B71

Understanding Database Recovery Control (DBRC) and the Recovery Control Data Set (RECON), Part 1 & Part 2

Pat Schroeck

IMS

Technical Conference

Sept. 27-30, 2004

Orlando, FL

RETURN TO INDEX

Agenda

- DBRC - background
- The RECON
- Database registration
- IMS calls to DBRC
- Other DBRC commands
- DBRC API
- Misc. RECON topics

What is DBRC?

- IMS **D**ata **B**ase **R**ecovery **C**ontrol facility
  - Logging and database recovery.
  - Database access in a data sharing environment.
  - Generates JCL for various IMS Utilities
    - Database Recovery
    - Change Accumulation
    - Image Copy
    - Log Archive
    - Log Recovery

DBRC, the early days

- The IMS Backup and Recovery Feature
  - Supported the main IMS Utilities
    - Data Base Recovery
    - Change Accumulation
    - Image Copy

- DBRC Release 1 shipped December 1978 with IMS 1.1.4

Why DBRC?

- Originally, DBRC was an optional, separately priced feature of IMS

- These IMS environments require DBRC
  - Online: DB / TM, DCCTL, or DBCTL
  - Data sharing environments
  - Extended Recovery Facility (XRF)
  - Remote Site Recovery (RSR)
  - HALDB

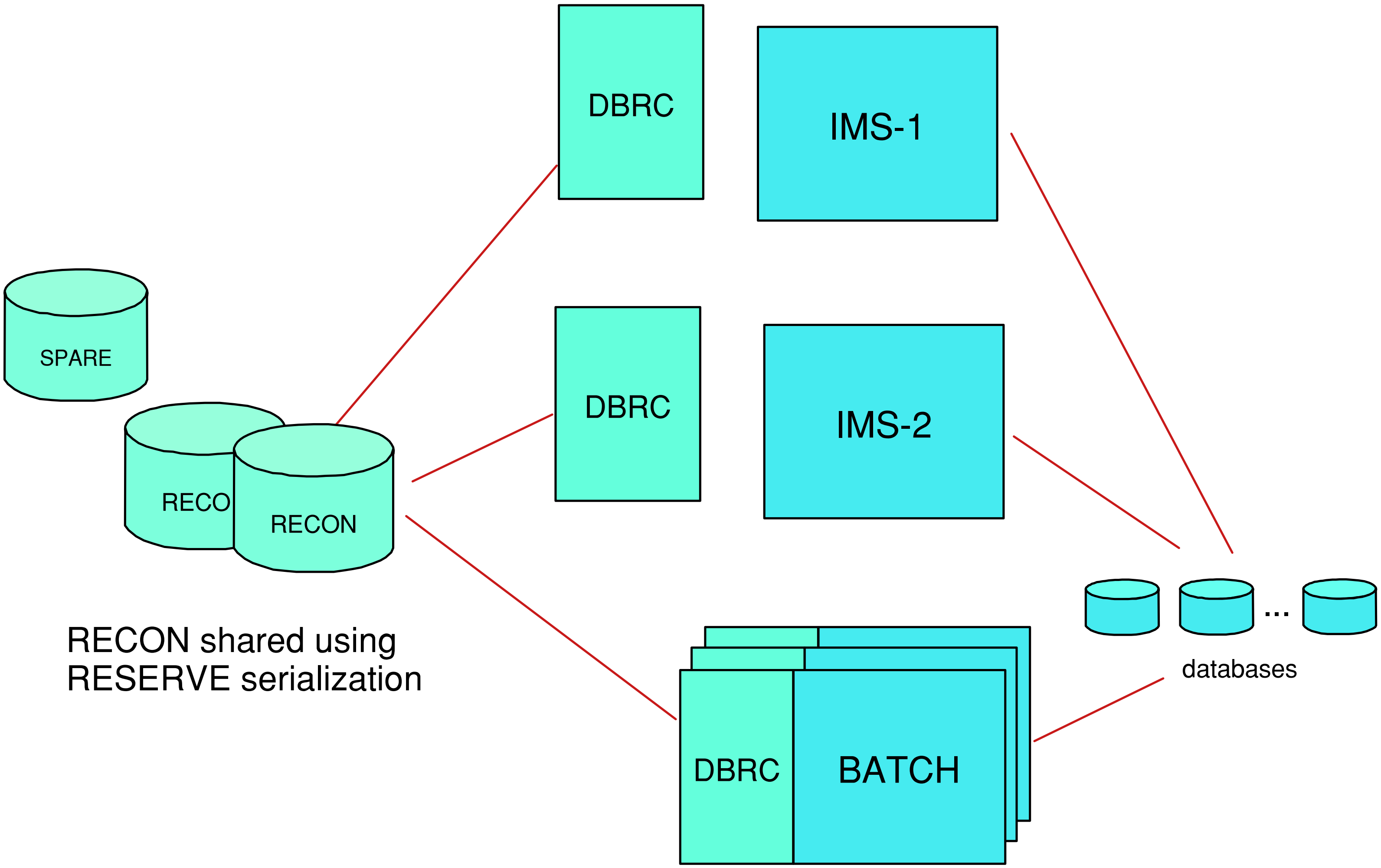
- DBRC should ALWAYS be used in order to simplify your recovery processes and protect your databases

DBRC

Includes:

- The **Re**covery **Con**trol (**RECON**) data sets
- Command support through the Database Recovery Control utility (DSPURX00)
- Application Programming Interface (V9)
- Skeletal Job Control Language (JCL) for various IMS recovery utilities

Online/Batch DBRC Address Space

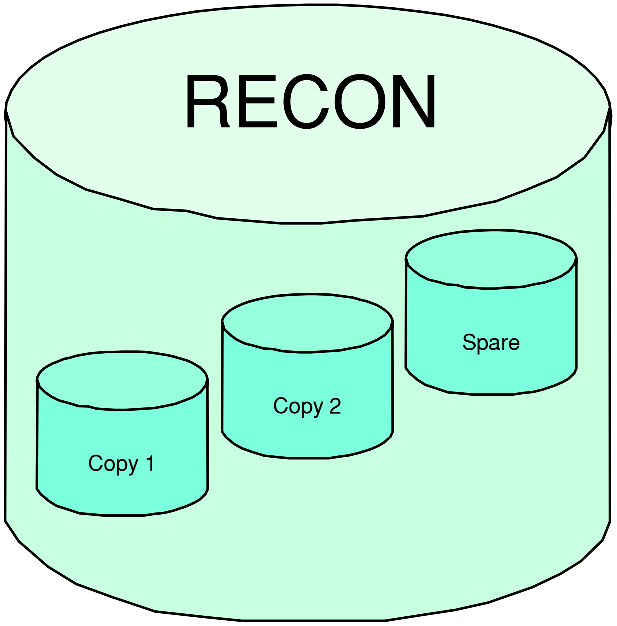


RECON shared using
RESERVE serialization

The RECON Data Set

The RECON Data Set

- DBRC's primary resource
  - Contains status information for IMS System and DB activity
    - Restart
    - DB backup and recovery support
  - Controls access and activity
    - IMS logs
    - IMS databases
    - Data sharing
- 3 RECON data sets are used to maintain dual-mode operation
  - 2 active RECONs and a spare
  - VSAM KSDSs



Creating the RECON with Access Method Services

```
//*     DELETE AND DEFINE RECON DATA SET
//DEFRCN  EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
  DELETE (IMSTESTL.IMS.RECON1 IMSTESTL.IMS.RECON2 IMSTESTL.IMS.RECON3)
  DEFINE CLUSTER (NAME(IMSTESTL.IMS.RECON1) -
       INDEXED KEYS(32 0) -
       VOL(USER01) CYL(50 25) RECSZ(128 32756) FSPC(20 20) -
       SHR(3 3) NONSPANNED)
DEFINE CLUSTER (NAME(IMSTESTL.IMS.RECON2) -
       INDEXED KEYS(32 0) -
       VOL(USER02) CYL(60 25) RECSZ(128 32756) FSPC(25 25) -
       SHR(3 3) NONSPANNED)
DEFINE CLUSTER (NAME(IMSTESTL.IMS.RECON3) -
       INDEXED KEYS(32 0) -
       VOL(USER03) CYL(70 25) RECSZ(128 32756) FSPC(30 30) -
       SHR(3 3) NONSPANNED)
/*
```

16M RECON Record Size (V8)

- Maximum size of 'logical' RECON record is 16 megabytes
  - Eliminates potential outages due to RECON records reaching maximum VSAM record size
    - PRILOG, DBDS, SUBSYS
- DBRC implements its own RECON record spanning
  - RECON records are segmented into multiple VSAM records

RECON Initialization

- Initialize the RECON datasets after defining them with AMS
- Issue the DBRC INIT.RECON command
  - DBRC Utility - DSPURX00
  - The first command to issue after defining the RECON
  - RECON must be empty
  - Example:

```
INIT.RECON SSID(IMS3)
```

RECON Record Contents

- The RECON record is comprised of 2 records
  - RECON Header record
    - Control system options
  - RECON Header Extension record
    - Controls RECON management
      - **COPY 1, COPY 2, SPARE**
- RECON Initialization Token
  - Used for DBRC Usage Indicator (DUI) processing
- Processing Options
  - FORCER, Timestamp Format Information, DMB#
- The LIST.RECON output also displays the status of the RECON data sets
  - DD Names, DSNs, DS status

RECON Record Example (V9)

```
 RECON
  RECOVERY CONTROL DATA SET, IMS V9R1
  DMB#=11                            INIT TOKEN=04190F1953352F
  NOFORCER  LOG DSN CHECK=CHECK17    STARTNEW=NO
  TAPE UNIT=3400      DASD UNIT=SYSDA     TRACEOFF   SSID=**NULL**
  LIST DLOG=YES                 CA/IC/LOG DATA SETS CATALOGED=NO
  MINIMUM VERSION = 7.1
  LOG RETENTION PERIOD=00.000 00:15:00.0
  COMMAND AUTH=NONE  HLQ=**NULL**
  SIZALERT DSNUM=15      VOLNUM=16     PERCENT= 10
  LOGALERT DSNUM=3       VOLNUM=16

  TIME STAMP INFORMATION:

    TIMEZIN = %SYS                -LABEL- -OFFSET-
                                   PDT      -07:00
                                   PST      -08:00

    OUTPUT FORMAT:  DEFAULT = LOCORG NONE   PUNC YY
                    CURRENT = LOCORG NONE   PUNC YY

IMSPLEX = ** NONE **

-DDNAME-      -STATUS-       -DATA SET NAME-
 RECON1        COPY1          IMSTESTL.IMS.RECON1
 RECON2        COPY2          IMSTESTL.IMS.RECON2
 RECON3        SPARE          IMSTESTL.IMS.RECON3
```

RECON Record Example (V7)

```
RECON
 RECOVERY CONTROL DATA SET, IMS/ESA V7R1        COEXISTENCE ENABLED
 DMB#=8                             INIT TOKEN=99060F2231139F
 NOFORCER  LOG DSN CHECK=CHECK17    STARTNEW=NO
 TAPE UNIT=3480      DASD UNIT=SYSDA     TRACEOFF   SSID=**NULL**
 LIST DLOG=NO                CA/IC/LOG DATA SETS CATALOGED=NO
 LOG RETENTION PERIOD=00.001 00:00:00.0
 SIZALERT DSNUM=15      VOLNUM=16     PERCENT= 95
 LOGALERT DSNUM=3       VOLNUM=16

 TIME STAMP INFORMATION:

   TIMEZIN = %SYS

   OUTPUT FORMAT:  DEFAULT = LOCORG NONE   PUNC YY
                   CURRENT = LOCORG NONE   PUNC YY

 -DDNAME-      -STATUS-       -DATA SET NAME-
  RECON1        COPY1          IMSTESTL.IMS.RECON1
  RECON2        COPY2          IMSTESTL.IMS.RECON2
  RECON3        UNAVAILABLE
THT
   -LOCAL START-   -OFFSET-
 0000.000 00:00:00.0 -08:00
```

RECON Data

How does information get recorded in the RECON?

- DB registration
  - INIT
- IMS processing
  - logging, DB update, utilities
- Other DBRC commands
  - CHANGE, NOTIFY, DELETE

RECON Records

- RECON (Header, Header Extension)
- PRILOG, SECLOG
- PRIOLD, SECOLD
- PRISLD, SECSLD
- LOGALL
- GSG
- SSYS
- BACKOUT
- DMB (V9)
- CAGRP
- CA
- DBDSGRP (DBGRP and RECOVGRP)
- DB
- DBDS
- ALLOC
- IMAGE
- REORG
- RECOV

DBRC Commands

- **BACKUP**
  - Make a copy of the RECON Data Set
- **CHANGE**
  - Modify the contents an existing RECON record
- **DELETE**
  - Delete an existing RECON record
- **GENJCL**
  - Generate JCL for a utility or function
- **INIT**
  - Initialize (create) a RECON record
- **LIST**
  - Produce formatted output of selected RECON records
- **NOTIFY**
  - Add information to the RECON that is normally written automatically
- **RESET**
  - RSR-only

DBRC Command Syntax

- Standard, command language syntax
  - Similar to TSO and Access Method Services
- Consists of
  - Verb
  - Modifier
  - Keyword parameters - required/optional, defaults

```
BACKUP.RECON
LIST.RECON STATUS
```

DB Registration

The INIT Command

- INIT commands with the following modifiers are used in the DB registration process:
  - DB
  - DBDS
  - ADS - Fast Path only
  - PART - HALDB only
  - IC
  - DBDSGRP
  - CAGRP
  - CA

INIT.DB Command

- Use the INIT.DB command to register a database with DBRC and to set the database sharing level desired (0,1,2,or 3)

- Three types of DBs
  - TYPEIMS, TYPEFP, TYPHALDB
    - HALDBs may also be registered using the *HALDB Partition Definition Utility*

- NONRECOV|<u>RECOVABL</u>

DB Record Contents

- Database Definition (DBD) name
- Status flags and counters
  - DB type
  - Backout Needed, Prohibit Authorization, IC needed, etc.
- Sharing status
  - Share Level
  - Global DMB number
  - IRLM ID
  - Authorized subsystems
- Recoverable status
  - Non-recoverable DEDB support added in IMS V8.
- RSR Information

IMS DB Record Example

```
DB
 DBD=DHVNTZ02                                     DMB#=2       TYPE=IMS
 SHARE LEVEL=1               GSGNAME=**NULL**     USID=0000000001
 AUTHORIZED USID=0000000000  RECEIVE USID=0000000000 HARD USID=0000000000
 RECEIVE NEEDED USID=0000000000
 DBRCVGRP=**NULL**
 FLAGS:                                  COUNTERS:
   BACKOUT NEEDED        =OFF              RECOVERY NEEDED COUNT   =0
   READ ONLY             =OFF              IMAGE COPY NEEDED COUNT =0
   PROHIBIT AUTHORIZATION=OFF              AUTHORIZED SUBSYSTEMS   =0
   RECOVERABLE           =YES              HELD AUTHORIZATION STATE=0
                                           EEQE COUNT              =0
   TRACKING SUSPENDED    =NO               RECEIVE REQUIRED COUNT  =0
   OFR REQUIRED          =NO
```

INIT.DBDS Command

- Use the INIT.DBDS command to register a DBDS or DEDB area
  - DBDLIB is required:
    - to verify the DBDS or DEDB area exists
    - to obtain the DBDS's data set identifier (DSID), its database organization (DBORG), and its dataset organization (DSORG)
  - Command parameters
    - IC GENMAX value
    - Recovery Period (RECOVPD)
    - names of your skeletal JCL members
    - IC REUSE/NOREUSE

- The INIT.DBDS command fails if you issue it while the DB is in use

- Not allowed for HALDBs

DBDS Record Contents

- DBD and DD names
- Data Set Name (DSN)
- Database type (e.g. IMS , HALDB, FP)
- DB and DS organization (e.g. DBORG=HDAM, DSORG=VSAM)
- Status flags and counters
  - IC NEEDED, RECOV NEEDED
  - Extented Error Queue Element (EEQE) count
- Image Copy information
  - REUSE or NOREUSE, GENMAX, IC AVAIL, IC USED
- Recovery information
  - RECOV NEEDED, RECOVPD
- Change Accumulation Group name
- Skeletal JCL execution member names
  - ICJCL, OICJCL, RECOVJCL, etc.

DBDS Record Example

```
DBDS
 DSN=DHVNTZ02.FKXXI02E                                         TYPE=IMS
 DBD=DHVNTZ02  DDN=HIDAM2   DSID=002 DBORG=HIDAM  DSORG=VSAM
 CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=0     DSSN=00000000
 NOREUSE         RECOVPD=0       OTHER DDN=**NULL**
 DEFLTJCL=**NULL**  ICJCL=ICJCL     OICJCL=OICJCL    RECOVJCL=RECOVJCL
 RECVJCL=ICRCVJCL
 FLAGS:                             COUNTERS:
   IC NEEDED      =OFF
   IC RECOMMENDED =ON
   RECOV NEEDED   =OFF
   RECEIVE NEEDED =OFF                EEQE COUNT              =0
```

HALDB Record Contents

- Master name
- Share level
- DB and DS organization
- Partition Selection Routine (optional)
- Change version number
- Current Partition ID
- Partition count
- Number of DSGs
- Status - recoverable, OLR capable
- GSG name (RSR)

HALDB Record Example

```
DB
 DBD=DBHDOJ01                        DMB#=2      CHANGE#=5     TYPE=HALDB
 SHARE LEVEL=0               GSGNAME=**NULL**
 PSNAME=**NULL**  DBORG=PHDAM    DSORG=OSAM    CURRENT PARTITION ID=00004
 FLAGS:                             COUNTERS:
   RECOVERABLE           =YES         PARTITIONS              =4
   ONLINE REORG CAPABLE  =YES         DATA SET GROUP MEMBERS  =1
```

INIT.PART Command

- Use INIT.PART command to register a HALDB Partition
  - DBRC creates the RECON HALDB partition structure
    - a PART record
    - the partition DB record
    - one or more DBDS records (determined by the DBD specification)
      - similar to a non-HALDB DBDS
    - TYPE=PART
  - Specify
    - names of skeletal JCL members
    - IC REUSE or NOREUSE
    - IC GENMAX value
- Command fails if the HALDB is being used by the HALDB Partition Definition utility

DB PART Record Contents

- Similar to a non-HALDB DB record, but with more partition related fields
- Partition High Key / String in character and hexidecimal forms
- Randomizer information
- Free Space information
- OSAM Block Size
- Current sharing status (i.e. authorized subsystems)

DB (PART) Record Example

```
DB
 DBD=PDHDOJA   MASTER DB=DBHDOJ01                CHANGE#=2     TYPE=PART
 USID=0000000002  AUTHORIZED USID=0000000002  HARD USID=0000000002
 RECEIVE USID=0000000002  RECEIVE NEEDED USID=0000000000
 DBRCVGRP=**NULL**
 DSN PREFIX=IMSTESTS.DBHDOJ01                          PARTITION ID=00001
 PREVIOUS PARTITION=**NULL**    NEXT PARTITION=PDHDOJB
 OLRIMSID=**NULL**    ACTIVE DBDS=A-J
 RANDOMIZER:
   NAME=DFSHDC20  ANCHOR=3    HIGH BLOCK#=3         BYTES=25
 FREE SPACE:
   FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0
 PARTITION HIGH KEY/STRING (CHAR):                (LENGTH=5   )
   J0200
 PARTITION HIGH KEY/STRING (HEX):
   D1F0F2F0F0404040404040404040404040404040404040404040404040404040

 OSAM BLOCK SIZE:
   A = 4096

 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
                                       EEQE COUNT              =0
   TRACKING SUSPENDED     =NO          RECEIVE REQUIRED COUNT  =0
   OFR REQUIRED           =NO
   PARTITION INIT NEEDED =NO
   OLREORG CURSOR ACTIVE =NO
   PARTITION DISABLED     =NO
   ONLINE REORG CAPABLE   =YES
```

DBDS (PART) Record Example

```
DBDS
 DSN=IMSTESTS.DBHDOJ01.A00001                                  TYPE=PART
 DBD=PDHDOJA   DDN=PDHDOJAA DSID=001 DBORG=HDAM   DSORG=OSAM
 CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=0     DSSN=00000004
 NOREUSE         RECOVPD=0       OTHER DDN=PDHDOJAM
 DEFLTJCL=**NULL**  ICJCL=PICJCL    OICJCL=POICJCL   RECOVJCL=PRECOJCL
 RECVJCL=PRECVJCL
 FLAGS:                             COUNTERS:
   IC NEEDED      =OFF
   IC RECOMMENDED =ON
   RECOV NEEDED   =OFF
   RECEIVE NEEDED =OFF                EEQE COUNT              =0
```

Fast Path DB Record Contents

- DEDB name
- Counters and status flags
  - Recovery needed, IC needed, EEQE, authorized areas
  - Non-recoverable (V8), prohibit auth
  - Authorization, IC needed, etc.
- Sharing status
  - Share Level
  - Global DMB number

```
DB
 DBD=DEDBDD01                                             DMB#=8        TYPE=FP
 SHARE LEVEL=1
 FLAGS:                                      COUNTERS:
                                               RECOVERY NEEDED COUNT    =0
                                               IMAGE COPY NEEDED COUNT  =0
   PROHIBIT AUTHORIZATION=OFF                  AUTHORIZED AREAS         =0
   RECOVERABLE           =YES                  EEQE COUNT               =0
```

FP DBDS Record Contents

- Combination of two RECON records
  - Area Auth
    - Fast Path DEDBs are authorized at the Area level
    - similar to DB
  - Area Recovery
    - similar to DBDS
    - also contains list of Area Data Sets (ADS)

FP DBDS Record Example

```
DBDS
 DBD=DEDBDD01  AREA=DD01AR0                                    TYPE=FP
 SHARE LEVEL=1             DSID=00001 DBORG=DEDB   DSORG=VSAM
 GSGNAME=IMSGSG1              USID=0000000003
 AUTHORIZED USID=0000000003  RECEIVE USID=0000000003 HARD USID=0000000003
 RECEIVE NEEDED USID=0000000000
 CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     DSSN=00000002
 NOREUSE         RECOVPD=0      NOVSO  PREOPEN    NOPRELOAD
 CFSTR1=**NULL**         CFSTR2=**NULL**          NOLKASID  NOMAS
 DEFLTJCL=**NULL**  ICJCL=ICJCL    RECVJCL=ICRCVJCL  RECOVJCL=RECOVJCL
 DBRCVGRP=**NULL**
 FLAGS:                              COUNTERS:
   PROHIBIT AUTHORIZATION=OFF          AUTHORIZED SUBSYSTEMS   =0
                                       HELD AUTHORIZATION STATE=0
   IC NEEDED             =OFF          ADS AVAIL #             =1
   RECOV NEEDED          =OFF          REGISTERED ADS #        =1
   DATABASE LEVEL TRACK  =YES          EEQE COUNT              =0
   RECEIVE NEEDED        =OFF
   OFR REQUIRED          =NO
   TRACKING SUSPENDED    =NO
   HSSP CIC IN PROGRESS  =NO

 ADS LIST:
                                                                 CREATE
   -ADS DDN--ADS DSN-                                    -STAT-  -RUNNING-
   DD01AR0  DD01AR0                                      AVAIL      NO
```

Group records

- Use the **INIT.DBDSGRP** command to define the following types of groups:
  - **DBDS Group**
    - DBDSs or DEDB areas
    - Used by GENJCL, LIST.DBDS, and LIST.HISTORY commands
  - **DB Group**
    - DL/I databases or DEDB areas
    - Used with /STA, /STO, /DBR DATAGROUP commands
  - **Recovery Group**
    - DL/I databases or DEDB areas
    - Used with Database Recovery Facility (DRF) recoveries
    - A DB/Area can only belong to one Recovery Group

Group Record Examples

```
DBGRP
 GRPNAME=DBGRP1                #MEMBERS=6    -DBD/AREA-
                                              DIVNTZ02
                                              DHVNTZ02
                                              DXVNTZ02
                                              DB21AR0
                                              DB21AR1
                                              DB21AR2

DBDSGRP
 GRPNAME=FJKGRP                #MEMBERS=5     -DBD-     -DDN/AREA-
                                              DIVNTZ02   DBHVSAM1
                                              DIVNTZ02   DBHVSAM2
                                              DHVNTZ02   HIDAM
                                              DHVNTZ02   HIDAM2
                                              DXVNTZ02   XDLBT04I

RECOVGRP
 GRPNAME=RCVGRP1               #MEMBERS=5     -DBD-      -AREA-
                                              DIVNTZ02
                                              DHVNTZ02
                                              DXVNTZ02
                                              DEDBJN21   DB21AR0
                                              DEDBJN21   DB21AR1
```

INIT.CAGRP Command

- Specifies the DBDSs you want to belong to a specific Change Accumulation Group
- Each of the DBDSs in the group must be registered in RECON
- A DBDS can only belong to one CA Group
- CA groups can be used anywhere DBDS groups are used
  - the opposite is not true

CAGRP Record Contents

- CA group name
- Status flags, counters
  - GRPMAX, REUSE|NOREUSE
- Name of the CA skeletal JCL for GENJCL
- Names of the DBDS members of the group

CAGRP Record Example

```
CAGRP
 GRPNAME=CAGRP1   GRPMAX=3    CA AVAIL=0    CA USED=1
 NOREUSE CAJCL=CAJCL          DEFLTJCL=**NULL**
                              #MEMBERS=4    -DBD-     -DDN-
                                            DEDBJN21  DB21AR1
                                            DEDBJN21  DB21AR3
                                            DEDBJN21  DB21AR6
                                            DEDBJN21  DB21AR7
```

INIT.IC Command

- Use the INIT.IC command to create an IMAGE copy record in the RECON to define the image copy dateset available for use during a subsequent execution of a supported Image Copy Utility.
  - DBDS must have the REUSE attribute
- Each INIT.IC command creates one IMAGE copy record
  - You can specify a duplicate image copy dataset

Available IMAGE Record Example

```
                             AVAILABLE DATA SET
                          DBD=DEDBJN21 DDN=DB21AR11

 IMAGE
* CREATE  = 04.190   09:13:09.7*

 IC1
  DSN=IMSVS.DEDBJN21.DB21AR11.IC.IC091307          FILE SEQ=0001
  UNIT=3400                          VOLS DEF=0000 VOLS USED=0000
```

IMS Interactions with DBRC

IMS interactions with DBRC

- IMS and DBRC need to communicate in a number of environments
  - Batch
  - Online
  - Utilities

DBRC Exit Processing (Initial)

- Allocate RECON1, RECON2 and RECON3

- RESERVE all 3 RECON data sets
  - in DDNAME sequence, or MDA entry sequence
  - QNAME is DSPURI01, RNAME is RECON DSN
    - In order to give onlines higher priority, batch jobs also enqueue on QNAME DSPURI02

- Open RECONs
  - Note spares

DBRC Exit Processing (Initial)...

- Read Header records, and Header Extensions

- Determine active pair
  - Restore duality if necessary, and possible

- Deallocate & release unused RECON data sets

- Perform partial update backouts if necessary

- Perform EXIT processing

- Release active RECONs

DBRC Exit Processing (subsequent)

- RESERVE active RECON data sets
  - in DDNAME sequence, or MDA entry sequence
  - QNAME is DSPURI01,RNAME is RECON DSN
    - QNAME DSPURI02 for batch

- Invalidate RECON buffers

- Read Header & Header Extensions

- Close and reopen RECONs if needed

DBRC Exit Processing...

- If active pair has changed, allocate other RECONs and determine active pair
  - Restore duality if necessary and possible
- Deallocate unused RECON data sets
- Perform partial update backouts if necessary
- Perform exit processing
- If ’TERM’ exit, close RECONs
- Release RECONs

IMS Batch/Online Interfaces

Batch/Online Interfaces

- Sign-on
- Log Open
- Authorization
- DB Open
- DB Update
- DB I/O Error
- Log Processing
- Termination

Sign-on Processing (normal)

- Normal Sign-on
  - Performed by DLI/DBB batch initialization
    - Subsystem name is the MVS Jobname
  - or by /NRE online processing
    - Subsystem name is the IMSID
    - Cold start updates the BACKOUT record (if one exists)
- Builds a SUBSYS record if one does not already exist for the subsystem
- Fails if a SUBSYS record already exists

Sign-on Processing (abnormal)

- Abnormal Sign-on
  - Recovery Start
    - Turns on ’Recovery Started’ bit if SUBSYS record already exists for the subsystem
    - fails if a SUBSYS record does not exist for the subsystem
  - Sign-on Recovery end
    - Releases database authorizations
    - When complete, environment looks like ’Sign-on normal’
  - Sequence issued by:
    - /ERE online processing
    - Batch Backout Utility

Subsystem (SSYS) Record Contents

- Subsystem name
- IRLM name
- Subsystem type
  - Batch
  - Online
  - Tracker
- Log open time stamp
- Status flags
- Authorized database names and status
  - the size of this part of the SSYS record varies
  - the number of databases or areas can grow large

SSYS Record Example

```
SSYS
 SSID=SYS3      LOG START=04.201 07:39:28.9
 SSTYPE=ONLINE  ABNORMAL TERM=OFF  RECOVERY STARTED=NO   BACKUP=NO
 TRACKED=NO     TRACKER TERM=OFF   SHARING COVERED DBS=NO
 IRLMID=**NULL**  IRLM STATUS=NORMAL        GSGNAME=IMSGSG1

 AUTHORIZED DATA BASES/AREAS=0      VERSION=9.1   XRF CAPABLE=NO
```

Log Open precessing (batch)

- Build PRILOG [& SECLOG] records
  - One per subsystem execution

- Build empty LOGALL record

- Update SUBSYS record with log start time

IMS TM & DBCTL Log Open processing

- Build PRIOLDS & SECOLDS for subsystem
  - if they do not exist
  - Note: There is one set of PRIOLDS-SECOLDS per subsystem
- Build PRILOG record
  - One per subsystem execution
- Build PRISLDS record
  - One per subsystem execution
- Build empty LOGALL (log allocation) record
- Update SUBSYS record with log start time

PRILOG Record Contents

- Start time, Stop time
- Subsystem name (i.e. SSID)
- Log Version
- Global Service Group Name (GSGNAME)
- PRILOG Token
- Number of data sets (i.e. #DSN)
- First Record ID
- Each data set entry contains the following
  - Dataset name
  - Start, Stop time
  - First, Last Dataset Log Sequence Number
  - Unit, File, Volume, and Checkpoint information

LOGALL Record Contents

- Start time of associated PRILOG
- Earliest allocation time stamp
- Number of Database datasets allocated
- DBD, DDN, and Alloc counts for each DBDS
  - This area of the LOGALL record varies in size depending on the number of DBDSs allocated.
  - Allocation count relates to the number of ALLOC records that exist in the RECON for a given DBDS for the log.

Subsystem Authorization Processing

- For Batch: At initialization time

- For Online TM system: At first schedule

- Unregistered databases - always granted
  - Unless ’FORCER’ has been specified

- Database treated as registered
  - if DBD, DD and DSN match RECON information

- Processing flow:
  - Check status flags in DB and DBDS records
  - Check current authorizations for compatibility
  - If all DB’s requests can be granted
    - update SUBSYS and DB records

Auth Example (SSYS)

```
SSYS
 SSID=SYS3       LOG START=04.201 07:39:28.9
 SSTYPE=ONLINE   ABNORMAL TERM=OFF  RECOVERY STARTED=NO   BACKUP=NO
 TRACKED=NO      TRACKER TERM=OFF   SHARING COVERED DBS=NO
 IRLMID=**NULL**   IRLM STATUS=NORMAL        GSGNAME=IMSGSG1

 AUTHORIZED DATA BASES/AREAS=1      VERSION=9.1   XRF CAPABLE=NO
                                                   ENCODED
   -DBD-         -AREA-    -LEVEL-  -ACCESS INTENT-  -STATE-
   DEDBDD01      DD01AR0      1        UPDATE           6
```

Auth Example (Area)

```
 DBDS
  DBD=DEDBDD01  AREA=DD01AR0                                    TYPE=FP
  SHARE LEVEL=1             DSID=001 DBORG=DEDB   DSORG=VSAM
  GSGNAME=IMSGSG1              USID=0000000003
  AUTHORIZED USID=0000000003  RECEIVE USID=0000000003 HARD USID=0000000003
  RECEIVE NEEDED USID=0000000000
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     DSSN=00000002
  NOREUSE         RECOVPD=0      NOVSO  PREOPEN    NOPRELOAD
  CFSTR1=**NULL**         CFSTR2=**NULL**          NOLKASID
  DEFLTJCL=**NULL**  ICJCL=ICJCL    RECVJCL=ICRCVJCL  RECOVJCL=RECOVJCL
  DBRCVGRP=**NULL**

  FLAGS:                             COUNTERS:
    PROHIBIT AUTHORIZATION=OFF         AUTHORIZED SUBSYSTEMS   =1
                                       HELD AUTHORIZATION STATE=6
    IC NEEDED             =OFF         ADS AVAIL #             =1
    RECOV NEEDED          =OFF         REGISTERED ADS #        =1
    DATABASE LEVEL TRACK  =YES         EEQE COUNT              =0
    RECEIVE NEEDED        =OFF
    OFR REQUIRED          =NO
    TRACKING SUSPENDED    =NO
    HSSP CIC IN PROGRESS  =NO

  ADS LIST:
                                                                  CREATE
    -ADS DDN--ADS DSN-                                    -STAT- -RUNNING-
    DD01AR0  DD01AR0                                      AVAIL     NO

ASSOCIATED SUBSYSTEM INFORMATION:
                               ENCODED
     -SSID-   -ACCESS INTENT-  -STATE-  -SS ROLE-
    SYS3         UPDATE           6      ACTIVE
```

Database Open Processing

- Performed at database open time
- Does not update RECONs
- Verifies that all data sets of a database are registered if the database is registered
- Performs Database Usage Intent (DUI) processing, with possible error messages:
  - **DFS0485W** RECOVERY DATA FOR [DATA BASE dbdname|AREA areaname] MISSING FROM THE RECON DATA SET
  - **DFS0486W** THE RECON DATA SET WILL NOT BE UPDATED FOR [DATA BASE dbdname|AREA areaname]
  - **DFS0487W** THE RECON DATA SET USED FOR [DATA BASE dbdname|AREA areaname] HAS CHANGED
- Return EEQEs

DBDS Update Processing

- Performed when first update to a DBDS occurs following DB authorization
- Checks to make sure database is registered by comparing DBD name, DD name, and DSN
- Build ALLOC (allocation) record
- Updates LOGALL record
- Returns data sharing information (DSSN)
- Returns RSR information Update Set Identifier (USID)

DBDS Update Example

```
ALLOC
 ALLOC   =04.196 09:30:57.0                 *  ALLOC LRID =0000000000000000
 DSSN=0000000004 USID=0000000005 START = 04.196 09:29:46.6



LOGALL
 START   = 04.196 09:29:46.6                 *
 EARLIEST ALLOC TIME = 04.196 09:30:57.0
 DBDS ALLOC=1                        -DBD-    -DDN-    -ALLOC-
                                     DBVHDJ05 CJVHDG1E  1
```

DATABASE I/O Error Processing

- Performed for read and write errors

- DB and DBDS records updated with EEQE (Extended Error Queue Element) information

- If write error - full function
  - "Recovery Needed" flag turned on in DBDS record
  - "Recovery Needed" counter incremented in DB record

- If severe error - fast path
  - "Recovery Needed" flag turned on in DBDS (area recov) record
  - "Recovery Needed" counter incremented in DB record

Backout Error Processing

- Invoked for
  - Dynamic backout failures
  - /ERE backout failures
  - /ERE NOBMP

- To prevent authorization prior to recovery
  - "Backout Needed" counter incremented in DB record
  - "Backout Needed" flag turned on in DB record

- BACKOUT record updated (created) with UOR (unit of recovery) information

- Not invoked for DLI/DBB batch abends

BACKOUT Record Example

```
BACKOUT
 SSID=SYS3      #UORS=2

   RECOVERY TOKEN=E2E8E2F3404040400000000300000002
   TIME=04.196 09:49:19.0                PSB=PLVAPZ12
               INFLT   BMP   COLDEND
     ASSOCIATED DATA BASES=3

                  BACKED   DYN BKOUT
     -DBD-        -OUT -   -FAILURE-
     DHVNTZ02      NO         NO
     DXVNTZ02      NO         NO
     DIVNTZ02      NO         NO

   RECOVERY TOKEN=E2E8E2F3404040400000000400000000
   TIME=04.196 09:49:19.1                PSB=PSBEJK05
               INFLT   BMP   COLDEND
     ASSOCIATED DATA BASES=3

                  BACKED   DYN BKOUT
     -DBD-        -OUT -   -FAILURE-
     DBOHIDK5      NO         NO
     DXVHIDK5      NO         NO
     DBVHDJ05      NO         NO
```

Log processing: Batch

- Updates PRILOG/SECLOG records
- Log EOV
  - Add new volume serial number to current (only) RLDS entry
  - Set EOV time stamp for prior volume in RLDS entry
- Log Close
  - Set EOV time stamp for last volume in RLDS entry
  - Set data set stop time for RLDS entry
  - Set subsystem stop time for PRILOG/SECLOG record

Log processing: TM & DBCTL

- Updates PRIOLDS/SECOLDS records

- OLDS switch
  - For the Current OLDS:
    - Set OLDS close time,
    - turn off "In Use" flag, and
    - set "Archive Needed" flag
  - For the new OLDS:
    - Set OLDS open time,
    - subsystem start time, and
    - turn on "In Use" flag

Log processing: TM & DBCTL ...

- To close an OLDS:
  - For the current OLDS:
    - Set OLDS close time,
    - turn off "In Use" flag, and
    - set "Archive Needed"

Archive Processing (OLDS)

- Create SECLOG/SECSLDS records for subsystem execution
  - if they do not exist
- Add SLDS entry to PRISLDS/SECSLDS records
  - using time stamps from the archived OLDS
- Add RLDS entry to PRILOG/SECLOG records
  - using time stamps from the archived OLDS
- If archive did not create a separate RLDS,
  - use SLDS information for RLDS entry
- Set status of archived OLDS data sets to "Available"

Archive Processing (OLDS)...

- Place "checkpoint ID prior to oldest unit of work"
  - in SLDS and RLDS data set entry
- If all OLDS for subsystem execution archived,
  - set subsystem stop time in PRILOG, SECLOG, PRISLDS, SECSLDS records

Archive Example - PRIOLD

```
PRIOLD
 SSID=SYS3            # DD ENTRIES=3
 EARLIEST CHECKPOINT = 04.196 08:51:30.4

 DDNAME=DFSOLP01   DSN=IMSTESTL.IMS01.OLDSP1
 START = 04.196 09:31:21.2                FIRST DS LSN= 00000000000005F8
 STOP  = 04.196 09:47:18.1                LAST  DS LSN= 00000000000007AB
 LOCK SEQUENCE# = 000000000000
 STATUS=ARC COMPLT                                FEOV=NO    AVAIL
 PRILOG TIME=04.196 09:29:46.6                ARCHIVE JOB NAME=JT085128
 VERSION=9.1

 DDNAME=DFSOLP02   DSN=IMSTESTL.IMS01.OLDSP2
 START = 04.196 08:51:28.5                FIRST DS LSN= 00000000000002BA
 STOP  = 04.196 08:53:43.2                LAST  DS LSN= 0000000000000444
 LOCK SEQUENCE# = 000000000000
 STATUS=ARC COMPLT                                FEOV=NO    AVAIL
 PRILOG TIME=04.196 08:51:28.5                ARCHIVE JOB NAME=JT085343
 VERSION=9.1

 DDNAME=DFSOLP00   DSN=IMSTESTL.IMS01.OLDSP0
 START = 04.196 08:53:43.2                FIRST DS LSN= 0000000000000445
 STOP  = 04.196 08:56:33.9                LAST  DS LSN= 00000000000006F3
 LOCK SEQUENCE# = 000000000000
 STATUS=ARC COMPLT                                FEOV=NO    AVAIL
 PRILOG TIME=04.196 08:51:28.5                ARCHIVE JOB NAME=JT085633
 VERSION=9.1
```

Archive Example - PRILOG

```
PRILOG                                       RECORD SIZE=     464
 START = 04.196 08:51:28.5              *    SSID=SYS3     VERSION=9.1
 STOP  = 04.196 08:56:33.9                   #DSN=2
 GSGNAME=IMSGSG1
 FIRST RECORD ID= 00000000000002BA       PRILOG TOKEN= 5
 EARLIEST CHECKPOINT = 04.196 08:51:30.4

 DSN=IMSVS.RLDSP.SYS3.D04196.T0851285.V01                 UNIT=SYSDA
 START = 04.196 08:51:28.5                FIRST DS LSN= 00000000000002BA
 STOP  = 04.196 08:53:43.2                LAST  DS LSN= 0000000000000444
 FILE SEQ=0001    #VOLUMES=0001

   VOLSER=000000 STOPTIME = 04.196 08:53:43.2
     CKPTCT=2    CHKPT ID = 04.196 08:51:30.4
     LOCK SEQUENCE#=  000000000000

 DSN=IMSVS.RLDSP.SYS3.D04196.T0853432.V02                 UNIT=SYSDA
 START = 04.196 08:53:43.2                FIRST DS LSN= 0000000000000445
 STOP  = 04.196 08:56:33.9                LAST  DS LSN= 00000000000006F3
 FILE SEQ=0001    #VOLUMES=0001

   VOLSER=000000 STOPTIME = 04.196 08:56:33.9
     CKPTCT=2    CHKPT ID = 04.196 08:56:33.6
     LOCK SEQUENCE#=  000000000000
```

Archive Processing (SLDS or RLDS)

- Search RECONs for SLDS or RLDS being archived
  - Match on data set name,file sequence number, and volume serial numbers
  - Look for "duplicates"
  - Can be time consuming
- Upon completion, update data set entry with
  - new data set name
  - file sequence number
  - unit type
  - volume serial numbers

PRILOG Compression

- Invoked
  - when new data set entry added to PRILOG, SECLOG, PRISLDS, or SECSLDS record, or
  - during DELETE.LOG INACTIVE command processed

- PRILOG compression is automatic
  - can be invoked manually with DELETE.LOG INACTIVE

- Deletes inactive data set entries from beginning of record
  - if compression threshold reached (V7)
    - 50% or 75% of maximum record size

PRILOG Compression...

- In IMS V8:
  - Attempted on every archive
  - If no DS entries compressed:

```
DSP1150I LOG RECORD(S) COULD NOT BE
         COMPRESSED, RECORD TIME = timestamp
         reason type = timestamp
```

    - Reason types:
      - EARLIEST ALLOC TIME
      - LOG RETENTION TIME
      - EARLIEST CHECK POINT

Compressed PRILOG Example

```
PRILOG                                       RECORD SIZE=     744
 START = 04.196 09:29:46.6              *    SSID=SYS3     VERSION=9.1
 STOP  = 00.000 00:00:00.0                   #DSN=4
 GSGNAME=IMSGSG1
 FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 2
 EARLIEST CHECKPOINT = 04.196 09:31:14.9

 DSN=**** COMPRESSED DATA SET ****                        UNIT=
 START = 04.196 09:29:46.6                FIRST DS LSN= 0000000000000001
 STOP  = 04.196 09:30:28.6                LAST  DS LSN= 000000000000030A
 FILE SEQ=0000    #VOLUMES=0000

 DSN=IMSVS.RLDSP.SYS3.D04196.T0930286.V01                 UNIT=SYSDA
 START = 04.196 09:30:28.6                FIRST DS LSN= 000000000000030B
 STOP  = 04.196 09:31:00.8                LAST  DS LSN= 00000000000004D9
 FILE SEQ=0001    #VOLUMES=0001

   VOLSER=000000 STOPTIME = 04.196 09:31:00.8
     CKPTCT=1    CHKPT ID = 04.196 09:30:28.6
     LOCK SEQUENCE#=  000000000000
  .
  .
  .
```

/DBRECOVER Processing

- Database data set update "span" terminated
  - ALLOC record updated with deallocation time (also occurs when database is closed)
- Database authorization released
  - DB and SUBSYS records updated
- "Prohibit Further Authorizations" flag in DB record
  - SET if /DBR GLOBAL
    - Not set if NOPFA specified

ALLOC Record Example (/DBR)

```
ALLOC
 ALLOC   =04.196 09:30:57.0              *  ALLOC LRID =0000000000000000
 DSSN=0000000004 USID=0000000005 START = 04.196 09:29:46.6
 DEALLOC =04.196 09:31:21.1               DEALLOC LRID =0000000000000000
```

Normal Subsystem Termination

- Sign-Off Normal
- Database Authorizations released
- DB Records updated
- SUBSYS record is deleted

Abnormal Subsystem Termination

- Sign-Off Abnormal (ESTAE)
- No Action (in case of MVS failure)
- Authorizations released
  - for databases that have NOT been updated
- SUBSYS record may be deleted,
  - .... ONLY if no databases* were updated

    * - registered or not registered databases
- SUBSYS record will remain
  - if <u>ANY</u> database(s) have been updated

Abnormally Terminated SSYS Example

```
SSYS
 SSID=SYS3      LOG START=04.201 07:39:28.9
 SSTYPE=ONLINE  ABNORMAL TERM=ON   RECOVERY STARTED=NO   BACKUP=NO
 TRACKED=NO     TRACKER TERM=OFF   SHARING COVERED DBS=NO
 IRLMID=**NULL**  IRLM STATUS=NORMAL        GSGNAME=IMSGSG1

 AUTHORIZED DATA BASES/AREAS=1      VERSION=9.1   XRF CAPABLE=NO
                                                   ENCODED
   -DBD-        -AREA-   -LEVEL-  -ACCESS INTENT-  -STATE-
   DEDBDD01     DD01AR0     1        UPDATE           6
```

Utility Interfaces

Utility Interfaces

- Unload/Scan/Reload/Prefix Update
- Image Copy
- Change Accumulation
- Batch Backout
- DB Recovery
- DBDSGRP Considerations

DB Unload and Scan Utilities

- Participate in database level sharing
  - ACCESS = RD
  - "IC Needed" and "Prohibit Authorizations" flags are ignored
  - "Recovery Needed" flag will fail authorization

- Database authorized as needed

- RECONs are not updated
  - These utilities do not alter the database

DB Reload Utilities

- HISAM Reload and HD Reload

- Participates in database level sharing
  - ACCESS = EX
  - "IC Needed" and "Prohibit Authorizations" flags are ignored

- Create REORG record

```
REORG
 RUN      = 04.194 11:03:02.2             *    USID = 0000000001

REORG
 RUN      = 04.194 11:12:58.8             *    USID = 0000000004
 STOP     = 04.194 11:13:06.2               ONLINE   RECOV = NO
```

Reload Utility Differences

- Authorization
  - HD occurs at initialization
  - HS occurs as required

- "IC Needed" Flag
  - HD always sets
  - HS sets if IC REUSE

- IC record
  - HD never creates
  - HS creates if IC NOREUSE

Prefix update

- Participates in database level sharing
  - ACCESS = EX
  - Authorization obtained as required
  - "IC Needed" and "Prohibit Authorizations" flags ignored
- If no logging, "IC Needed" flag is turned on
- If logging, acts like normal batch update job
  - ALLOC, PRILOG, and LOGALL records created

Image Copy processing

- GENJCL
- Sign-On
- Validation
- Authorization
- Data set processing
- Data set completion
- Sign-Off

Image Copy GENJCL & Validation

- Primary value of GENJCL is with REUSE option (pre-defined ICs)
  - If "available" ICs exist, select first available
  - If no "available" ICs exist, select oldest IC for reuse
  - Might be useful with DBDSGRPs
- Validation only effective with REUSE option
  - Verifies that output data set is the same that GENJCL would have selected

Image Copy Authorization

- Participates in database level sharing

- Authorization obtained as required
  - ACCESS = RD (RO for CIC)
  - "IC Needed" and "Prohibit Authorization" flags are ignored
  - Authorization will fail if "Recovery Needed" flag is on

Image Copy Completion

- IC record created
  - If REUSE and "available" IC used, IC record describing "available" data set is deleted
- If GENMAX is exceeded and RECOVPD exceeded
  - delete oldest IC record

IMAGE Record Deletion

- REORG and RECOV records created prior to oldest IC record are deleted

- ALLOC records created prior to oldest IC record and not needed for recovery are deleted (OLIC and CIC implications)

- ALLOC records created prior to oldest IC and needed for recovery are updated
  - ALLOC timestamp set to IC timestamp
  - START timestamp set to starting log volume (PRILOG compression implications)
  - If ALLOC records are deleted, the associated LOGALL records are updated

IMAGE Record Example

```
IMAGE
 RUN     = 04.196 09:31:42.8              *  RECORD COUNT =33
 STOP    = 04.196 09:31:43.0                 CONCUR     USID=0000000005

IC1
 DSN=IMSVS.DBVHDJ05.CJVHDG1E.IC.IC093140          FILE SEQ=0001
 UNIT=SYSDA                         VOLS DEF=0001 VOLS USED=0001
                                    VOLSER=222222
```

Change Accumulation processing

- GENJCL
- Validation
- Completion

Change Accumulation GENJCL

- If REUSE option
  - Set first "available" CA as new accum file
  - If no "available" CAs exist, select oldest CA for reuse
- Select latest valid CA for old accum file
- Select all needed log volumes (next foil)
- Create DB0 control cards for all members of CAGRP
  - Purge time stamp = latest valid image copy time stamp
    - Will be earlier for CIC

Change Accumulation Log Selection

- For each database data set in the CAGRP, select all log volumes that
  - Contain change records for the DBDS (ALLOC records) and
  - Are available (log volume has stop time in PRILOG record) and
  - Have a stop time greater than the computed purge time and
  - Have not been processed in the old accum file and
  - Do not follow an archiving "gap"
- Merge resulting list of log volumes and order in log volume start sequence
- Log selection ignores impacts of time stamp recoveries

Change Accumulation Completion

- CA record created
  - If REUSE and "available" CA used
    - CA record describing "available" data set is deleted
- If GRPMAX exceeded, delete oldest CA

CA Record Example

```
CA
 DSN=IMSVS.CAGRP1.CA.CA165501                     FILE SEQ=1
 CAGRP=CAGRP1       STOP    = 04.196 09:24:54.6              *
                    UNIT=SYSDA      VOLS DEF=1    VOLS USED=1
                                    VOLSER=222222
 RUN     = 04.196 08:55:39.9
  DBD=DEDBJN21 DDN=DB21AR1  PURGETIME = 04.196 09:22:45.2
    CHANGES ACCUMULATED=YES COMPLETE CA=YES INDOUBT EEQES=NO
    LSN    = 000000000000      DSSN   = 0000000001
    LRID   = 00000000000003E4  USID   = 0000000002
  DBD=DEDBJN21 DDN=DB21AR3  PURGETIME = 04.196 09:22:46.0
    CHANGES ACCUMULATED=YES COMPLETE CA=YES INDOUBT EEQES=NO
    LSN    = 000000000000      DSSN   = 0000000001
    LRID   = 00000000000003FE  USID   = 0000000002
  DBD=DEDBJN21 DDN=DB21AR6  PURGETIME = 04.196 09:22:47.2
    CHANGES ACCUMULATED=YES COMPLETE CA=YES INDOUBT EEQES=NO
    LSN    = 000000000000      DSSN   = 0000000001
    LRID   = 0000000000000418  USID   = 0000000002
  DBD=DEDBJN21 DDN=DB21AR7  PURGETIME = 04.196 09:22:47.7
    CHANGES ACCUMULATED=YES COMPLETE CA=YES INDOUBT EEQES=NO
    LSN    = 000000000000      DSSN   = 0000000001
    LRID   = 000000000000042D  USID   = 0000000002
```

Batch backout processing

- DBRC validates input log to ensure that
  - For Batch:
    - log is last non-backout generated log for the subsystem
    - no "volume gaps" exist
  - For Online
    - no "volume gaps" exist
- DBRC returns UOWs that need backing out

Batch backout processing...

- If COLDSTART or ACTIVE,
  - batch backout will return all UOWs that need backing out to DBRC
  - DBRC will add UOWs to the BACKOUT record if they do not already exist

- Following successful backout,
  - UOWs will be removed from backout record
  - if no more UOWs
    - backout record will be deleted
  - Backout counter in DB record will be decremented
    - If all counter in record are zero, Backout Needed flag will be turned off

Database Recovery Utility Processing

- GENJCL
- Sign-on
- Validation
- Authorization
- Recovery Processing
- Sign-off

DB Recovery Authorization & Termination

- Participates in database level sharing
  - ACCESS = EX

- Special authorization logic
  - All currently authorized subsystems must be marked as abnormally terminated

- List of valid USIDs passed at execution time

- "Recovery Needed" flags and counters set at beginning of execution for full function

- "Recovery Needed" flags and counters must already be set at beginning of execution for Fast Path

- "Recovery Needed" flags and counters reset at end of execution

Database Recovery (Full)

- GENJCL.RECOV DBD(name) DDN(name)
- Determine "gaps" caused by prior time stamp recoveries
- Select latest valid IC that does not include changes from a "gap"
- Select latest valid CA that
  - Has same purge time stamp as selected IC
  - Does not include changes associated with a "gap"
  - Does not span a REORG
  - Contains changes

Database Recovery (Full)...

- Select log volumes that
  - Contain changes (updates)
  - Have a stop time greater than the IC "purge" time
  - Not included in selected CA
  - Not created in a "gap"

- If logs cross a REORG boundary,
  - fail request

- Order logs in EOV time sequence

- If logs need merging,
  - then fail with "Merged Needed" message

Time Stamp Recovery

- Used to recover a database data set to an earlier state
  - ►Cannot ensure application or multi-database system integrity
  - ►Is sometimes used as a substitute for application error recovery

- GENJCL.RECOV DBD(name) DDN(name) RCVTIME(timestamp)

- Determine "gaps" caused by prior time stamp recoveries

- Validate RCVTIME
  - ►Must not be within "span" of an ALLOC record

Time Stamp Recovery...

- Select latest valid IC that
  - Does not include changes from a "gap"
  - Run time is less than or equal to RCVTIME
  - Stop time (OLIC) is less than or equal to RCVTIME

- Select latest valid CA that
  - Has same purge time stamp as selected IC
  - Does not include changes associated with a "gap"
  - Does not span a REORG
  - Contains changes within recovery window and
  - Contains no changes beyond that window

Time Stamp Recovery...

- Select log volumes that
  - Contain changes within recovery window
  - Have a stop time greater than the IC "purge" time
  - Not included in selected CA
  - Not created in a "gap"
- If logs cross a REORG boundary,
  - fail request
- Order logs in EOV time sequence
- If logs need merging,
  - then fail with "Merge Needed" message

Recovery Using Current DBDS

- Used normally with non-standard backups
- Requires that user restore DBDS and tell DBRC about restore
  - NOTIFY.RECOV DBD(name) DDN(name) RCVTIME(uic-time)
- GENJCL.RECOV DBD(name) DDN(name) USEDBDS
- Validate request for USEDBDS
  - If ALLOC records were created after latest recovery run time, fail request
  - If IC records exist after latest RCVTIME, fail request

RECOV Record Contents

- Time when utility was run

- Time to which the DBDS was recovered
  - RECOV TO= timestamp

- Update Set IDentifier (USID)
  - RUN USID=
  - RECOV TO USID=

```
RECOV
 RUN      = 04.196 09:18:47.9              *  RUN USID      = 0000000001
```

Non-recoverable DB Considerations

- Registration
  - INIT.DB ... NONRECOV
- Operation
  - No ALLOC records created
- Recovery
  - GENJCL.RECOV DBD(name) DDN(name) RESTORE

DBDSGRP Considerations

- Used with GENJCL.IC, GENJCL.OIC, GENJCL.RECOV, and GENJCL.USER commands
  - Result is equivalent to issuing multiple identical GENJCL commands
    - for each member of the DBDSGRP
  - DBD is an implied group that contains all the DBDS in the DBD
  - CAGRP is treated as a valid DBDSGRP

- Used with LIST.DBDS and LIST.HISTORY commands
  - Same comments as above

Other DBRC Commands

The CHANGE Commands

- Use the CHANGE command with these Modifiers to alter an existing RECON record.
  - **ADS** - Area Dataset
  - **BKOUT**
  - **RECON**
    - CATDS, TRACEON or TRACEOFF,LOGALERT, SIZALERT
    - LISTDL|NOLISTDL
    - UPGRADE (introduced in V7)
  - **DB**
  - **DBDS**
  - **DBDSGRP**
  - **CA** - Change Accumulation
    - INVALID
  - **CAGRP**
  - **IC or UIC** - Image Copy, User IC
  - **PART** - V7/V8 SPEs
  - **PRILOG (SECLOG)**
    - ERROR
  - **SUBSYS**

The NOTIFY Commands

- Use the NOTIFYcommand with these Modifiers to record an event in RECON:
  - **PRILOG, SECLOG**
  - **ALLOC**
  - **BKOUT**
  - **CA**
  - **IC, UIC**
  - **RECOV**
  - **REORG**
  - **SUBSYS**

- Not used in normal operation
  - Used to restore deleted records
  - Highly discouraged

GENJCL Command

- Use the GENJCL command to generate the JCL and utility control statements to execute these IMS utilities
  - GENJCL.ARCHIVE
    - Log Archive
  - GENJCL.CA
    - CA utility
  - GENJCL.CLOSE
    - Log Recovery utility
  - GENJCL.IC or OIC
    - Image Copy or Online Image Copy utility
  - GENJCL.RECOV
    - Database Recovery utility
  - GENJCL.RECEIVE
    - Database Recovery utility (RSR)

- Use the GENJCL.USER command to generate JCL or any kind of user determined output

Skeletal JCL

- DBRC provides the following Skeletal JCL execution Partitioned Data Set (PDS) members and uses them as a model for performing keyword substitution and JCL generation.
  - ARCHJCL - for the IMS Log Archive Utility (DFSUARC0)
  - CAJCL - for the IMS Database Change Accumulation utility (DFSUCUM0)
  - LOGCLJCL - for the IMS Log Recovery utility (DFSULTR0) to close a log
  - ICJCL - for the IMS Database Image Copy utility (DFSUDMP0)
  - OICJCL - for the IMS Online Image Copy utility (DFSUICP0)
  - ICRCVJCL - for GENJCL.RECEIVE to generate JCL to run the IMS Database Recovery utility for image copy receive
  - RECOVJCL - for the IMS Database Recovery utility (DFSURDB0).
  - JOBJCL - single JCL statement you modify with your installation's requirements
  - DSPUPJCL also provided
    - User JCL to rebuild the Index and/or ILE data sets for a HALDB Partition
  - User Options:
    - modify supplied skeletal member slightly
    - create own skeletal member

The LIST Command

- Use these LIST command modifiers to display existing RECON records.
  - BKOUT - Backout
  - CAGRP - CA Group
  - DB - Database
  - DBDS - Database dataset or data entry database
  - DBDSGRP - DBDS Group
  - GSG - Global Service Group
  - HISTORY - activity for DBDSs or areas
  - LOG - RLDS/SLDS, OLDS
  - SUBSYS - subsystem
  - RECON - full contents of the RECON or its STATUS

LIST.RECON Command

- LIST.RECON
  - Lists the contents of the entire RECON
  - Generally creates lots of output
  - DBRC reserves the RECON for the duration of the command - may take a long time

- LIST.RECON STATUS
  - the RECON record may contain all you need
  - generates less than a page of output
  - quick
  - the IMS MTO can issue via the /RM command

    /RML DBRC='RECON STATUS'.

DBRC Application Programming Interface (V9)

API Summary

- DBRC API
  - Allows users to write programs to read data from the RECON
    - Provides API to return information from all records
- Benefits
  - Supported method for retrieving data
    - Easier than parsing the output of LIST.RECON commands
  - Release independent
    - Future releases will not require modifications to programs
  - Users do not need to understand the segmenting of records

DBRC API - Highlights

- General-use assembler macro interface

- Functions provided:

  - RECON query

  - Supporting functions:
    - Start the API environment
    - Release buffer storage
    - Stop the API environment

Misc. RECON Topics

Misc. RECON Topics

- Recommendations
- Maintenance
- Security

Recommendations

- To avoid both active RECONs filling up at the same time
  - Make COPY2 larger than COPY1
  - Spare must be larger than the active RECONs
- To avoid CI/CA splits on both RECONs at the same time
  - Use different free space sizes
- Record size
  - V7 - as big as necessary
  - V8 - 16MB
    - NONSPANNED, record size less than CI size

Recommendations...

- Access to RECON is serialized using the z/OS RESERVE macro service (DSPURI01)
  - ►Batch jobs serialize on another resource first (DSPURI02) keeping the onlines from being locked out when a series of batch jobs are submitted.

- GRS Star environment
  - ►Convert RESERVE to a global ENQUEUE

- GRS Ring environment
  - ►Don't convert the RESERVE

RECON Maintenance

- RECON Reorganization
  - CHANGE.RECON REPLACE(RECON*n*)
- RECON Cleanup
  - DELETE.LOG INACTIVE

The BACKUP Command

- Use the BACKUP command to create a backup copy of the RECON.

```
BACKUP.RECON RECON1 | RECON2 | BOTH
```

- The command invokes IDCAMS REPRO
  - **Note**: Backup to a sequential data set has a 32K record size restriction

Automatic RECON Loss Notification (V8)

- RECON reconfiguration with previous IMS Releases
  - When a DBRC instance detects bad RECON, it begins the reconfiguration process
    - Copies good RECON to spare
    - Deallocates the bad RECON
  - Bad RECON must be deleted and redefined to create "new" spare
    - Deletion requires deallocation of "bad" RECON by each subsystem (i.e. DBRC instance)
      - Requires each DBRC to access RECON
- Automatic RECON Loss Notification (ARLN)
  - Option in IMS V8 to make RECON reconfiguration with other IMS subsystems automatic
  - DBRC instance beginning reconfiguration notifies other DBRCs via SCI
  - Other DBRCs invoke reconfiguration process immediately
  - Eliminates wait for next access to RECONs

Security

- DBRC Command Authorization
  - Allows the installation to control the use of DBRC commands
    - Commands can be authorized at the "command verb" level, the "resource type" level or the "resource" level
      - DELETE
      - CHANGE.DB
      - CHANGE.DB DBD(AAA)
  - Supported Environments:
    - DBRC Utility (DSPURX00)
    - HALDB Partition Definition Utility
    - Online DBRC commands (/RMxxxxxx)
- DBRC RECON Security White Paper

Summary

- DBRC - background
- The RECON
- Database registration
- IMS calls to DBRC
- Other DBRC commands
- DBRC API
- Misc. RECON topics

Appendix

Further Information

- IMS website: http://www-3.ibm.com/software/data/ims/
  - IMS Publications Library
    - DBRC Guide and Reference
    - IMS Release Planning Guide
  - Presentations
  - Redbooks
    - Database Recovery Control (DBRC) Examples and Usage Hints (SG24-3333-01)
- DBRC in Practice (Peter Armstrong)
- Follow the IMS Discussion forum (IMS-L)

RECON Record Types: Table 1

Common Name	Type	DSECT	List Name	Comment
RECON Header	X'01'	DSPRCNRC	RECON	
RECON Header Extension	X'01'	DSPRCR1	n/a	Not listed.
RECON Audit Record	X'02'	DSPMUPHD	n/a	Not listed; rarely seen.
PRILOG	X'05'	DSPLOGRC	PRILOG	
Interim PRILOG	X'06'	DSPLOGRC	IPRI	Seldom seen
LOGALL	X'07'	DSPLGARC	LOGALL	
SECLOG	X'09'	DSPLOGRC	SECLOG	
Interim SECLOG	X'0A'	DSPLOGRC	ISEC	Seldom seen
Change Accum Group	X'0F'	DSPCAGRC	CAGRP	
Change Accum Execution	X'11'	DSPCHGRC	CA	

RECON Record Types: Table 2

Common Name	Type	DSECT	List Name	Comment
Available CA	X'51'	DSPCHGRC	CA	
DBDS Group	X'16'	DSPDGRC	DBDSGRP, DBGRP, RECOVGRP	
Data Base Header	X'18'	DSPDBHRC	DB	
DB Partition	X'19'	DSPPTNRC	DB	Listed in combination with the Part DB rcd
Data Base Data Set	X'20'	DSPDSHRC	DBDS	
Area Recov	X'20'	DSPDSHRC	DBDS	Listed in combination with the Area Auth
Area Auth	X'21'	DSPDBHRC	DBDS	Listed in combination with the Area Recov
ALLOC	X'28'	DSPALLRC	ALLOC	
Image Copy	X'2D'	DSPIMGRC	IMAGE	
Reorg	X'32'	DSPRRGRC	REORG	

RECON Record Types: Table 3

Common Name	Type	DSECT	List Name	Comment
Recovery	X'37'	DSPRCVRC	RECOV	
Available IC	X'6D'	DSPIMGRC	IMAGE	
Backout	X'35'	DSPBKORC	BACKOUT	
Global Service Group	X'3A'	DSPGSGRC	GSG	
Tracking Subsystem	X'3E'	DSPSSRC	SSYS	RSR
Subsystem	X'3F'	DSPSSRC	SSYS	
Primary SLDS	X'43'	DSPSLDRC	PRISLD	
Tracking Primary SLDS	X'44'	DSPSLDRC	PRITSLD	RSR
Interim PRISLD	X'45'	DSPSLDRC	IPRISL	Seldom seen
Interim Tracking Primary SLDS	X'46'	DSPSLDRC	IPRITSLD	Seldom seen (RSR)

RECON Record Types: Table 4

Common Name	Type	DSECT	List Name	Comment
Secondary SLDS	X'47'	DSPSLDRC	SECSLD	
Tracking Secondary SLDS	X'48'	DSPSLDRC	SECTSLD	Seldom seen (RSR)
Interim Secondary SLDS	X'49'	DSPSLDRC	ISECSL	Seldom seen
Interim Tracking Secondary SLDS	X'50'	DSPSLDRC	ISECTSLD	Seldom seen (RSR)
Primary OLDS	X'53'	DSPOLDRC	PRIOLD	
Interim Primary OLDS	X'55'	DSPOLDRC	IPRIOL	Seldom seen
Secondary OLDS	X'57'	DSPOLDRC	SECOLD	
Interim Primary OLDS	X'59'	DSPOLDRC	ISECOL	Seldom seen
RECON DMB Table	X'03'	DSPRDTRC	N/A	Only used if Global DMB number wraps

RETURN TO INDEX