

S31

# Making DBRC Work for **You**

Rick Long

Senior IT Specialist

Silicon Valley Laboratory (Australian Branch)

[ricklong@au1.ibm.com](mailto:ricklong@au1.ibm.com)



Miami Beach, FL

October 22-25, 2001

# Introduction



*The world depends on it*

## ➤ Rick Long

### ▶ IMS Development BI tools (remotely)

- IMS Data Propagator
- IMS Data Refresher

### ▶ ITSO

- Redbooks
- IMS Specialist

- ◆ SG24 3333 - DBRC Usage and Hints
- ◆ SG24 2211-1 - Making your IMS Ready for Year 2000: Migrating to IMS Version 5
- ◆ sg24 5352 - IMS Primer

### ▶ IBM and life before IBM

- IMS Systems Programmer
- Database Administrator
- Application Programmer

### ▶ [ricklong@au1.ibm.com](mailto:ricklong@au1.ibm.com)

- I watch the [IMS-L@lists.missouri.edu](mailto:IMS-L@lists.missouri.edu) forum

# Objectives



*The world depends on it*

- To present some practical ideas on how to get DBRC to perform some of the day to day functions
  - ▶ Uses standard DBRC commands
  - ▶ Some home grown REXX programs
- Talk about any issues you might have with DBRC (can be continued during the conference)
- Can be continued on the **ims-l@lists.missouri.edu** forum

# What we are covering today



*The world depends on it*

- **Generating data set information**
  - ▶ IDCAMS DELETE/DEFINE
  - ▶ MDA members
  
- **Automatic generation to run IMS utilities**
  - ▶ Automatic generation of IMS Recovery Jobs
  - ▶ Automatic generation of Batch Backout Jobs
  
- **Retrieving RECON information**
  - ▶ DBDSGRP recovery timestamps
  - ▶ Database registration information
  - ▶ Last Image copy
  - ▶ # of volumes needed for Recovery
  - ▶ Online Usage times

# IDCAMS DELETE/DEFINE



*The world depends on it*

- **HALDB databases may have up to 1000 partitions of 10 data sets each**
  - ▶ All partitions are defined much the same
  - ▶ Easily changed when DSN change on moving to new RECON
  
- **Once registered to the RECON most of the information needed to define the data sets is known**
  - DSN
  - number of partitions (HALDB)
  - DBRC Symbolics used to define variable information
    - ◆ Size
    - ◆ csize
    - ◆ volumes (if needed)
    - ◆ Key size (if needed)
    - ◆ LRECL (if needed)

# Generating Delete/Defines



*The world depends on it*

```
GENJCL.USER GROUP(HALDBS) MEMBER(DBDDDEF) LIST JOB(JOBCARD) -  
ONEJOB DEFAULTS(DBGGDFLT) USERKEYS((%UPRIM,"500"),(%USEC,"100"))  
GENJCL.USER GROUP(HALDBSX) MEMBER(DBDDDEFL) LIST JOB(JOBCARD) -  
ONEJOB DEFAULTS(DBGGDFLT) USERKEYS((%UPRIM,"500"),(%USEC,"100"))
```

## DBDDDEF

```
%DELETE (%STPNO NE '00000')  
//S%STPNO EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
%ENDDEL  
%SELECT DBDS((%DBNAME,%DDNAME))  
DELETE %DBDSN  
SET MAXCC = 0  
DEFINE CLUSTER (NAME(%DBDSN) -  
VOLUMES (* * *) -  
CYL(%UPRIM %USEC) -  
REUSE SHR(3 3) -  
CISZ(4096) -  
SPEED -  
NONINDEXED -  
RECORDSIZE (4089 4089)) -  
DATA(NAME(%DBDDN.DATA))  
%ENDSEL
```

## DBDDDEFL

```
%DELETE (%STPNO NE '00000')  
//S%STPNO EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
%ENDDEL  
%SELECT DBDS((%DBNAME,%DDNAME))  
DELETE %DBDSN  
SET MAXCC = 0  
DEFINE CLUSTER (NAME(%DBDSN) -  
VOLUMES (* * *) -  
CYLINDERS (%UPRIM %USEC) -  
RECORDSIZE (50 50) -  
FREESPACE (10 10) -  
REUSE SHR(3 3) -  
CISZ(1024) -  
SPEED -  
KEYS (9 0)) -  
INDEX(NAME(%DBDSN.INDEX)) -  
DATA (NAME(%DBDSN.DATA))  
%ENDSEL
```

# Generating Delete/Defines



*The world depends on it*

```
GENJCL.USER GROUP(HALDBS) MEMBER(DBDDDEX) LIST JOB(JOBCARD) -  
ONEJOB DEFAULTS(DBGGDFLT) USERKEYS((%UPRIM,"500"),(%USEC,"100"),(%UCISZ,'1024'))
```

## DBDDDEX

```
%SELECT DBDS((%DBNAME,%DDNAME))  
  DEFINE CLUSTER (NAME(%DBDSN) -  
    VOLUMES (%UVOLS) -  
    CYLINDERS (%UPRIM %USEC) -  
    REUSE -  
    SHR(3 3) -  
    KEYS (6 5)) -  
    CISZ(%UCISZ) -  
    SPEED REPLICATE IMBED -  
    RECORDSIZE (12 12) -  
    FREESPACE (10 10) -  
    INDEX(NAME(%DBDSN.DATA))  
    DATA(NAME(%DBDSN.DATA) -
```

# Generating MDA members



*The world depends on it*

- **Once registered to the RECON, all the required information is known**
  - ▶ **DDNAME**
  - ▶ **DSN**
  
- **Non-HALDB databases**
  
- **Easily maintained**

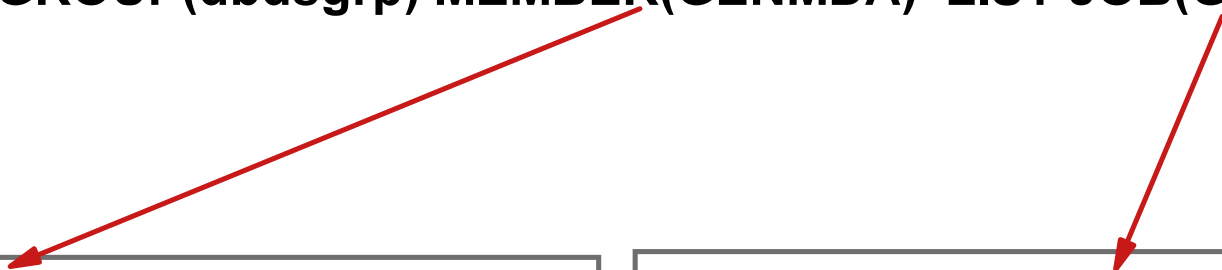


# Generating MDA member



*The world depends on it*

GENJCL.USER GROUP(dbdsgrp) MEMBER(GENMDA) LIST JOB(GENMDAJ)



## GENMDA

```
SELECT DBDS ( (%DBNAME, %DDNAME) )
  DFSMDA  TYPE=DATABASE, DBNAME=%DBNAME
  DFSMDA  TYPE=DATASET, DDNAME=%DDNAME,
          DSNAME=%DBDSN,
          DISP=SHR
%ENDSEL
  DFSMDA  TYPE=FINAL
END
```

## GENMDAJ

```
//GENMDBJB JOB TIME=10,
// CLASS=A,MSGCLASS=U,MSGLEVEL=(1,1),
// REGION=4M
//MYLIB JCLLIB ORDER=DPR.IMSB.PROCLIB
//STEP01 EXEC PROC=GENMDA,SOUT='*'
//SYSIN DD *
DFSMDA  TYPE=INITIAL
```

# Generating Recovery Jobs



*The world depends on it*

- **Automatic recovery from failure**
  - ▶ No operator intervention required (no mistakes)
  - ▶ Job scheduler driven (no wasted time)
  - ▶ To pre-determined recovery point
  
- **When?**
  - ▶ On failure of database reorganisation
  - ▶ During application batch cycle (DLIBATCH)

# Generating Recovery Jobs



*The world depends on it*

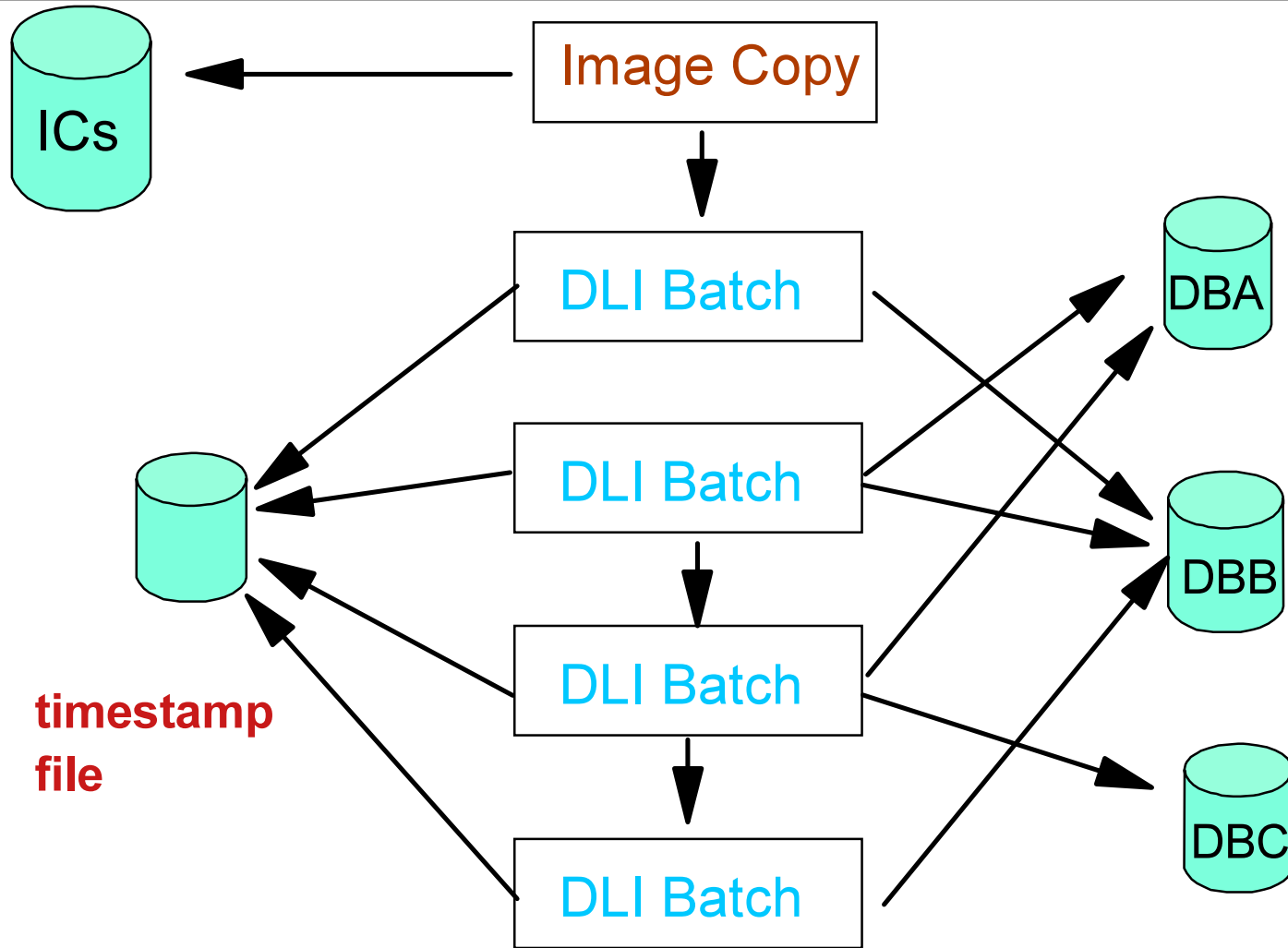
## ➤ How?

- ▶ Program creates application timestamp file
- ▶ Application timestamp file contains
  - timestamp
    - ◆ Valid DBRC timestamp
  - DBDSGRP to be recovered
    - ◆ DBDSGRP group name
  - JOB name
    - ◆ To delete SUBSYS record from RECON after recovery
  - PSB name
    - ◆ Used to generate Batch Backout Job
- ▶ Runs between DLI JOBs
- ▶ First step of update JOBs
- ▶ First step of database reorganization JOB
- ▶ Another program read timestamp file and creates GENJCL.RECOV DBRC command



*The world depends on it*

# Pre-defined Recovery Points



# Generating Batch Backout Jobs



*The world depends on it*

## ➤ Why?

- ▶ Faster than doing a DB recovery
  - small number of updates compared to DB size
- ▶ Application requires whole job or none to be processed
  - Can't use BKO=Y as this only goes back to last checkpoint

## ➤ How?

- ▶ Use same application timestamp file
- ▶ a program reads timestamp file and create GENJCL.USER

# Generating Batch Backout Jobs



The world depends on it

## Program generated DBRC command

```
GENJCL.USER MEMBER(BBO) USERKEYS((%USYSID,'jobname'),(%UPSB,'psbname'),
(%UTIME,'LAST'))
```



**BBO**

```
//STEP01 EXEC PGM=DFSRR00,REGION=4096K,
//          PARM=(DLI,DFSBB00,%PSB,,0000,,0,,N,0,T,,N,Y,,,Y)
//STEPLIB DD DSN=IMS.IMSV7.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.IMSV7.SDFSRESL,DISP=SHR
%SELECT RLDS (%USYSID,%UTIME)
//IMSLOGR DD DSN=%LOGDSN,DISP=SHR
%ENDSEL
//IEFRDER DD DSN=IMS.IMSV7.LOG(+1),
//          SPACE=(TRK,(100,10),RLSE),
//          UNIT=SLDSB,DISP=(NEW,CATLG),
//          DCB=(DUMDSCB,RECFM=VB,LRECL=23468,BLKSIZE=23472)
//IMS DD DSN=IMS.PSBLIB,DISP=SHR
// DD DSN=IMS.DBDLIB,DISP=SHR
//DFSVSAMP DD DSN=IMS.UTIL(VSAMP),DISP=SHR
//SYSIN DD *
```

# DBDSGRP timestamp information



*The world depends on it*

## ➤ Why?

- ▶ When recovering a DBDS group you need to know which IC for the group as the lastest time.
- ▶ Depends on how the IC were taken
- ▶ Which order the DBDs are registered in the DBDSGRP

## ➤ How?

- ▶ GENJCL.USER commands
- ▶ Modified members

# Extract DB IC times from DBDSGRP



*The world depends on it*

## ➤ Extract the list IC Time for a DBDSGRP

- ▶ This can be used to obtain an IC time to perform a recovery.
- ▶ Must use the last IC time for the group to get the correct recovery results
- ▶ Getting the DSN helps to verify the results
- ▶ Change SYSOUT to a dataset and sort on timestamps to find last time

```
//RLONGICX JOB (@IMS,FA-C),'RICK LONG - DBG',MSGCLASS=V
//DBRC EXEC PGM=DSPURX00,COND=(0,NE)
//STEPLIB DD DSN=IMS.SJIMSC.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
GENJCL.USER GROUP(DBGPRIM) MEMBER(ICTIME) LIST -
NOJOB USERKEYS(%WHICHIC,'LAST')
/*
```

```
%SET TIMEFMT(,N)
%SELECT IC ((%DBNAME,%DDNAME),%WHICHIC)
%DBNAME %ICTIME %ICDSN
%ENDSELECT
```

```
DBGAMAP 991581400049 IMS.SJIMSC.DBGAMAP1.BKUP.G0012V00
DBGAMAP 991581400064 IMS.SJIMSC.DBGAMAP2.BKUP.G0012V00
DBGAMBX 991581400074 IMS.SJIMSC.DBGAMBX.BKUP.G0012V00
DBGAMP 991581400085 IMS.SJIMSC.DBGAMP.BKUP.G0012V00
```



# Extracting Registration Information



*The world depends on it*

## ➤ What?

- ▶ This JOB will extract the registration information into a form which can be used to re-register the information. It is effective for copying the information from a test system to a production one.

## ➤ Why?

- ▶ When creating a test system and you want to test the DBRC functions. To accomplish this you copy the production system registration information to the test system.
- ▶ When upgrading the IMS release and you don't want to use the RECON upgrade utility.
- ▶ When, having tested the DBRC functions in the test system, you want to delete the registration information to either reproduce the same test results or leave the database unregistered during unit testing. You could then recreate the DBRC test case at a later time.

# Extracting registration information



*The world depends on it*

- To extract the registration information for a database group the following job can be used

```
//DBRCEXT JOB (@TS3,FA33),'DBRCEXT',  
// CLASS=A,NOTIFY=&SYSUID,MSGCLASS=U  
//*  
//* GENERATE DB REGISTRATION EXTRACT JOB  
//*  
//DBRC EXEC PGM=DSPURX00  
//STEPLIB DD DSN=IMS.IMS6.RESLIB,DISP=SHR  
//JCLPDS DD DSN=IMS.IMS6.JCLLIB,DISP=SHR  
//IMS DD DSN=IMS.IMS6.DBDLIB,DISP=SHR  
//JCLOUT DD DSN=IMS.IMS6.RUN(REGOUT),DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
GENJCL.USER GROUP(DBGPRIM) MEMBER(DBDSREG) -  
LIST ONEJOB DEFAULTS(DBGGDFLT)  
//
```

# Extracting registration information



*The world depends on it*

➤ This is the skeletal member used in the extract registration information

```
%DELETE (%STPNO NE '00000')
//DBRCREG JOB (@TS1,FA-C),'DB REG',
// CLASS=A,NOTIFY=&SYSUID,MSGCLASS=U
//S%STPNO EXEC PGM=DSPURX00,COND=(0,NE)
//STEPLIB DD DSN=IMS.IMS7.SDFSRESL,DISP=SHR
//IMS DD DSN=IMS.IMS7.DBDLIB,DISP=SHR
//JCLPDS DD DSN=IMS.IMS7.JCLLIB,DISP=SHR
//JCLOUT DD DSN=IMS.IMS7.RUN(REGOUT),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
%ENDDDL
%SELECT DBDS(%DBNAME,%DDNAME)
INIT.DB DBD(%DBNAME) SHARELVL(1)
INIT.DBDS DBD(%DBNAME) DDN(%DBDDN) -
GENMAX(%MAXGEN) DSN(%DBDSN) -
ICJCL(%JCLIC) RECOVJCL(%JCLRECV) -
DEFAULTS(%JCLDFLT)
%ENDSEL
```



*The world depends on it*

# Extracting registration information

## ➤ This is the generated job

▶ There were 2 databases in the DBDSGRP

▶ The DEFAULTS member resolved JCL member symbolics

➤ **Note:** it is a JOB ready to be submitted to re-register that information into a new RECON

```
//DBRREG JOB (@TS1,FA-C),'DB REG',
// CLASS=A,NOTIFY=&SYSUID,MSGCLASS=U
//S%STPNO EXEC PGM=DSPURX00,COND=(0,NE)
//STEPLIB DD DSN=IMS.IMS7.SDFSRESL,DISP=SHR
//IMS DD DSN=IMS.IMS7.DBDLIB,DISP=SHR
//JCLPDS DD DSN=IMS.IMS7.JCLLIB,DISP=SHR
//JCLOUT DD DSN=IMS.IMS7.RUN(REGOUT),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INIT.DB DBD(DBGAMAP) SHARELVL(1)
INIT.DBDS DBD(DBGAMAP) DDN(DBGAMAP) -
GENMAX(8) DSN(IMS.IMS7(DBGAMAP) -
ICJCL(DBGIC)RECOVJCL(DBGRECOV) -
DEFAULTS(DBGDFLT)
INIT.DB DBD(DBGAMP) SHARELVL(1)
INIT.DBDS DBD(DBGAMP) DDN(DBGAMP) -
GENMAX(8) DSN(IMS.IMS7(DBGAMP) -
ICJCL(DBGIC)RECOVJCL(DBGRECOV) -
DEFAULTS(DBGDFLT)
```

# Last Image Copy



*The world depends on it*

## ➤ Why?

- ▶ Coping production data to test system
- ▶ Creating new RECON

## ➤ How?

- ▶ GENJCL.USER
- ▶ Modified member to create NOTIFY.IC cards



The world depends on it

# Last Image Copy

➤ To extract the IC data set names, times and volume information

- ▶ "LAST" is used in this example
- ▶ Any timestamp could be used
- ▶ Any DBDBGRP can be used

```
//SELECTIC JOB (@IMS,FA-C),'SEL IC',MSGCLASS=V
//DBRC      EXEC  PGM=DSPURX00,COND=(0,NE)
//STEPLIB   DD   DSN=IMS.IMS6.RESLIB,DISP=SHR
//JCLOUT    DD   DSN=IMS.IMS6.RUN(REGOUT),DISP=SHR
//SYSPRINT  DD   SYSOUT=*
//SYSIN     DD   *
GENJCL.USER GROUP(DBGPRIM) MEMBER(ICTIME) LIST -
NOJOB  USERKEYS(%WHICHIC,'LAST')
/*
```



*The world depends on it*

# Last Image Copy

➤ The is the skeletal member used.

- ▶ The JCL is for a different IMS system for the NOTIFY.
- ▶ Can be any version of IMS

```
%DELETE (%STPNO NE '00000')
//DBRCREG JOB (@TS1,FA-C),'DB REG',
//  CLASS=A,NOTIFY=&SYSUID,MSGCLASS=U
//S%STPNO EXEC PGM=DSPURX00,COND=(0,NE)
//STEPLIB DD DSN=IMS.IMS7.SDFSRESL,DISP=SHR
//IMS DD DSN=IMS.IMS7.DBDLIB,DISP=SHR
//JCLPDS DD DSN=IMS.IMS7.JCLLIB,DISP=SHR
//JCLOUT DD DSN=IMS.IMS7.RUN(REGOUT),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
%ENDDDEL
%SELECT IC((%DBNAME,%DDNAME),%WHICHIC)
NOTIFY.IC DBD(%DBNAME) DDN(%DDNAME) -
ICDSN(%ICDSN) -
RUNTIME(%ICTIME) -
UNIT(%ICUNIT) VOLLIST(%ICCOLS)
%ENDSEL
```



The world depends on it

# Last Image Copy

➤ This is the generated job

▶ There were 2 databases in the DBDSGRP

➤ **Note:** it is a JOB ready to be submitted to re-register that information into a new RECON

```
//DBRCREG JOB (@TS1,FA-C),'DB REG',  
// CLASS=A,NOTIFY=xxxxxxx,MSGCLASS=U  
//S%STPNO EXEC PGM=DSPURX00,COND=(0,NE)  
//STEPLIB DD DSN=IMS.IMS7.SDFSRESL,DISP=SHR  
//IMS DD DSN=IMS.IMS7.DBDLIB,DISP=SHR  
//JCLPDS DD DSN=IMS.IMS7.JCLLIB,DISP=SHR  
//JCLOUT DD DSN=IMS.IMS7.RUN(REGOUT),DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
NOTIFY.IC DBD(DBGAMAP) DDN(DBGAMAP) -  
ICDSN(IMS.IMSA(DBGAMAP.IC) -  
RUNTIME(011281653407) -  
UNIT(3390) VOLLIST(TST001)  
NOTIFY.IC DBD(DBGAMBP) DDN(DBGAMBP) -  
ICDSN(IMS.IMSA(DBGAMBP.IC) -  
RUNTIME(011281684327) -  
UNIT(3390) VOLLIST(TST104)
```



# Online Usage Time



*The world depends on it*

## ➤ Why?

- ▶ Want to know what times a database gets used
  - Want to find a time to take it off line
- ▶ Service Level Agreements
- ▶ Which databases are not being used
- ▶ Which databases are being taken off-line (how long for)

## ➤ How?

- ▶ GENJCL.USER command
- ▶ Non JCL skeletal member

# Online Usage Times



*The world depends on it*

➤ GENJCL.USER MEMBER(PRIALLOC) NOJOB -  
USERKEYS((%UTIME1,'013220000000'),(%UTIME2,'0132259599'))

```
/**
/**      REPORT ON ALL DBDS UPDATED BY ONLINE SUBSYSTEM SESSION
/**
REPORT #:F0000      DATABASE      START TIME      STOP TIME
%SELECT ALLOC(PRILOG,(FROM(%UTIME1),TO(%UTIME2)))
                %DBNAME          %ALLTIME       %DALTIME
%ENDSEL
```

```
/**
/**      REPORT ON ALL DBDS UPDATED BY ONLINE SUBSYSTEM SESSION
/**
REPORT #:F0000      DATABASE      START TIME      STOP TIME
                DBGAMPB          013220715316    000000000000
                DBGAMPX          013220715323    000000000000
                DBGAMAP          013220703200    013221159013
                DBGAMAP          013221303432    013221301014
                DBGAMAP          013221219542    000000000000
```

# Volumes Needed for Recovery



*The world depends on it*

## ➤ Why?

- ▶ How many volumes will be needed for a specific GENJCL.RECOV command?
- ▶ Setting IMS ORS READNUM parameter number of concurrent devices
- ▶ How many logs?
  - Should I run a change accum?
  - Pick better time?
- ▶ Split a group up to separate GENJCL.RECOV commands
- ▶ Volumes on site?
- ▶ Migrated data sets?

## ➤ How?

- ▶ GENJCL.RECOV command
- ▶ Modified skeletal member

# Volumes Needed for Recovery



*The world depends on it*

**GENJCL.RECOV GROUP(XXXXX) MEMBER(GENSERS) LIST NOJOG**

## ➤ The skeletal member

- ▶ There were 5 databases in the DBDSGRP
- ▶ Don't need DSN information but you could include it if relevant
- ▶ The IC/CA/LOG is optional but nice to know

```
//*  
%DELETE (%ICDSN EQ '' )  
//IC UNIT=%ICUNIT ,VOL=SER=%ICVOLS  
%ENDDDEL  
%DELETE (%CADSN EQ '' )  
//CA UNIT=%CAUNIT ,VOL=SER=%CAVOLS  
%ENDDDEL  
%SELECT RLDS ( ( %DBNAME , %DBDDN ) , FROM ( %DSLLOGTM ) )  
//LOG UNIT=%LOGUNIT ,VOL=SER=%LOGVOLS  
%ENDSEL
```

# Volumes Needed for Recovery



*The world depends on it*

## ➤ The generated output

- ▶ Output set to a data set
- ▶ REXX program to total number of volumes
- ▶ Number of "Tapes"

```
//*  
//IC      UNIT=3480,VOL=SER=T04303  
//LOG     UNIT=3390,VOL=SER=SM4214  
//*  
//IC      UNIT=3480,VOL=SER=T04303  
//LOG     UNIT=3390,VOL=SER=SM4214  
//*  
//IC      UNIT=3480,VOL=SER=T04117  
//*  
//IC      UNIT=3480,VOL=SER=T04321  
//LOG     UNIT=3390,VOL=SER=SM4211  
//*  
//IC      UNIT=3480,VOL=SER=T04214  
//LOG     UNIT=3390,VOL=SER=SM4213
```