

S29

DBRC Friend OR Foe?

Rick Long

Senior IT Specialist, Silicon Valley Laboratory
(Australian Branch)

ricklong@au1.ibm.com



Miami Beach, FL

October 22-25, 2001

➤ Rick Long

▶ **IMS Development BI tools (remotely)**

- IMS Data Propagator
- IMS Data Refresher

▶ **ITSO**

- Redbooks
- IMS Specialist

▶ **IBM and life before IBM**

- IMS Systems Programmer
- Database Administrator
- Application Programmer

➤ **ricklong@au1.ibm.com**

- ▶ **I watch the IMS-L@lists.missouri.edu forum**

Friend?



The world depends on it

-
- **Its Sole Purpose is to ensure database integrity**
 - **Reduces operational/human errors**
 - **Allows a data sharing environment**

Foe?



The world depends on it

- **Enforcement of procedural rules**
 - ▶ Forces the order of some processes
 - ▶ Forces the sequence/inclusion of some events

- **Changes to operational procedures**

- **Changes to recovery strategy**

- **Differences in test system environment**



The world depends on it

What is DBRC?

- **In its simplest form it is those IMS functions which provide database integrity**
 - ▶ Database authorization processing
 - ▶ RECON definitions and usage
 - ▶ GENJCL functions for IMS recovery utilities



The world depends on it

Related Functions

- **Functions not part of DBRC which play an integral part of data integrity**
 - ▶ **IMS logging**
 - ▶ **IMS restart/checkpoint restart**
 - ▶ **Dynamic backout**
 - ▶ **Database utilities**
 - ▶ **Database locking**
 - ▶ **Remote Site Recovery (RSR)**



Where is DBRC Required?

- **In IMS online environments**
 - ▶ Database usage is still optional

- **When databases are used in a data sharing environment database must be registered .**

- **When IMS Data Propagator (DPropNR) is used to propagate changes to DB2 tables (IMS Data Propagator).**

- **When Remote Site Recovery (RSR) is used for tracking changes to databases at a remote site.**

- **When using IMS Online Recovery Service for data base recovery operations.**

What Does DBRC Provide?

- **Database integrity by controlling access via database authorization processing**
 - ▶ **Controls concurrent updates in a data sharing environment**
 - ensure data sharing rules are followed
 - ▶ **Ensures update procedures have log datasets**
 - IEFORDER DD card required (log file for DLIBATCH jobs)
 - Can't be DD DUMMY
 - ▶ **Ensures operational procedures are followed**
 - Image copy needed
 - **Recovery needed**
 - **Backout needed**



The world depends on it

What Does DBRC Provide?

- **Creates valid inputs to database recovery utilities**
 - ▶ Database recovery
 - ▶ Change accumulation
 - ▶ Image copy

- **Keeps historical record of update allocations**
 - ▶ Which subsystems actually update the databases
 - ▶ Can be used to identify when DB is available (SLA targets)

- **Groups databases into recovery groups**
 - ▶ Ensure a entire group of DB's is recovered together



The world depends on it

Where Do I See DBRC?

- **Database allocation/OPEN**
- **Subsystem startup/termination**
- **IMS emergency restart**
- **IMS OLDS switching**
- **Dynamic backout failure**
- **Database I/O error**
- **Database recovery/image copy/reorg**

Database Authorization Processing



The world depends on it

At database allocation time, IMS will check the status of the database in the RECON and either grant or reject the allocation, thus *authorizing* or *not authorizing* the database for use by this subsystem.

Basic Question answered by database authorization processing for each database to be opened is

➤ **Considering**

- ▶ The RECON status flags for this database
- ▶ The HELD AUTHORIZATIONS of subsystems already using this database (who else has authorized the database and at what level)
- ▶ The ACCESS INTENT of the new subsystem

➤ **Can this database be authorized to the new subsystem while maintaining database integrity**



Database Authorization Processing

➤ Status flags/counters

- ▶ IC needed
 - Reorg
- ▶ Backout needed
 - Dynamic backout failure
- ▶ Recovery needed
 - Recovery started
 - I/O error
- ▶ Read only
 - Command
- ▶ Prohibit AUTH
 - Command

```

DB
DBD=DBGAMBP                      DMB#=75    TYPE=IMS
SHARE LEVEL=1                    GSGNAME=**NULL**  USID=0000000004
AUTHORIZED USID=0000000004 RECEIVE USID=0000000004 HARD
USID=0000000004
RECEIVE NEEDED USID=0000000000
FLAGS:                            COUNTERS:
BACKOUT NEEDED                    =OFF          RECOVERY NEEDED COUNT  =0
READ ONLY                        =OFF          IMAGE COPY NEEDED COUNT =0
PROHIBIT AUTHORIZATION=OFF        AUTHORIZED SUBSYSTEMS  =1
RECOVERABLE                      =YES         HELD AUTHORIZATION STATE =6
                                      EEQE COUNT          =0

DBDS
DSN=IMS.SJIMSC.DBGAMBP           TYPE=IMS
DBD=DBGAMBP DDN=DBGAMBP DSID=001 DBORG=HIDAM DSORG=VSAM
CAGRP=**NULL** GENMAX=10 IC AVAIL=0 IC USED=1 DSSN=00000003
NOREUSE RECOVPD=0
DEFLTJCL=DBGDFLT ICJCL=SJIMSCC OICJCL=DBGOIC
RECOVJCL=DBGRECOV
RECVJCL=ICRCVJCL
FLAGS:                            COUNTERS:
IC NEEDED                        =OFF
RECOV NEEDED                     =OFF
RECEIVE NEEDED =OFF              EEQE COUNT              =0

```



Database Authorization Processing

➤ Access Intent

- ▶ Exclusive (EX)
- ▶ Update (UP)
- ▶ Read with integrity (RD)
- ▶ Read without integrity (RO)

➤ Online - DATABASE x,ACCESS=

➤ Batch - PCB DBD=x,PROCOPT=

- ▶ Exclusive (L or xE)
- ▶ Update (A,I,D,R)
- ▶ Read with integrity (G)
- ▶ Read without integrity (GO)

Database Authorization Processing



The world depends on it

➤ Held Authorization

- ▶ Highest access intent of "Running" subsystems which have this database authorized
- ▶ "Running" subsystems are defined as those subsystems running and any failed subsystems still holding authorizations until the backout/recovery is completed.

Note: DLIBATCH is considered a subsystem.

```
DB
DBD=DBGAMBP                      DMB#=75   TYPE=IMS
SHARE LEVEL=1                    GSGNAME=**NULL**  USID=0000000004

  FLAGS:                          COUNTERS:
BACKOUT NEEDED                    =OFF      RECOVERY NEEDED COUNT =0
READ ONLY                        =OFF      IMAGE COPY NEEDED COUNT =0
PROHIBIT AUTHORIZATION=OFF      AUTHORIZED SUBSYSTEMS =1
RECOVERABLE                      =YES      HELD AUTHORIZATON STATE =6
                                      EEQE COUNT =0

ASSOCIATED SUBSYSTEM INFORMATION:
                                     ENCODED B/O NEEDED
  -SSID-   -ACCESS INTENT-   -STATE-   -COUNT-   -SS ROLE-
  IMSC     UPDATE           6           0         ACTIVE

SYS
SID=IMSC   LOG START=99.148 13:56:20.9
SYPE=ONLINE ABNORMAL TERM=OFF RECOVERY STARTED=NO   BACKUP=NO
TRACKED=NO  TRACKER TERM=OFF  SHARING COVERED DBS=NO
IRLMID=**NULL**  IRLM STATUS=NORMAL  GSGNAME=**NULL**

AUTHORIZED DATA BASES/AREAS=6  VERSION=6.1
                                     ENCODED
  -DBD-   -AREA-  -LEVEL-  -ACCESS INTENT-  -STATE-
DBGAMBX  **NULL**  1      UPDATE        6
DBGAMBP  **NULL**  1      UPDATE        6
DBGAMAP  **NULL**  1      UPDATE        6
DBGAMBY2 **NULL**  1      UPDATE        6
DBGAMBY  **NULL**  1      UPDATE        6
DBGAMAY  **NULL**  1      UPDATE        6
```

Database Authorization Processing



The world depends on it

➤ If DBRC fails the authorization

- ▶ **DFS047A - UNABLE TO OBTAIN AUTHORIZATION FOR DATA BASE DBGAMAP . REASON CODE = 05. IMSC**
- ▶ **The database is marked as needing an IMAGE COPY (message text from the IMS Messages and Codes manual)**

LIST.HISTORY

(shows allocations in time sequence)



The world depends on it

LIST.HISTORY DBD(DBGAMBP)

IMAGE COPY

IMAGE

RUN = 99.144 23:23:29.5 * RECORD COUNT =3
STOP = 00.000 00:00:00.0 BATCH USID=0000000001

IC1

DSN=IMS.SJIMSC.DBGAMBP.BKUP.G0004V00 FILE SEQ=0001
UNIT=3390 VOLS DEF=0001 VOLS USED=0001
VOLSER=TSMS18

ALLOC

ALLOC =99.144 23:27:01.7 * ALLOC LRID =0000000000000000
DSSN=0000000001 USID=0000000002 START = 99.144 23:26:57.4

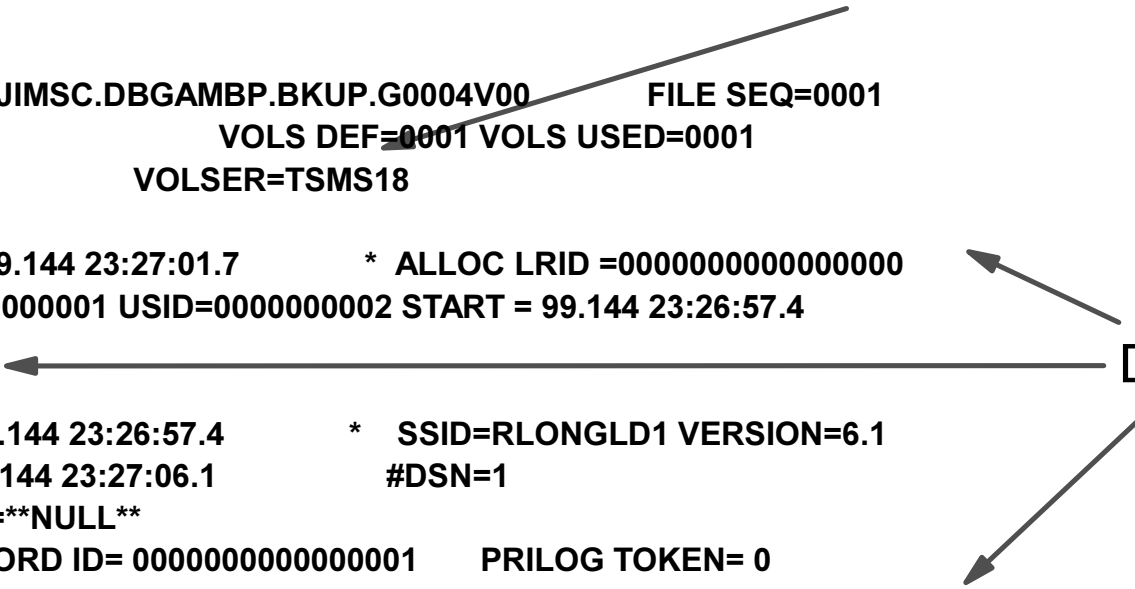
PRILOG

START = 99.144 23:26:57.4 * SSID=RLONGLD1 VERSION=6.1
STOP = 99.144 23:27:06.1 #DSN=1
GSGNAME=**NULL**
FIRST RECORD ID= 0000000000000001 PRILOG TOKEN= 0

DSN=IMS.SJIMSC.DBGB01.G0146V00 UNIT=3390
START = 99.144 23:26:57.4 FIRST DS LSN= 0000000000000001
STOP = 99.144 23:27:06.1 LAST DS LSN= 000000000001C9B
FILE SEQ=0001 #VOLUMES=0001

VOLSER=TOTTSM STOPTIME = 99.144 23:27:06.1
CKPTCT=0 CHKPT ID = 00.000 00:00:00.0

DLI BATCH JOB



LIST.HISTORY

(shows allocations in time sequence)



The world depends on it

ALLOC

ALLOC =99.145 00:05:38.3 * ALLOC LRID =0000000000000000
DSSN=0000000002 USID=0000000003 START = 99.144 22:05:12.1

PRILOG

START = 99.144 22:05:12.1 * SSID=IMSC VERSION=6.1
STOP = 99.145 12:07:48.1 #DSN=1
GSGNAME=**NULL**
FIRST RECORD ID= 000000000000ABB PRILOG TOKEN= 0
EARLIEST CHECKPOINT = 99.139 19:14:03.4

DSN=IMS.SJIMSC.SLDSP.IMSC.D99144.T2205121.V0C UNIT=3390
START = 99.144 22:05:12.1 FIRST DS LSN= 000000000000ABB
STOP = 99.145 12:07:48.1 LAST DS LSN= 0000000000001595
FILE SEQ=0001 #VOLUMES=0001
VOLSER=TOTTS4 STOPTIME = 99.145 12:07:48.1
CKPTCT=2 CHKPT ID = 99.145 12:07:47.6

IMS TM or
DBCTL System

ALLOC

ALLOC =99.145 12:11:38.5 * ALLOC LRID =0000000000000000
DSSN=0000000003 USID=0000000004 START = 99.145 12:08:53.8

PRILOG

START = 99.145 12:08:53.8 * SSID=IMSC VERSION=6.1
STOP = 00.000 00:00:00.0 #DSN=0
GSGNAME=**NULL**
FIRST RECORD ID= 0000000000001596 PRILOG TOKEN= 0
EARLIEST CHECKPOINT = 00.000 00:00:00.0



The world depends on it

Where Do I Start

- **Define backup strategy**
- **Define recovery strategy**
- **Modify update procedures**
- **Register databases**
- **Replace recovery procedures**
- **Create change accumulation procedures**
- **Modify test system procedures**



The world depends on it

Define Backup Strategy

- **Frequency**
 - ▶ Daily/Weekly/Monthly

- **Backup or rebuild secondary indexes**
 - ▶ Backup - register as recoverable
 - ▶ Rebuild - register as non-recoverable

- **Build JCL or GENJCL**
 - ▶ Built JCL can include pointer checker



The world depends on it

Define Recovery Strategy

- **Database dataset groups**
 - ▶ Databases and indexes
 - ▶ Logically related databases
 - ▶ Application related databases

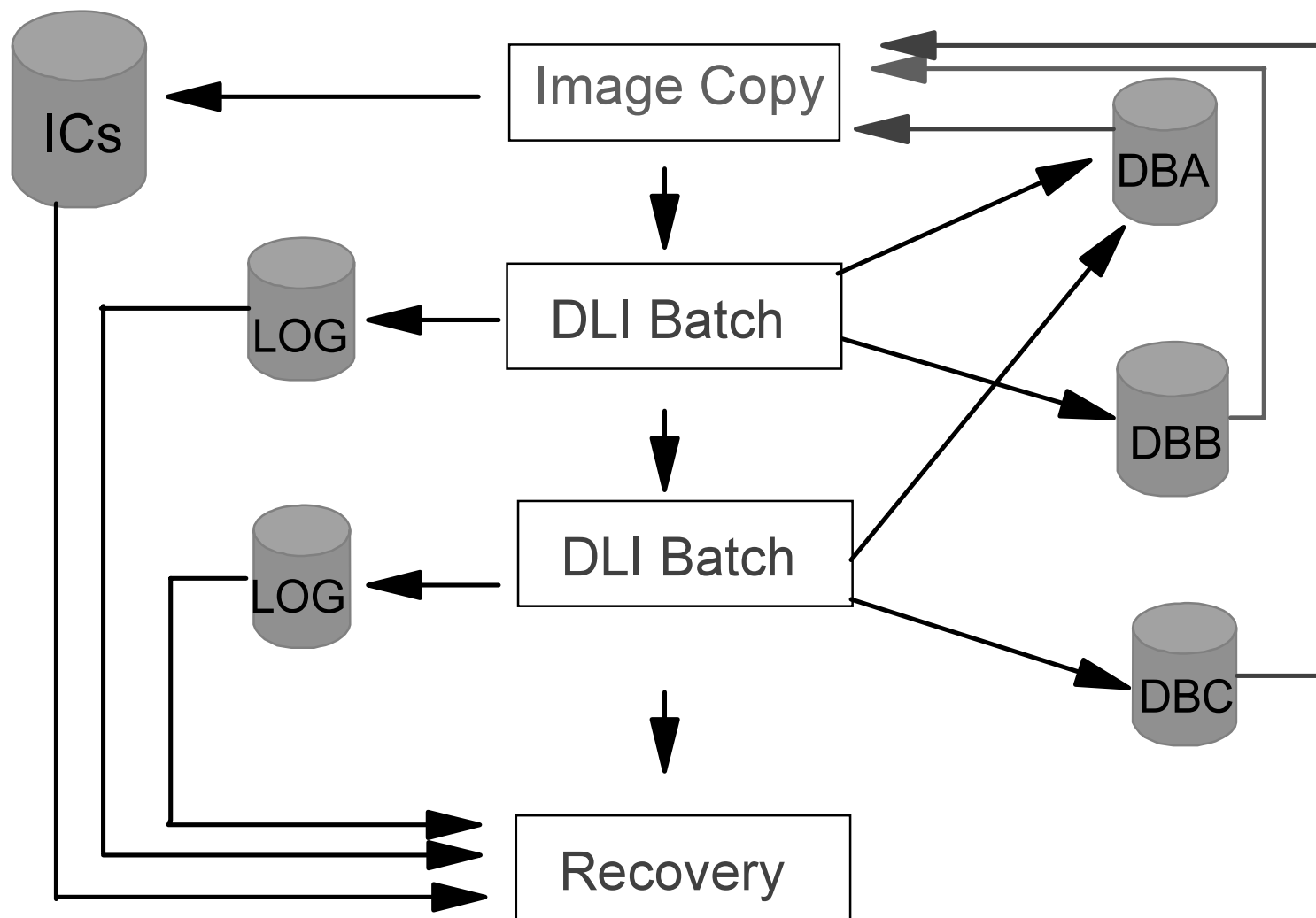
- **CHANGE ACCUMULATION groups**
 - ▶ Fewer groups mean fewer passes of the SLDSs
 - ▶ Smaller groups mean quicker recovery time

- **Recovery points**
 - ▶ Pre-defined point allows automated recovery jobs

Define Recovery Points



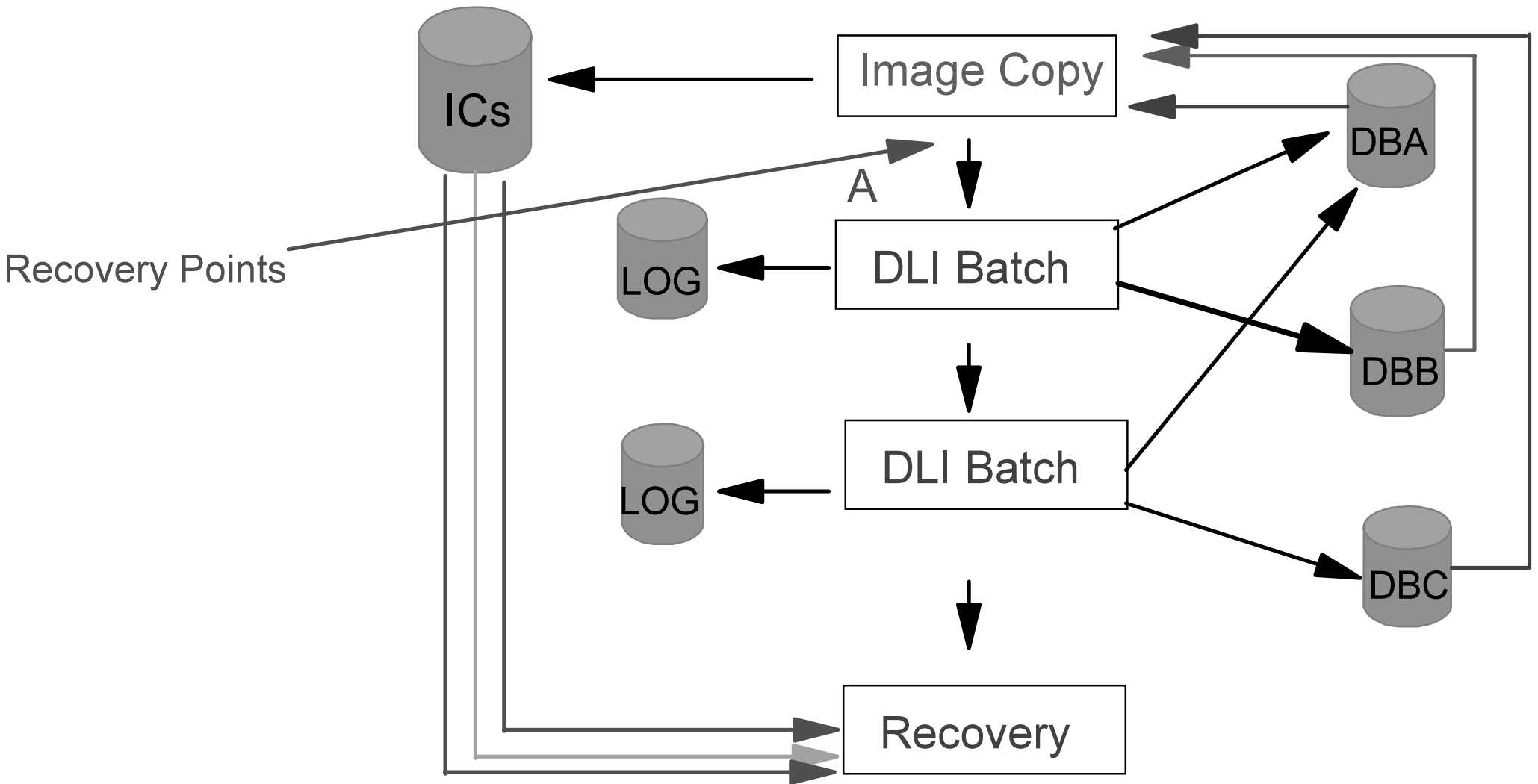
The world depends on it



Define Recovery Strategy

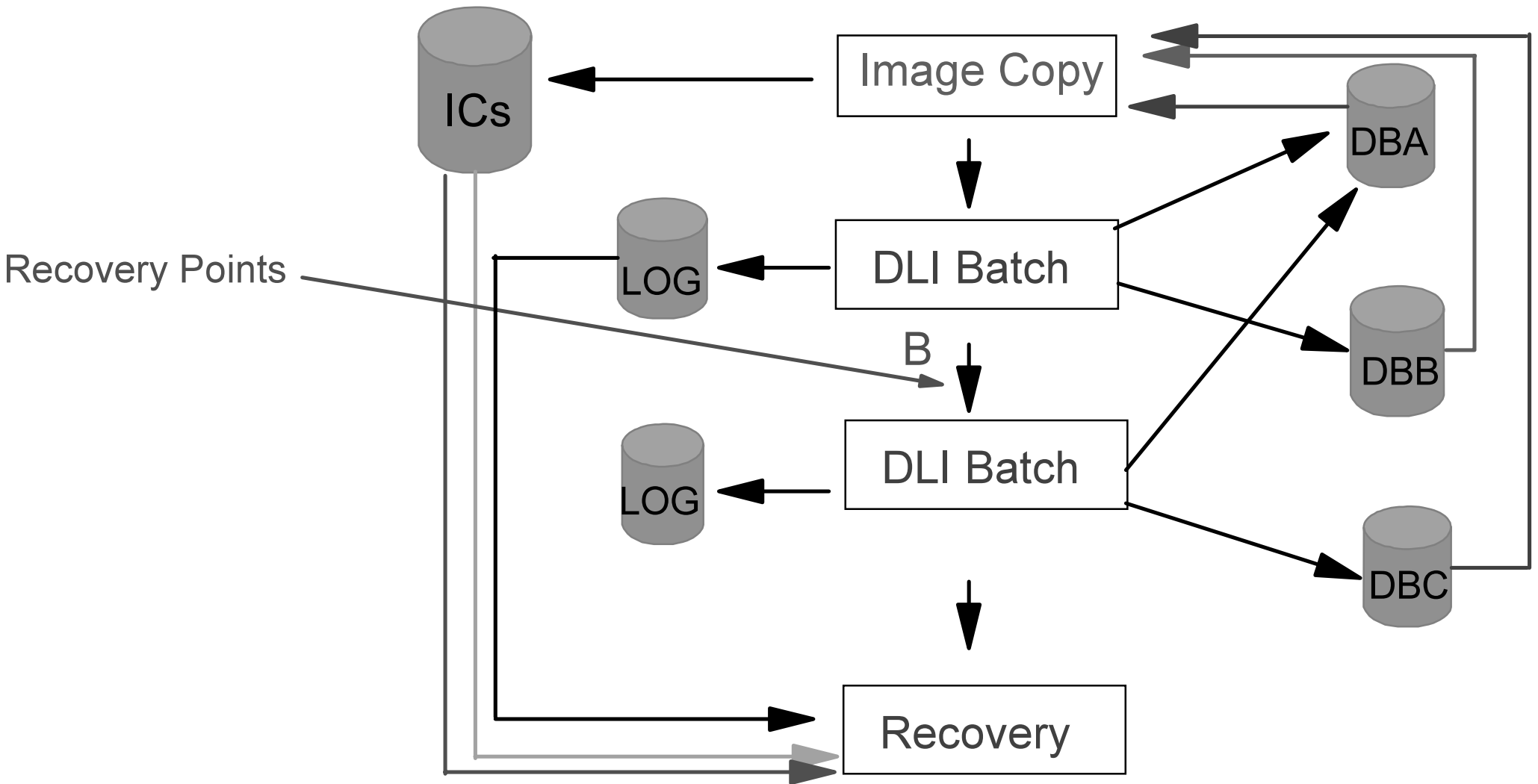


The world depends on it





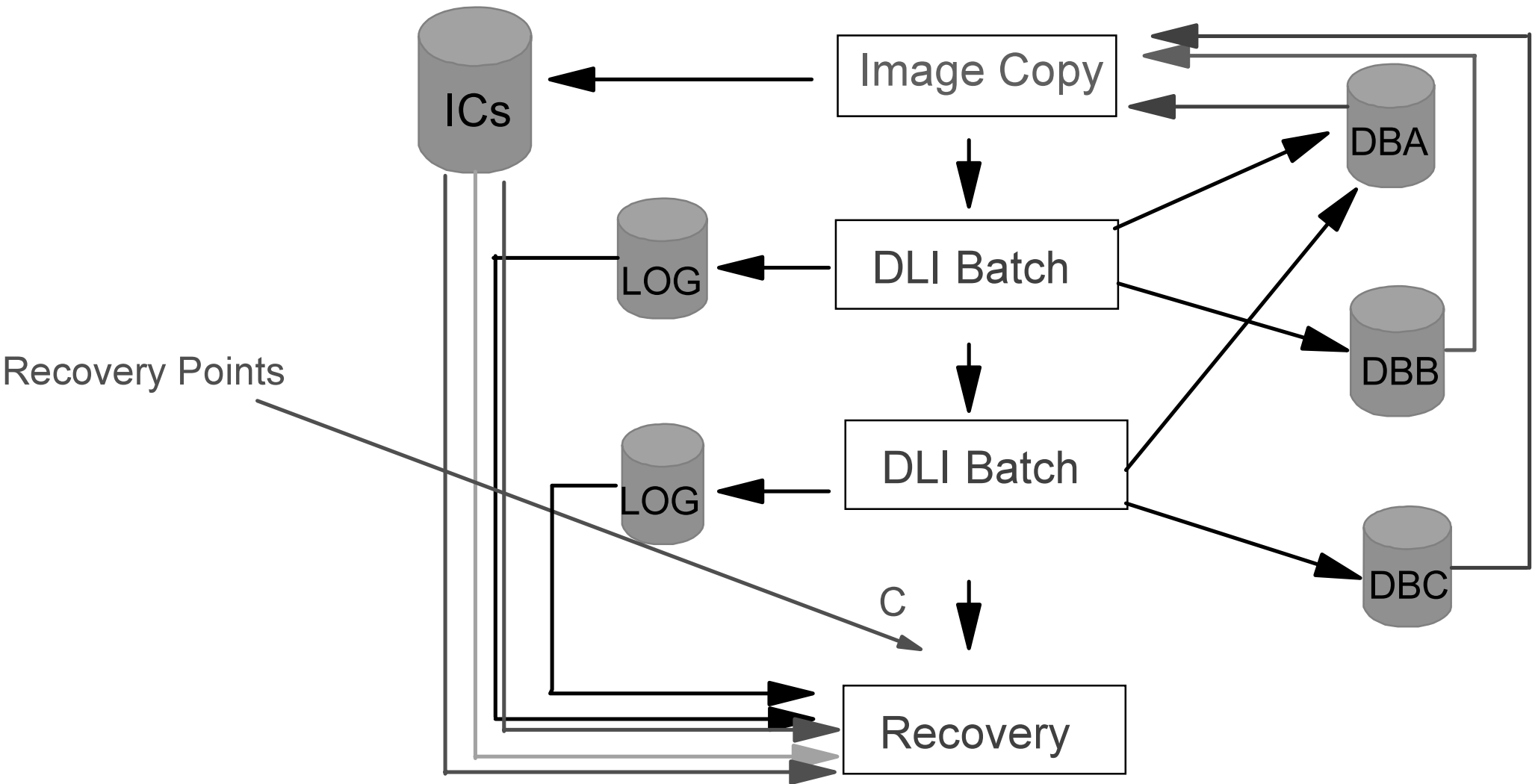
Define Recovery Strategy



Define Recovery Strategy



The world depends on it





The world depends on it

Define Recovery Strategy

- **Recovery Point A - recover a group which contains all three databases.**
- **2. Recovery Point B - recover a group which has DBA and DBB only.**
- **3. Recovery Point C - recover a group which has all three databases**



The world depends on it

Replace Recovery Procedures

➤ **Update Skeletal Members**

- ▶ **Application based RECOV members**
- ▶ **Make use if DEFAULTS member for system defaults**
 - Library names
 - Change Accumulation key size

➤ **Create GENJCL JOBS**

➤ **Update OPCA (JOB scheduler) to track both generating and generated JOBS**

Test System Differences



The world depends on it

➤ Production systems

- ▶ Scheduled image copies for all databases
- ▶ Managed SLDS/RLDS datasets
- ▶ Scheduled DB reorgs or DB Loads
- ▶ RECON status of FORCER
- ▶ Unique JOB names
- ▶ One DSN for a DBDNAME

➤ Test systems

- ▶ Infrequent image copies if at all
- ▶ Unmanaged and fewer SLDS/RLDS data sets
- ▶ Unscheduled DB Reorgs or DB loads
- ▶ RECON status of NOFORCER
- ▶ Duplicate JOB names
- ▶ Unit testing DSNs



The world depends on it

Image Copies and SLDSs

- **Infrequent image copies and unmanaged SLDS/RLDS data sets**
 - ▶ **Not all SLDS/RLDS datasets available to do database recovery**
 - Create RLDS GDGs with high limits and SMS migrate to cartridge
 - Force recoveries to IC timestamps
 - ▶ **Large PRILOG records**
 - Cycle the IMS system more frequently (daily/weekly)

➤ **Unscheduled DB Reorgs or DB loads**

▶ **IC needed flag gets set**

- Force the image copy to be run
- Use the CHANGE.DBDS ICOFF

▶ **IC GENMAX reached if too many DB loads**

- Increase GENMAX to 30 or so
- Run DELETE.LOG INACTIVE to reduce PRILOG record size



The world depends on it

NOFORCER/Unit Testing DSNs

➤ **Can not use unregistered databases**

- ▶ **Use CHANGE.RECON NOFORCER**
 - Allows all DBs to be used
 - Warning messages produced

➤ **More than one DSN for a DBD name**

- ▶ **Create recovery jobs to image copies only**
 - GENJCL - JCLOUT to library member
 - After creating recovery jobs unregister all databases

- ▶ **Make use of Batch Backout jobs to avoid recoveries**



The world depends on it

Non-Unique JOB Names

- **Unrelated JOBS may have same JOB name**
 - ▶ Can not solve this JOB names must be unique

- **Failed DLI JOBS still in RECON**
 - ▶ **Delete subsystem record**
 - CHANGE.SUBSYS SSID(jobname) STARTRCV
 - CHANGE.SUBSYS SSID(jobname) ENDRECOV
 - DELETE.SUBSYS SSID(jobname)

Modify Update Procedures



The world depends on it

- **JOB PARM needs to have Y in the DBRC Parm**

- **IEFRDER DD card must be added**
 - ▶ DSN can not be DUMMY
 - ▶ Should be unique DSN

- **If IRLM is used**
 - ▶ IRLMNM=irlmname
 - ▶ IRLM=Y

- **Dynamic backout can be used to avoid some batch backout requirements**
 - ▶ BKO=Y

Register Databases



The world depends on it

- **Running registration requires the DBDLIB**
- **Sets DEFAULTS skeletal recovery member**
- **Defines number of IMAGE COPY tracked for a database which defines the recovery window**
- **Defines database SHARELVL**
- **Can define a database as NONRECOV**
 - ▶ Reduces the log records available (used only for backout)
 - ▶ Can recover to IMAGE COPY only

Database Registration



The world depends on it

INIT.DB DBD(DBGAMAP) -
 SHARELVL(1)
INIT.DBDS DBD(DBGAMAP) -
 DDN(DBGAMAP1) -
 DSN(IMS.SJIMSC(DBGAMAP1) -
 GENMAX(10) -
 DEFLTJCL(DBGDFLT) -
 RECOVJCL(DBGRECOV)
INIT.DBDS DBD(DBGAMAP) -
 DDN(DBGAMAP2) -
 DSN(IMS.SJIMSC(DBGAMAP2) -
 GENMAX(10) -
 DEFLTJCL(DBGDFLT) -
 RECOVJCL(DBGRECOV)
INIT.DB DBD(DBGAMAY) -
 NONRECOV -
 SHARELVL(1)
INIT.DBDS DBD(DBGAMAY) -
 DDN(DBGAMAY) -
 DSN(IMS.SJIMSC(DBGAMAY) -
 GENMAX(10) -
 DEFLTJCL(DBGDFLT) -
 RECOVJCL(DBGRECOV)

INIT.DB DBD(DBGAMBP) -
 SHARELVL(1)
INIT.DBDS DBD(DBGAMBP) -
 DDN(DBGAMBP) -
 DSN(IMS.SJIMSC(DBGAMBP) -
 GENMAX(10) -
 DEFLTJCL(DBGDFLT) -
 RECOVJCL(DBGRECOV)

INIT.DB DBD(DBGAMBX) -
 SHARELVL(1)
INIT.DBDS DBD(DBGAMBX) -
 DDN(DBGAMBX) -
 DSN(IMS.SJIMSC(DBGAMBX) -
 GENMAX(10) -
 DEFLTJCL(DBGDFLT) -

INIT.DB DBD(DBGAMBY) -
 NONRECOV -
 SHARELVL(1)
INIT.DBDS DBD(DBGAMBY) -
 DDN(DBGAMBY) -
 DSN(IMS.SJIMSC(DBGAMBY) -
 GENMAX(10) -
 DEFLTJCL(DBGDFLT) -
 RECOVJCL(DBGRECOV)

INIT.DB DBD(DBGAMBY2) -
 NONRECOV -
 SHARELVL(1)
INIT.DBDS DBD(DBGAMBY2) -
 DDN(DBGAMBY2) -
 DSN(IMS.SJIMSC(DBGAMBY2) -
 GENMAX(10) -
 DEFLTJCL(DBGDFLT) -
 RECOVJCL(DBGRECOV)

INIT.DBDSGRP GRPNAME(DBGGRP1)
MEMBERS(-
 (DBGAMAP ,DBGAMAP1), -
 (DBGAMAP ,DBGAMAP2), -
 (DBGAMAY ,DBGAMAY) , -
 (DBGAMBP ,DBGAMBP) , -
 (DBGAMBX ,DBGAMBX) , -
 (DBGAMBY ,DBGAMBY) , -
 (DBGAMBY2,DBGAMBY2))

FRIEND



The world depends on it

- **Provides database integrity**
- **Provides additional report functions (history)**
- **Simplifies Recovery**