
//* *************************************************************************
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
 DELETE STIMS220.RECONB
 SET LASTCC=0
 DEFINE CLUSTER (NAME(STIMS220.RECONB)               -
                VOL (SBV010) INDEXED                 -
                KEYS(24 0) CYL(5 2)                  -
                RECSIZE(128 4089) NONSPANNED         -
                FRSP(30 80) CISZ(4096) BUFSP(24576) -
                NOERASE NERAS SPEED REPL IMBED       -
                UNORDERED SHR(3 3)                   -
         INDEX (NAME(STIMS220.RECONB.INDEX)) -
         DATA  (NAME(STIMS220.RECONB.DATA))
//* *************************************************************************
//* * BACKUP.RECON STEP (DBRC AUTOMATICALLY BACKS UP A VALID RECON COPY      *
//* *************************************************************************
//BACKUP EXEC PBMPZ021072,REGION=4096K
//STEPLIB  DD DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//RUPDLOG  DD DSN=STIMS220.RECON.LOG,DISP=SHR
//BACKPUP1 DD DSN=STIMS220.RECONB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
BACKUP.RECON
//* *************************************************************************
//* * EXPORT STEP OF THE NEWLY CREATED RECON BACKUP DATA SET TO             *
//* *     A PORTABLE DEVICE (TAPE)                                          *
//* *************************************************************************
//EXPORT  EXEC PGM=PX021072,REGION=4096K
//STEPLIB  DD  DSN=SYSC.STIMS220.RESLIB,DISP=SHR
//SYSUT2   DD  DSN=PRDSEC.RECONS(+1),DISP=(,CATLG,DELETE),
//             UNIT=TAPE,DCB=GDGMOD,VOL=(,RETAIN),LABEL=EXPDT=99000
//SYSUT1   DD  DSN=STIMS220.RECONB,DISP=SHR
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD *
EXPORT - STIMS220.RECONB -
INFILE(SYSUT1)
OUTFILE(SYSUT2) -
TEMPORARY -
NOINHIBITSOURCE
/*
```

Figure 120. RECON Backup

8.4.3 RECON Reorganization

Figure 121 shows the JCL for RECON reorganization that contains the following steps:

1. A DELETE.LOG step deleting all inactive PRILOG, LOGALL, and ALLOC records. Those commands ask for deletion of:
   - Records not related to DB updates and older than 24 hours
   - Records related to DB updates but older than the oldest image copy record for each DBDS
2. A REORGANIZATION step creating a spare RECON and replacing the spare RECON with a valid RECON copy.
3. A BASIC step which sets the RECON data sets to the original positions:

```
//*
//* ' A U T O H A T I C R E C O N R E 0 R G A N I Z A T I 0 N
//*
//* FOR DBRC R3/R4 USER'S WITH SPARE RECON
//*
//DELLOG EXEC PGM=PX021072,REGION=2048K
//*
//* DELETE.LOG.INACTIVE
//*
//STEPLIB  DD DSN=SYSC.STIMS22X.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//RUPDLOG  DD DSN=STIMS22X.RECON.LOG,DISP=SHR
//SYSIN    DD *
DELETE.LOG INACTIVE
/*
//REORG  EXEC PGM=PX021078,PARM='REORG',REGION=4096K
//STEPLIB  DD DSN=SYSC.STIMS22X.RESLIB,DISP=SHR
//**
//* REORG RECONS
//**
//STEPLIB  DD DSN=SYSC.STIMS22X.RESLIB,DISP=SHR
//RUPDLOG  DD DSN=STIMS22X.RECON.LOG,DISP=SHR
//SYSPRINT DD SYSOUT=*
//RCUSYSIN DD DSN=&&SSIN,DISP=(,KEEP),UNIT=SYSDA,SPACE=(TRK,2),
//            DCB=(LRECL=SO,BLKSIZE=SO,RECFM=F)
//DELDEFR1 DD DSN=STIMS22X.DBG.UTIL(RECON1),DISP=SHR
//DELDEFR2 DD DSN=STIMS22X.DBG.UTIL(RECON2),DISP=SHR
//DELDEFR3 DD DSN=STIMS22X.DBG.UTIL(RECON3),DISP=SHR
//*
//BASIC EXEC PGM=PX021084,REGION=4096K
//*
//* SET BASIC RECON POSITION
//**
//STEPLIB  DD DSN=SYSC.STIMS22X.RESLIB,DISP=SHR
//RUPDLOG  DD DSN=STIMS22X.RECON.LOG,DISP=SHR
//SYSPRINT DD SYSOUT=
//RCUSYSIN DD DSN=&&SYSIN,DISP=(,KEEP),UNIT=SYSDA,SPACE=(TRK,2),
//            DCB=(LRECL=80,BLKSIZE=80,RECFM=F)
//DELDEFR1 DD DSN=STIMS22X.DBG.UTIL(RECON1),DISP=SHR
//DELDEFR2 DD DSN=STIMS22X.DBG.UTIL(RECON2),DISP=SHR
//DELDEFR3 DD DSN=STIMS22X.DBG.UTIL(RECON3),DISP=SHR
```

Figure 121. Creating the RECON Reorganization JCL with DBICF

- RECON1 = COPY1
- RECON2 = COPY2
- RECON3 = SPARE

This performs as many RECON reorganization runs as needed.

This procedure should be run at a low RECON activity time (it is better if there is no activity). This function can also be run during IMS online or CICS.

If, for any reason, the RECON basic position cannot be reached (for example, because a RECON data set is still used by IMS online), message ICF144A informs the user about what can be done.

Figure 122 shows the generated JCL for actually performing the RECON backups.

```
//DBRC     EXEC PGM=DSPURX00,COND=(O,NE)
//STEPLIB  DD DSN=SYSC.STIMS22X.RESLIB,DISP=SHR
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
DELETE.LOG INACTIVE
//*
//DBRC     EXEC PGM=DSPURX00
//STEPLIB  DD DSN=SYSC.STIMS22X.RESLIB,DISP=SHR
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
CHANGE.RECON REPLACE(RECON1)
//*
//RECON1  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=
//SYSIN    DD DSN=STIMS22X.DBG.UTIL(RECONl),DISP=SHR
//*
//DBRC2         EXEC PGM=DSPURX00
//STEPLIB  DD DSN=SYSC.STIMS22X.RESLIB,DISP=SHR
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
CHANGE.RECON REPLACE(RECON2)
//
//RECON2  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DSN=STIMS22X.DBG.UTIl(RECON2),DISP=SHR
//*
//DBRC3   EXEC PGM=DSPURX00
//STEPLIB  DD DSN=SYSC.STIMS22X.RESLIB,DISP=SHR
//IMS      DD DSN=STIMS22X.DBDLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
CHANGE.RECON REPLACE(RECON3)
//*
//RECON3  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DSN=STIMS22X.DBG.UTIL(RECON3),DISP=SHR
```

Figure 122. RECON Reorganization Job Stream

Figure 123 on page 121 through Figure 126 on page 124 show the outputs of the RECON reorganization steps.

```
        IMS/VS DATA BASE RECOVERY CONTROL SERVZCES, RELEASE 4  PAGE 0001
DELETE.LOG INACTIVE
DSPO126I   NUMBER OF INACTIVE LOG DATA SETS DELETED WAS 00000
DSPO203I   COMMAND COMPLETED WITH CONDITION CODE 00
D5P0220I   COMMAND COMPLETION TIHE 88.327   18:21:46.3
         IMS/VS DATA BASE RECOVERY CONTROl SERVICES, RELEASE 4  PAGE 0002
DSPO2llI   COMMAND PROCESSING COMPLETE
DSPO2llI   HIGHEST CONDITION CODE = 00
           IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4  PAGE 0001
LIST.RECON STATUS

88327    18:21:52.0                  LISTING-OF RECON            PAGE 002
----------------------------------------------------------------------
RECON
 RECOVERY CONTROL DATA SET, RELEASE 4
 DMBN=20    HIGHEST DBDS SEQ=20 INIT TIMESTAMP=88.326  12:23:45.3
 NOFORCER   LOG DSN CHECK=CHECKl7 STARTNEW=NOCONTROL=SHARE
 TAPE UNIT=3400  DASD UNIT=3400              TRACEOFF SSID=I22X

-DDNAME-   -STATUS- -DATA SET NAHE-

 RECON1     COPY1 STIMS22X.RECONI
 RECON2     COPY2 STIMS22X.RECON2
 RECON3     SPARE STIMS22X.RECON3

DSPOlBOI   NUMBER OF RECORDS LISTED IS      1
D5P0203I   COMMAND COMPLETED NITH CONDITION CODE 00
DSPO22OI   COMMAND COMPLETION TIHE 88.327   18:21:55.6
           IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4  PAGE 0003
DSPO2llI   COMMAND PROCESSING COMPLETE
DSPO2llI   HIGHEST CONDITION CODE = 00
           IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4  PAGE 0001
CHANGE.RECON REPLACE(RECON1)

DSPO3BOI   RECON2 COPY TO RECON3 START
DSPO3BlI   COPY COMPLETE, RC = 000
DSPO242I   RECON1 DSN=STIMS22X.RECON1
DSPO242I   REPLACED BY
DSPO242I   RECON3 DSN=STIMS22X.RECON3
D5P0203I   COMMAND COMPLETED WITH CONDITION CODE 00
DSPO22OI   COMMAND COMPLETION TIME 88.327 18:22:01.9
           IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4 PAGE 0002
DSPO2llI   COMHAND PROCESSING COHPLETE
DSPO2llI   HIGHEST CONDITION CODE = 00
        IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4        PAGE 0001
LIST.RECON STATUS
88.327   18:22:04.3                LISTING OF RECON                  PAGE 0002
---------------------------------------------------------------------------
RECON
 RECOVERY CONTROL DATA SET, RELEASE 4
 DMB#=20   HIGHEST DBDS SEQ=20      INIT TIMESTAMP=88.326   12:23:45.3
 NOFORCER  LOG DSN CHECK=CHECKl7    STARTNEW=NO     CONTROL=SHARE
 TAPE UNIT=3400      DASD UNIT=3400      TRACEOFF   SSID=I22X

 -DDNAME-     -STATUS-       -DATA SET NAME-
RECON2         COPY1          STIMS22X.RECON2
RECON3         COPY2          STIMS22X.RECON3
RECON1         DISCARDED      STIMS22X.RECON1

DSPOlBOI NUMBER OF RECORDS LISTED IS        1
DSPO2O3I COMMAND COMPLETED WITH CONDITION CODE 00
DSPO22OI COMMAND COMPLETION TIME 88.327 18:22:07.9
       IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4         PAGE 0003
DSPO2llI COMMAND PROCESSING COMPLETE
DSPO2llI HIGHEST CONDITION CODE = 00
IDCAMS SYSTEM SERVICES                                           TIME:18:22:12
  DELETE (STIMS22X.RECON1) CL PRG

IDCO55OI ENTRY (D) STIMS22X.RECONl.DATA DELETED
IDCO55OI ENTRY (I) STIMS22X.RECONl.INDEX DELETED
IDCO5SOI ENTRY (Cl STIMS22X.RECON1 DELETED
IDCOOOlI FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

  DEFINE CLUSTER(                              -
                 NAME(STIMS22X.RECON1)         -
                 FREESPACE(20 20) INDEXED       -
                 KEYS(24 0) RECORDSIZE(128 6137) -
                 SHR(3 3) IMBED REPLICATE       -
                 RECOVERY NOERASE SPANNED       -
                 NOREUSE UNORDERED WRITECHECK   -
                 UNIQUE VOL(STIMS4              -
                 CYLINDERS(02 Ol)              -
                )-
            DATA(-
                 NAME(STIMS22X.RECONl.DATA)    -
                 CONTROLINTERVALSIZE(6144)     -
                )-
           INDEX(-
                 NAME(STIMS22X.RECONl.INDEX)-
                       )
IDCOSOBI DATA ALLOCATION STATUS FOR VOLUME STIMS4 IS 0
IDCO5O9I INDEX ALLOCATION STATUS FOR VOLUME STIMS4 IS 0
IDCOOOlI FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
IDCOOO2I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

Figure 123. Output of RECON Reorganization (Part 1 of 4)

```
        IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4          PAGE 0001
LIST.RECON STATUS
88.327   18:22:18.1                 LISTING OF RECON                   PAGE 0002
---------------------------------------------------------------------------------
RECON
 RECOVERY CONTROL DATA SET, RELEASE 4
 DMB#=20   HIGHEST DBDS SEQ=20      INIT TIMESTAMP=88.326   12:23:45.3
 NOFORCER  LOG DSN CHECK=CHECKl7    STARTNEW=NO     CONTROL=SHARE
 TAPE UNIT=3400     DASD UNIT=3400       TRACEOFF   SSID=I22X

-DDNAME-       -STATUS-     -DATA SET NAME-
 RECON2         COPY1        STIMS22X.RECON2
 RECON3         COPY2        STIMS22X.RECON3
 RECON1         SPARE        STIMS22X.RECON1

DSPOlBOI NUMBER OF RECORDS LISTED IS       1
DSPO203I COMMAND COMPLETED WITH CONDITION CODE 00
DSPO22OI COMMAND COMPLETION TIHE 88.327    18:22:22.6

        IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4          PAGE 0003
DSPO2llI COMMAND PROCESSING COMPLETE
DSPO2llI HIGHEST CONDITION CODE = 00
        IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4          PAGE 0001
CHANGE.RECON REPLACE(RECON3)

DSPO3BOI RECON2 COPY TO RECON1       STARTED
DSPO3BlI COPY COMPLETE, RC = 000
DSPO242I RECON3 DSN=STIHS22X.RECON3
DSPO242I REPLACED BY
DSPO242I RECON1 DSN=STIMS22X.RECON1
DSPO203I COMMAND COMPLETED WITH CONDITION CODE 00
DSPO22OI COMMAND COMPLETION TIME 88.327   18:22:28.7
           IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4          PAGE 0002
DSPO2llI COMMAND PROCESSING COMPLETE
DSPO2llI HIGMEST CONDITION CODE = 00
        IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4          PAGE 0001
LIST.RECON STATUS
88.327   18:22:31.0                  LISTING OF RECON                  PAGE 0002
--------------------------------------------------------------------------------
RECON
 RECOVERY CONTROL DATA SET, RELEASE 4
 DMB#=20   HIGHEST DBDS SEQ=20       INIT TIMESTAMP=88.326  12:23:45.3
 NOFORCER  LOG DSN CHECK=CHECKl7     STARTNEW=NO     CONTROL=SHARE
 TAPE UNIT=3400     DASD UNIT=3400        TRACEOFF   SSID=I22X

-DDNAME- -STATUS-           -DATA SET NAME-

 RECON2   COPY1              STIMS22X.RECON2
 RECON1   COPY2              STIMS22X.RECON1
 RECON3   DISCARDED          STIMS22X.RECON3

DSPOlBOI NUMBER OF RECORDS LISTED IS       1
DSPO203I COMMAND COMPLETED WITH CONDITION CODE 00
DSPO22OI COMMAND COMPLETION TIME 88.327    18:22:34.6
        IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4         PAGE 0003
DSPO2lI COMMAND PROCESSING COMPLETE
DSPO2llI HIGHEST CONDITION CODE = 00
lDCAMS SYSItM SERVICES                                          TIME:l8:22:38
  DELETE (STIMS22X.RECON3) CL PRG
IDCOS5OI ENTRY (Dl STIMS22X.RECON3.DATA DELETED
                                                                I
IDCO55OI ENTRY (I) STIMS22X.RECON3.INDEX DELETED
IDCO55OI ENTRY (Cl STIHS22X.RECON3 DELETED
IDCOOOlI FUNCTION COMPlETED, HIGHEST CONDITION CODE WAS 0
   DEFINE CLUSTER(                        -
                  NAME(STIMS22X.RECON3)   -
                  FREESPACE(20 20) INDEXED -
                  KEYS(24 0) RECORDSIZE(128 6137) -
                  SHR(3 3) IMBED REPLICATE -
                  RECOVERY NOERASE SPANNED -
                  NOREUSE UNORDERED WRITECHECK    -
                  UNIQUE VOL(STIMS4)       -
                  CYL (02 01)        -
                  )                       -
              DATA(                       -
                   NAME(STIMS22X.RECON3.DATA) -
                   CONTROLINTERVALSIZE(6144)  -
                  )                           -
             INDEX(                           -
                   NAME(STIMS22X.RECON3.INDEX)                      -
                        )
IDCO5OBI DATA ALLOCATION STATUS FOR VOLUME STIMS4 IS 0
IDCO509I INDEX ALLOCATION STATUS FOR VOLUME STIMS4 IS 0
IDCOOOlI FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
IDCOOO2I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

Figure 124. Output of RECON Reorganization (Part 2 of 4)

```
        IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4            PAGE 0001
LIST.RECON STATUS
88.327   18:22:43.8 LISTING OF RECON                                      PAGE 0002
----------------------------------------------------------------------------------
RECON
 RECOVERY CONTROL DATA SET, RELEASE 4
 DMB#=20   HIGHEST DBDS SEQ=20      INIT TIMESTAMP=88.326  12:23:45.3
 NOFORCER  LOG DSN CHECK=CHECK17    STARTNEW=NO     CONTROL=SHARE
 TAPE UNIT=3400     DASD UNIT=3400  TRACEOFF        SSID=I22X

-DDNAME-      -STATUS- -DATA SET NAME-

 RECON2        COPY1    STIMS22X.RECON2
 RECON1        COPY2    STIMS22X.RECON1
 RECON3        SPARE    STIMS22X.RECON3

DSP01BOI   NUMBER OF RECORDS LISTED IS      1
DSP0203I   COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I   COMMAND COMPLETION TIME 88.327   18:22:48.4
           IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4          PAGE 0003
DSP0211I   COMMAND PROCESSING COMPLETE
DSP0211I   HIGHEST CONDITION CODE = 00
           IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4          PAGE 0001
CHANGE.RCON REPLACE(RECON2)
DSP03BOI   RECON1 COPY TO RECON3 STARTED
DSP03B1I   COPY COMPLETE, RC = 000
DSP0242I   RECON2 DSN=STIMS22X.RECON2
DSP0242I   REPLACED BY
DSP0242I   RECON3 DSN=STIMS22X.RECON3
DSP0203I   COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I   COMMAND COMPLETION TIME 88.327   18:22:54.4
           IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4          PAGE 0002
DSP0211I   COMMAND PROCESSING COMPLETE
DSP0211I   HIGHEST CONDITION CODE = 00
         IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4              PAGE 0001
LIST.RECON STATUS
88327    1822:569                   LISTING-OF-RECON                        PAGE-0002
RECON
  RECOVERY CONTROL DATA SET, RELEASE 4

  DMB#=20   HIGHEST DBDS SEQ=20      INIT TIMESTAMP=88.326   12:23:45.3
  NOFORCER  LOG DSN CHECK=CHECK17    STARTNEW=NO     CONTROL=SHARE
  TAPE UNIT=3400     DASD UNIT=3400       TRACEOFF   SSID=I22X

-DDNAME-        -STATUS-     -DATA SET NAME-
  RECON1         COPY1        STIMS22X.RECON1
  RECON3         COPY2        STIMS22X.RECON3
  RECON2         DISCARDED    STIMS22X.RECON2
DSP01BOI NUMBER OF RECORDS LISTED IS 1
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I COMMAND COMPLETION TIME 88.327 18:23:00.5
        IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4               PAGE 0003
DSP0211I COMMAND PROCESSING COMPLETE
DSP0211I HIGHEST CONDITION CODE = 00
IDCAMS SYSTEM SERVICES TIME:1B:23:04
  DELETE (STIMS22X.RECON2) CL PRG
IDC0550I ENTRY (D) STIMS22X.RECON2.DATA DELETED
IDC05SOI ENTRY (I) STIMS22X.RECON2.INDEX DELETED
IDC0550I ENTRY (C) STIMS22X.RECON2 DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

  DEFINE CLUSTER(                      -
                 NAME(STIMS22X.RECON2) -
                 FRSP(20 20) INDEXED   -
                 KEYS(24 0) RECSIZE(128 6137)-
                 SHR(3 3) IMBED REPLICATE  -
                 RECOVERY NOERASE SPANNED  -
                 NOREUSE UNORDERED WRITECHECK -
                 UNIQUE VOL(STIMS4)  -
                CYLINDERS(02 O1)    -
                )                    -
           DATA(                     -
                 NAME(STIMS22X.RECON2.DATA) -
                 CONTROLINTERVALSIZE(6144)  -
                )                           -
          INDEX(                            -
                 NAME(STIMS22X.RECON2.INDEX)-
                      )
IDC05SI DATA ALLOCATION STATUS FOR VOLUME STIMS4 IS 0
IDC0509I INDEX ALLOCATION STATUS FOR VOLUME STIMS4 IS 0
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
         IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4           PAGE 0001
LIST.RECON STATUS
88.327   18:23:09.6                 LISTING OF RECON                     PAGE 0002
RECON
 RECOVERY CONTROL DATA SET, RELEASE 4
 DMB1I=20   HIGHEST DBDS SEQ=20     INIT TIMESTAMP=88.326  12:23:45.3
 NOFORCER   LOG DSN CHECK=CHECK17   STARTNEW=HO     CONTROL=SHARE
 TAPE UNIT=3400      DASD UNIT=3400      TRACEOFF   SSID=I22X

-DDNAME-   -STATUS-      -DATA SET NAME-

 RECON1     COPY1         STIMS22X.RECON1
 RECON3     COPY2         STIMS22X.RECON3
 RECON2     SPARE         STIMS22X.RECON2

DSP01BOI NUMBER OF RECORDS LISTED IS  1
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I COMMAND COMPLETION TIME 88.327  18:23:14.2
```

Figure 125. Output of RECON Reorganization (Part 3 of 4)

```
        IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4           PAGE 0003
DSPO21lI COMMAND PROCESSIHG COMPLETE
DSPO21lI HIGHEST CONDITION CODE = 00
        IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4           PAGE 0001
CHANGE.RECON REPLACE(RECON3)
DSPO3BOI RECON1 COPY TO RECON2  STARTED
DSPO3BlI COPY COMPLETE, RC = 000
DSPO242I RECON3 DSN=STIMS22X.RECON3
DSPO242I REPLACED BY
DSPO242I RECON2 DSN=STIMS22X.RECON2
DSPO203I COMMAND COMPLETED WITH CONDITION CODE 00
DSPO22OI COMMAND COMPLETION TIME 88.327  18:23:20.0
        IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4           PAGE 0002
D5P0211I COMMAND PROCESSING COMPLETE
DSPO21lI HIGHEST CONDITION CODE = 00
        IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4           PAGE 0001
LIST.RECON STATUS
88.327 18:23:22.4        lISTING Of RECON                               PAGE 0002
-------------------------------------------------------------------------
RECON
RECOVERY CONTROL DATA SET, RELEASE 4
DHB#=20    HIGHEST DBDS SEQ=20     INIT TIHESTAMP=88.326  12:23:45.3
NOFORCER   LOG DSN CHECK=CHECKl7   STARTNEW=NO     CONTROL=SHARE
TAPE UNIT=3400      DASD UNIT=3400      TRACEOFF   SSID=I22X

-DDNAME-   -STATUS-      -DATA SET NAME-

RECON1      COPY1         STIMS22X.RECON1
RECON2      COPY2         STIMS22X.RECON2
RECON3      DISCARDED     STIMS22X.RECON3

DSPOlBOI NUMBER OF RECORDS LISTED IS  1
DSPO203I COMMAND COMPLETED WITH CONDITION CODE 00
DSPO22OI COMMAND COMPLETION TIME 88.327  18:23:25.9
        IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4            PAGE 0003
DSPO21lI COMMAND PROCESSING COMPLETE
DSPO21lI HIGHEST CONDITION CODE = 00
IDCAMS SYSTEN SERVICES


   DELETE (STIMS22X.RECON31 CL PRG
IDCO5SOI ENTRY (D) STIMS22X.RECON3.DATA DELETED
IDCO55OI ENTRY (I) STIMS22X.RECON3.INDEX DELETED
IDCO55OI ENTRY (Cl STIMS22X.RECON3 DELETED
IDCOOOlI FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

   DEFINE CLUSTER(
                  NAME(STIMS22X.RECON3)        -
                  FREESPACE(20 20) INDEXED     -
                  KEYS(24 0) RECSIZE 128 6137) -
                  SHR(3 3) IMBED REPLICATE     -
                  RECOVERY NOERASE SPANNED     -
                  NOREUSE ORDERED WRITECHECK   -
                  UNIQUE VOL(STIMS4)           -
                  CYLINDERS(02 Ol))            -
             DATA(
                  NAME(STIMS22X.RECON3.DATA)   -
                  CONTROLINTERVALSIZE(6144)    -
                  )                            -
            INDEX(
                  NAME(STIMS22X.RECON3.INDEX)  -
                       )
IDCO5OBI DATA ALLOCATION STATUS FOR VOLUME STIMS4 IS 0
IDCO509I INDEX ALLOCATION STATUS FOR VOLUHE STIMS4 IS 0
IDCOOOlI FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDCOOO2I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4  PAGE 0001
LIST.RECON STATUS
88327- -182334.1- -- -- -- -- - LISTING-OF RECON--   -- -  -- - PAGE-0002
RECON
RECOVERY CONTROL DATA SET, RELEASE 4
DMB#=20   HIGHEST DBDS SEQ=20      INIT TIMESTAMP=88.32612:23:45.3
NOFORCER  LOG DSN CHECK=CHECKl7    STARTNEW=NO     CONTROL=SHARE
TAPE UNIT=3400     DASD UNIT=3400       TRACEOFF   SSID=I22X

-DDNAME-      -STATUS- -DATA SET NAME-
RECON1         COPY1    STIMS22X.RECON1
RECON2         COPY2    STIHS22X.RECON2
RECON3         SPARE    STIMS22X.RECON3

DSPOlBOI NUMBER OF RECORDS LISTED IS 1
DSPO2O3I COMMAND COMPLETED WITH CONDITION CODE 00
DSPO2OI COMMAND COMPLETION TIME 88.327 18:23:38.6
       IMS/VS DATA BASE RECOVERY CONTROL SERVICES, RELEASE 4 PAGE 0003
DSPO21lI COMMAND PROCESSING CONPLETE
DSPO21lI HIGHEST CONDITION CODE = 00
```

Figure 126. Output of RECON Reorganization (Part 4 of 4)

Appendix A. Special Notices

This publication is intended to help IMS system programmers set up the environment to provide Web applications that connect to IMS transactions. The information in this publication is not intended as the specification of any programming interfaces that are provided by IMS. See the PUBLICATIONS section of the IBM Programming Announcement for IMS product for more information about what publications are considered to be product documentation.

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 IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

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

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

This document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

You can reproduce a page in this document as a transparency, if that page has the copyright notice on it. The copyright notice must appear on each page being reproduced.

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

DFSMSdfp	DFSMSdss
DFSMShsm	DFSORT
IBM	Language Environment
Netfinity	NetView
OS/390	Parallel Sysplex
PR/SM	Processor Resource/Systems Manager
RACF	S/390
RS/6000	SP
System/390	WebSphere
VisualAge	XT
400	

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

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

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

SET and the SET logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

Appendix B. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

B.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 129.

- *Making Your IMS System Ready for Year 2000:Mirating to Version 5 SG24-2211*
- *IMS/ESA Version 6 Guide*, SG24-2228

B.2 Redbooks on CD-ROMs

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at `http://www.redbooks.ibm.com/` for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr Format)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** `http://www.redbooks.ibm.com/`

  Search for, view, download, or order hardcopy/CD-ROM redbooks from the redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this redbooks site.

  Redpieces are redbooks in progress; not all redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the redbooks fax order form to:

	e-mail address
In United States	usib6fpl@ibmmail.com
Outside North America	Contact information is in the "How to Order" section at this site: `http://www.elink.ibmlink.ibm.com/pbl/pbl/`

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the "How to Order" section at this site: `http://www.elink.ibmlink.ibm.com/pbl/pbl/`

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the "How to Order" section at this site: `http://www.elink.ibmlink.ibm.com/pbl/pbl/`

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and redbooks by accessing the IBM Intranet Web site at `http://w3.itso.ibm.com/` and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at `http://w3.ibm.com/` for redbook, residency, and workshop announcements.

IBM Redbook Fax Order Form

Please send me the following:

Title	Order Number	Quantity

First name | Last name

Company

Address

City | Postal code | Country

Telephone number | Telefax number | VAT number

☐ Invoice to customer number

☐ Credit card number

Credit card expiration date | Card issued to | Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

List of Abbreviations

ABEND	Abnormal End of Task
ADS	Area Data Set
AMS	Access Method Services
AUTH-ID	Authorization ID
BCS	Basic Catalog Structure
BMP	Batch Message Processing
CA	Change Accumulation
CA	Control Area
CAGRP	Change Accumulation Group
CI	Control Interval
CIC	Concurrent Image Copy
CICS/0S/VS	Customer Information Control System/Operation System/Virtual Storage
CLIST	Command List
CHKPT	Checkpoint
DB	Data Base
DBA	Data Base Admistrator
DBDS	Data Base Data Set
DBDSGRP	Data Base Data Set Group
DBGCA	Data Base Group Change Accumulation
DBICF	Data Base Integrity Control Facility
DBM	Data Base Management
DBRC	Data Base Recovery Control
DC	Data Communication
DD	Data Definition
DEDB	Data Entry Data Base
DEQ	Dequeue
DFHSM	Data Facility Hierarchical Storage Manager
DLI	Data Language Interface
DL/I	Data Language /Interface
DLISAS	DLI Seperate Address Space
DMB	Data Management Block
DP	Data Processing
DSN	Data Set Name
DSSN	Data Set Sequence Number
ENQ	Enqueue
FP	Fast Path
GDG	Generation Data Group
GPAR	Generalized Performance Analysis Reporting
GSAM	Generalized Sequential Access Method
HIDAM	Hierarchical Indexed Direct Access method
HISAM	Hierarchical Indexed Sequential Access Method
HSM	Hierarchical Storage Manager
IBM	International Business Machines Corporation
IC	Image Copy
ICF	Integrated Catalog Facility
ID	Identifier
IFP	IMS/VS Fast Path
IMS/VS	Information Management System/Virtual Storage
IRLM	IMS/VS Resouce Lock Manager
ISPF	Interactive System Productivity Facility
ISPF/PDF	Interactive System Productivity Facility/Program Development
ITSO	International Technical Support Organization
JCL	Job Control Language
K (Bytes)	1024(Bytes)
KSDS	Key Sequence Data Set
MN	Merge Needed
MPP	Message Processing Program
MTO	Master Terminal Operator
MVS	Multiple Virtual Storage (system)
NOFEOV	No Forced End Of Volume
OIC	Online Image Copy
OLDS	Online Log Data Set
OS	Operation System
PCB	Program Control Block
PDS	Partitioned Data Set
PI	Program Isolation
PSB	Program Specification Block

PTF	Program Temporary Fix
RACF	Resource Access Control Facility
RD	Read With Integrity
RECON	Recovery Control (Data Set)
RLDS	Recovery Log Data Set
RO	Read-Only (Without Integrity)
SLDS	System Log Data Set
SSID	Sub System Identifier
TP	Teleprocessing
TSO	Time Sharing Option
VSAM	Vitural Storage Access Method
VVDS	Virtual Volume Data Set

Index

S

T

U

ITSO Redbook Evaluation

Database Recovery Control (DBRC) Examples and Usage Hints
SG24-3333-01

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at `http://www.redbooks.ibm.com`
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to `redbook@us.ibm.com`

Which of the following best describes you?
_ Customer _ Business Partner _ Solution Developer _ IBM employee
_ None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction __________

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

If no, please explain:

What other redbooks would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

SG24-3333-01

Printed in the U.S.A.

Database Recovery Control (DBRC) Examples and Usage Hints

SG24-3333-01