



IBM Software Group

Does worrying about how to recover your  
IMS databases keep you awake at night?



IMS Tools  
Database Recovery Facility (DRF)  
March 2006

Victor Lim SW Engineer vlim@us.ibm.com



# Agenda

- Objectives of this call
- Case of the Bad REORG Utility
- Alternative to Testing with Production Database
- Updating Your Image Copies
- Resources
- Next steps
- Questions

## Objectives of this Call

- How do you ensure during offsite, disaster, or local recovery, that the image copy data sets, IMS logs, and change accumulation data sets are accessible?
- Would you like to be able to issue IMS commands during batch recovery to help ensure that the databases are offline before recovery starts and restarted at the completion?
- Is the process for your recovery related tasks too complicated... validation, image copy and rebuild? Do you get a consolidated report upon completion of your recovery tasks?
- Do you need to build a test, audit copy or image copy of your production data without disrupting your production data?
- Do you need to run incremental image copies to the next logical commit point on the log without disrupting the databases?

## Case 1: Case of the Bad REORG Utility

- **Your company just received a Beta version of a long awaited DB REORG tool.**
- **Your team is at the edge of their seats and eager to put it to work.**
- **However, it's been 2 days since you've felt the comfort of your bed so you decided to hold off testing until the next morning.**
- **As you're getting ready to leave the office you hear foot steps running towards you and voices yelling your name.**
  
- **Uh oh...**
  - ▶ In their eagerness to impress you, your team decided to begin testing the tool.
  - ▶ They installed the tool on your production environment.
  - ▶ The tool terminated part way through the test and corrupted several databases.
  - ▶ The most current image copies of your databases are a week old.
  
- **What can you do?**

## Case 1: Case of the Bad REORG Utility - Solving the Case

- **We will show you how a DRF batch job can help you:**
  - ▶ Run a VERIFY to ensure that the required input data sets (image copies, change accumulation, and logs) for recovery are accessible.
  - ▶ Issue an IMS /DBR command to ensure the databases are offline.
  - ▶ Automatically delete and redefine your database data sets.
  - ▶ Take a new image copy in parallel to recovery process.
  - ▶ Run pointer checker to validate database contents in parallel to recovery process.
  - ▶ Issue an IMS /STA command to start the databases after recovery completes.

## Case 1: Run VERIFY job to ensure all required input data sets are accessible (1 of 3)

```

EDIT          VLIM.DRF.JOBS(DRFV01) - 01.24          Columns 00001 00080
Command ==>          Scroll ==>  CSR
000100 //DRFV01  JOB  REGION=0M,MSGLEVEL=(1,1),MSGCLASS=H          00010021
000110 //*****          00011000
000131 /** IMS DATABASE RECOVERY FACILITY FOR Z/OS - V3R1          * 00013103
000132 /**      DRF VERIFY OF DATA SET INPUTS FOR 6 HALDB DATABASES          * 00013223
000134 /**          * 00013412
000150 /**      PROCEDURE FOR DATABASE RECOVERY FACILITY (DRF) MASTER REGION          * 00015000
000160 /**          * 00016000
000170 //*****          00017000
000186 /**          00018611
000190 /**      FRXJCL3P PROC FOR DATABASE RECOVERY FACILITY          * 00019000
000200 /**          DRFV3 PROC for RPTTYPE=SEP          * 00020000
000210 /**          00021000
000220 //FRXJCL3P PROC RGN=0M,SOUT='*',          00022000
000230 /**          RESLIB='IMSB LD.IMSTOOL.DRFV3R1.SFRXRESL',          00023017
000240 /**          RESLIB='IMSB LD.DRFV31.FRX.SFRXRESL',          00024017
000241 /**          HPICRES='IMSB LD.IMTOOL1.HPICV4.PK11605.RESLIB',          00024124
000250 /**          HPPCRES='IMSB LD.IMSTOOL.DRFV3R1.HPPC.SFRXRESL',          00025000
000260 /**          DEPCRES='IMSB LD.IMSTOOL.DRFV3R1.DEDBPC.SFRXRESL',          00026000
000270 /**          IBRES='IMSB LD.IMSTOOL.DRFV3R1.HPIB.SFRXRESL',          00027000
000280 /**          DRFMBR=T1,          00028000
000290 /**          BPECFG=FRXBPECF,          00029000
000291 /**          DRFPROC=DRFV32,          00029115
000292 /**          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=6118),OUTLIM=0'          00029200
000293 /**          U=SYSDA          - Required only when RPTTYPE=APP          00029300
000294 //*****          00029400
000295 /** BRING UP A RECOVER DATA MANAGER          * 00029500
000296 //*****          00029600
000297 //STEP1 EXEC PGM=FRXSDR00,          00029700
000298 //      PARM='DRF,BPECFG=&BPECFG,DRFMBR=&DRFMBR,DRFPROC=&DRFPROC',          00029800
000299 //      REGION=&RGN,TIME=1440          00029900
000301 //STEPLIB DD DSN=&RESLIB,DISP=SHR          00030120
000302 //      DD DSN=&HPICRES,DISP=SHR          00030200
000303 //      DD DSN=&HPPCRES,DISP=SHR          00030300
000304 //      DD DSN=&DEPCRES,DISP=SHR          00030400
000305 //      DD DSN=&IBRES,DISP=SHR          00030500
000306 //      DD DSN=IMTOOL1.I91RTS14.LRESLIB1,DISP=SHR          00030600
000307 //IMSDALIB DD DSN=IMTOOL1.DRFV3.DYNALL64,DISP=SHR          00030724
000308 //PROCLIB DD DSN=IMTOOL1.IMS.PROCLIB,DISP=SHR          00030800
000309 //      DD DSN=IMTOOL1.PROCLIB,DISP=SHR          00030900
000310 //DBDLIB DD DSN=IMTOOL1.DSFF00.DBDLIB32,DISP=SHR          00031015
000311 //RECON1 DD DSN=DKSILVA.DRFV3#1.RECON91,DISP=SHR          00031113
000312 //RECON2 DD DSN=DKSILVA.DRFV3#1.RECON92,DISP=SHR          00031213
000313 //SYSIN DD DUMMY          00031300
000314 //SYSPRINT DD SYSOUT=&SOUT          00031400
F1=Help      F2=Split      F3=Exit      F5=Rfind      F6=Rchange      F7=Up      F8=Down
F9=Swap      F10=Left      F11=Right      F12=Cancel

```

- The next 2 slides shows a sample DRF proc JCL to setup your primary address space.
- The same proc will be used for each of the DRF batch jobs in this presentation.
- The 3<sup>rd</sup> slide will show the SYSIN statements you need to specify in order to invoke the DRF VERIFY function.
- DRF V3R1 gives you the ability to execute 3 other IMS Tools:
  - ▶ HPIC
  - ▶ HPPC
  - ▶ DEDBPC

## Case 1: Run VERIFY job to ensure all required input data sets are accessible (2 of 3)

```

EDIT          VLIM.DRF.JOBS(DRFV01) - 01.24          Columns 00001 00080
Command ==>          Scroll ==>  CSR
000315 //SYSUDUMP DD  SYSOUT=&SOUT                    00031500
000316 //REPORT DD  SYSOUT=&SOUT,&DCB                 00031600
000317 /*-----* 00031700
000318 /* DRF DD's for IAUs:                          * 00031800
000319 /*      FRXWTO - only required when RPTTYPE=SEP * 00031900
000320 /*-----* 00032000
000321 //FRXWTO  DD  SYSOUT=&SOUT,&DCB                 00032100
000322 //DFSRESLB DD  DSN=IMTOOL1.I91RTS14.LRESLIB1,DISP=SHR 00032200
000323 //IMS     DD  DSN=IMTOOL1.DSFF00.DBDLIB32,DISP=SHR 00032324
000324 //ICE#DOUT DD  SYSOUT=&SOUT,&DCB                 00032409
000325 /*-----* 00032500
000326 /* HPIC DD's fo IAUs when RPTTYPE=SEP          * 00032600
000327 /*-----* 00032700
000328 //ICEPRINT DD  SYSOUT=&SOUT,&DCB                 00032800
000329 //ICERPR   DD  SYSOUT=&SOUT,&DCB                 00032900
000330 //DFSPRIN DD  SYSOUT=&SOUT,&DCB                 00033000
000331 /*-----* 00033100
000332 /* HPPC DD's fo IAUs when RPTTYPE=SEP          * 00033200
000333 /*-----* 00033300
000334 //PRIMAPRT DD  SYSOUT=&SOUT,&DCB                 00033400
000335 //EVALUPRT DD  SYSOUT=&SOUT,&DCB                 00033500
000336 //SUMMARY  DD  SYSOUT=&SOUT,&DCB                 00033600
000337 //STATIPRT DD  SYSOUT=&SOUT,&DCB                 00033700
000338 //VALIDPRT DD  SYSOUT=&SOUT,&DCB                 00033800
000339 //SNAPPIT  DD  SYSOUT=&SOUT,&DCB                 00033900
000340 /*-----* 00034000
000341 /* DEDB PC DD's for IAUs when RPTTYPE=SEP      * 00034100
000342 /*-----* 00034200
000343 //FABARPRT DD  SYSOUT=&SOUT,&DCB                 00034300
000344 //FABAMSG  DD  SYSOUT=&SOUT,&DCB                 00034400
000345 //FABASNAP DD  SYSOUT=&SOUT,&DCB                 00034500
000346 /*-----* 00034600
000347 /* IB DD's for IAUs when RPTTYPE=SEP           * 00034700
000348 /*-----* 00034800
000349 //IIUSNAP  DD  SYSOUT=&SOUT,&DCB                 00034900
000350 //IIUSOUT  DD  SYSOUT=&SOUT,&DCB                 00035000
000351 //IIUSTAT DD  SYSOUT=&SOUT,&DCB                 00035100
000352 //IIUPRIN DD  SYSOUT=&SOUT,&DCB                 00035200
000353 /*-----* 00035300
000354 /* DFSPREC0 DD's for IAUs when RPTTYPE=SEP      * 00035400
000355 /*-----* 00035500
000356 //PRPRINT  DD  SYSOUT=&SOUT,&DCB                 00035600
000357 /*-----* 00035700
000358 //          PEND                                00035800
F1=Help   F2=Split   F3=Exit   F5=Rfind   F6=Rchange F7=Up     F8=Down
F9=Swap   F10=Left  F11=Right F12=Cancel

```

- This 2<sup>nd</sup> slide is a continuation from the previous slide.
- This slide identifies some of the new DD's you need to specify in order to be able to invoke HPIC, HPPC, and DEDBPC as part of DRF.

## Case 1: Run VERIFY job to ensure all required input data sets are accessible (3 of 3)

```

EDIT          VLIM.DRF.JOBS(DRFV01) - 01.26          Columns 00001 00080
Command ==> _____ Scroll ==> CSR
000360 /*                                           00036000
000501 //FRXJCL01 EXEC FRXJCL3P                    00050100
000502 //SYSIN DD *                                00050200
000509 ADD DB(HOTELDBA, -                           00050925
000510         HOTELDBB, -                           00051025
000511         HOTELDBC)                             00051125
000512 ADD DB(HOTELDBD, -                           00051225
000513         HOTELDBE, -                           00051325
000514         HOTELDBF)                             00051425
000519 START ERROR(CONT) VERIFY(ALLOC)              00051926
000520 RCVTIME('06.056 06:00:00.0' PITR NOCHECK)   00052026
000530 /*                                           00053026
***** ***** Bottom of Data *****

```

- We are ready to specify the SYSIN statements necessary to invoke the VERIFY function.
- The ADD statement is used to build a recovery list of the databases involved.
- The START statement initiates the job, specifying the recovery time we desire.
- VERIFY option on the START statement has 3 flavors:
  - ▶ **LIST:** return a list of required inputs (image copies, logs, change accumulation).
  - ▶ **ALLOC:** allocate each data set to verify existence & accessibility.
  - ▶ **OPEN:** open each dataset to further very accessibility.
- This example shows ALLOC.



## Case 1: DRF VERIFY report output (1 of 2)

```

Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY DRFV01  JOB02143  DSID   104 LINE 16      COLUMNS 02- 121
COMMAND INPUT ==>          SCROLL ==>  CSR
D A T A B A S E  R E C O V E R Y  F A C I L I T Y  R E C O V E R Y  P A R A M E T E R S

PROCESS : VERIFY(ALLOC)    RECOVERY OPTIONS
SOURCE  : PRI              RCVTIME : 2006.056 06:00:00.000000 -08:00
READNUM : 10,10           TYPE    : PITR              LOGNUM  : 10,10      ICNUM   : 10,10
ERROR   : CNT             CHECK   : No                OUTPUT  : Pro
RCVTOKEN: DRFV01         TIME FMT: LCL
DRFPROC : DRFV32         LBI    : No                SPSIZE  : 1024      CACHE   : No

----- Sort-Related Parameters -----
NUM      : 10              MAINSIZE: 100              AVGRLEN : 1024
FILSZ    : 400000         HIPRMAX : OPTIMAL          ASPREF  : FRXI
DYNALLOC : N/A , N/A

D A T A B A S E  R E C O V E R Y  F A C I L I T Y  S U M M A R Y  R E P O R T

Database  DD/Area  DSID  ----- Records Read -----  Records  Subord.  Final
Name      Name     DSID  IC      CA      LOG    Written  Reg Name  Status
HOTELDBA  HOTELDBA  1     N/A     N/A     N/A     N/A     N/A     Verify alloc complete
HOTELDBB  HOTELDBB  1     N/A     N/A     N/A     N/A     N/A     Verify alloc complete
HOTELDBC  HOTELDBC  1     N/A     N/A     N/A     N/A     N/A     Verify alloc complete
HOTELDBD  HOTELDBD  1     N/A     N/A     N/A     N/A     N/A     Verify alloc complete
HOTELDBE  HOTELDBE  1     N/A     N/A     N/A     N/A     N/A     Verify alloc complete
HOTELDBF  HOTELDBF  1     N/A     N/A     N/A     N/A     N/A     Verify alloc complete
1FRD0000I D A T A B A S E  R E C O V E R Y  F A C I L I T Y  02/25/2006 12:42
                                                    Page 2
    
```

- The next 2 slides captures 2 pages of report data identifying the final status of the VERIFY run.
- The 1<sup>st</sup> circle above identifies the type of DRF run this was for. In this case it was a VERIFY(ALLOC).
- The 2<sup>nd</sup> circle towards the bottom right shows that the overall status for each of the database data sets completed verification successfully.

## Case 1: DRF VERIFY report output (2 of 2)

```

Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY DRFV01  JOB02143  DSID   104 LINE 65      COLUMNS 02- 121
COMMAND INPUT ==>          SCROLL ==> CSR
DATABASE RECOVERY FACILITY DATA SET I/O REPORT

Recover to point: 2006.056 06:00:00.000000

Image Copy Data Set Name          Volume      IC DS      IC      ----- Time Stamp Range
----- Serial      Read Count  Type      1st Record

DAVK.HOTELDBA.IC1                 IMT156      0  STD      ALLOC RC = 0
DAVK.HOTELDBB.IC1                 IMT156      0  STD      ALLOC RC = 0
DAVK.HOTELDBC.IC1                 IMT156      0  STD      ALLOC RC = 0
DAVK.HOTELDBD.IC1                 IMT156      0  STD      ALLOC RC = 0
DAVK.HOTELDBE.IC1                 IMT156      0  STD      ALLOC RC = 0
DAVK.HOTELDBF.IC1                 IMT156      0  STD      ALLOC RC = 0

Change Accum Data Set Name        Volume      CA DS      ----- Time Stamp Range
----- Serial      Read Count  Type      1st Record

No data available for this type data set

Log Data Set Name                 Volume      Log DS      IMS      ----- Time Stamp Range
----- Serial      Read Count  SYSID     1st Record

IMTOOL1.SLDSP.IMK1.D06055.T2337108.V00  IMT191      0  IMK1      2006.055 23:37:10.8
Prilog: 2006.055 23:37:10.8           ALLOC RC = 0
IMTOOL1.SLDSP.IMK1.D06055.T2346163.V00  IMT170      0  IMK1      2006.055 23:46:16.3
Prilog: 2006.055 23:37:10.8           ALLOC RC = 0
IMTOOL1.SLDSP.IMK1.D06055.T2352121.V00  IMT042      0  IMK1      2006.055 23:52:12.1
Prilog: 2006.055 23:37:10.8           ALLOC RC = 0
IMTOOL1.SLDSP.IMK1.D06056.T0002270.V00  IMT051      0  IMK1      2006.056 00:02:27.0
Prilog: 2006.055 23:37:10.8           ALLOC RC = 0
IMTOOL1.SLDSP.IMK1.D06056.T0012348.V00  IMT154      0  IMK1      2006.056 00:12:34.8
Prilog: 2006.055 23:37:10.8           ALLOC RC = 0
IMTOOL1.SLDSP.IMK1.D06056.T0018313.V00  IMT110      0  IMK1      2006.056 00:18:31.3
Prilog: 2006.055 23:37:10.8           ALLOC RC = 0
IMTOOL1.SLDSP.IMK1.D06056.T0024275.V00  IMT025      0  IMK1      2006.056 00:24:27.5
Prilog: 2006.055 23:37:10.8           ALLOC RC = 0
IMTOOL1.SLDSP.IMK1.D06056.T0030235.V01  IMT128      0  IMK1      2006.056 00:30:23.5
Prilog: 2006.055 23:37:10.8           ALLOC RC = 0

1FRD0000I DATABASE RECOVERY FACILITY 02/25/2006 12:42 Page 4

```

- This final slide shows the individual return codes for each of the input data sets.
- In this case, each of the required image copies and logs were able to be allocated successfully as part of the verification process.

## Case 1: Now run a DRF recovery job for the 6 databases

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT          VLIM.DRF.JOBS(DRFF02) - 01.38          Columns 00001 00080
Command ==>                                         Scroll ==> CSR
000360 /**
000501 //FRXJCL01 EXEC FRXJCL3P                      00036000
000502 //SYSIN DD *                                  00050100
000503 REPORT (RPTTYPE=SEP, DRFUNIT=SYSDA, DRFHLQ=KCHENH1) 00050200
000505 SORTPARM(NUM(6))                              00050322
000506 DBDSL101(DISP(NEW))                            00050520
000509 ADD DB(HOTELDBA, -                             00050624
000510         HOTELDBB, -                             00050921
000511         HOTELDBC, -                             00051021
000512         HOTELDBD, -                             00051121
000513         HOTELDBE, -                             00051221
000514         HOTELDBF) -                             00051321
000515 DBATRB(DBDSL(101)) -                            00051422
000516 IC(COMP=N, -                                    00051526
000517         EXPDT=(2008100), -                        00051636
000518         ICBUF=(15), -                            00051736
000519         ICHLQ=(VLIMIC1), -                      00051836
000520         VOLSER(IMT055), -                       00051937
000521         UNIT(3390) -                             00052036
000522         SPACE=(CYL,1000,500,RLSE)) -           00052136
000523 PC(DUMPFORM=FORMAT, -                          00052236
000524         PRINTDATA=NO, -                          00052323
000525         RUNTM=YES, -                              00052423
000526         INTST=YES, -                            00052523
000527         BITMAP=YES, -                           00052623
000528         FSEMAP=YES, -                            00052723
000529         MAXFSD=YES, -                           00052823
000530         INTFS=YES) -                             00052923
000531 START ERROR(CONT DBRCMD(GLOBAL STACMD(GLOBAL - 00053023
000532 RCVTIME('06.056.06:00:00.0' PTRR_NOCHECK) 00053138
000540 /*                                             00053238
000540 /*                                             00054000
***** ***** Bottom of Data *****

```

- Now that we've verified that all ICs and Logs are ready we can proceed with recovery.
- This JCL will cover the remaining bullets listed in the beginning of this example:
  1. Issue an IMS /DBR command to ensure the databases are offline.
  2. Automatically delete and redefine your database data sets.
  3. Take a new image copy in parallel to recovery process.
  4. Run pointer checker to validate database contents in parallel to recovery process.
  5. Issue an IMS /STA command to start the databases after recovery completes.

## Case 1: DRF recovery report output (1 of 3)

```

Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY DRFF01  JOB02226  DSID   104 LINE 36          COLUMNS 02- 121
COMMAND INPUT ==> -          SCROLL ==> CSR
PROCESS : RCVDBDS           RECOVERY OPTIONS
SOURCE  : PRI              RCVTIME : Not Specified
READNUM : 10,10           TYPE   : Full
ERROR   : CONT            CHECK   : Yes
RCVTOKEN: DRFF01         TIME FMT: LCL
DRFPROC : DRFV32         LBI    : No
LOGNUM  : 10,10           ICNUM  : 10,10
OUTPUT  : Pro
SPSIZE  : 1024           CACHE  : No

----- Sort-Related Parameters -----
NUM      : 6              MAINSIZE: 100
FILSZ    : 400000        HIPRMAX : OPTIMAL
DYNALLOC : N/A , N/A    AVGRLEN : 1024
                                ASPREF  : FRXI

D A T A B A S E   R E C O V E R Y   F A C I L I T Y   S U M M A R Y   R E P O R T
Database  DD/Area  DSID  -----  Records  Read  -----  Records  Subord.  Final
Name      Name     DSID  IC       CA      LOG    Written  Reg Name  Status
HOTELDBA  HOTELDBA  1     3015    0      56982  3015    FRXI0001  Delete / define complete
HOTELDBB  HOTELDBB  1     3015    0      48648  3015    FRXI0002  Delete / define complete
HOTELDBC  HOTELDBC  1     3015    0      52761  3015    FRXI0003  Delete / define complete
HOTELDBD  HOTELDBD  1     3015    0      63360  3015    FRXI0004  Delete / define complete
HOTELDBE  HOTELDBE  1     3015    0      63282  3015    FRXI0005  Delete / define complete
HOTELDBF  HOTELDBF  1     3015    0      52752  3015    FRXI0006  Delete / define complete
FRD0000I  D A T A B A S E   R E C O V E R Y   F A C I L I T Y   02/25/2006 16:05
                                                Page 2

D A T A B A S E   R E C O V E R Y   F A C I L I T Y   U T I L I T Y   R E P O R T
Database  DD/Area  Database Data Set Name  IC  PC/OP  IB  PR  Utility Final Status
HOTELDBA  HOTELDBA  DAVK.DBSA.HOTEL      00  00    N/A N/A
HOTELDBB  HOTELDBB  DAVK.DBSB.HOTEL      00  00    N/A N/A
HOTELDBC  HOTELDBC  DAVK.DBSC.HOTEL      00  00    N/A N/A
HOTELDBD  HOTELDBD  DAVK.DBSD.HOTEL      00  00    N/A N/A
HOTELDBE  HOTELDBE  DAVK.DBSE.HOTEL      00  00    N/A N/A
HOTELDBF  HOTELDBF  DAVK.DBSF.HOTEL      00  00    N/A N/A

Final Return (RC) and Reason (RSN) Codes
---IC---  ---PC---  ---DP---  ---IB---  ---PR---
RC  RSN   RC  RSN   RC  RSN   RC  RSN   RC  RSN
00  00    00  00    N/A N/A  N/A N/A  N/A N/A
FRD0000I  D A T A B A S E   R E C O V E R Y   F A C I L I T Y   02/25/2006 16:05
                                                Page 3
    
```

- The DRF report still a good place to see the final status of the delete/define, image copy, and pointer checker options of our recovery job. See red circles above.

## Case 1: DRF recovery report output (2 of 3)

```

Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY DRFF01 JOB02226 DSID 106 LINE 2 COLUMNS 02- 121
COMMAND INPUT ==> SCROLL ==> CSR
NOTIFY.IC DBD(HOTELDBA ) DDN(HOTELDBA )
RUNTIME('2006.056 16:04:07.1 -08:00')
BATCH
ICDSN(VLIMIC1.IC1.HOTELDBA.HOTELDBA) ←
FILESEQ(0001) UNIT(3390)
VOLLIST(IMT187 )
RECDCT(0000003015)
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I COMMAND COMPLETION TIME 06.056 16:04:36.8
IMS VERSION 9 RELEASE 1 DATA BASE RECOVERY CONTROL PAGE 0002
NOTIFY.IC DBD(HOTELDBB ) DDN(HOTELDBB )
RUNTIME('2006.056 16:04:07.1 -08:00')
BATCH
ICDSN(VLIMIC1.IC1.HOTELDBB.HOTELDBB) ←
FILESEQ(0001) UNIT(3390)
VOLLIST(IMT051 )
RECDCT(0000003015)
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I COMMAND COMPLETION TIME 06.056 16:04:37.0
IMS VERSION 9 RELEASE 1 DATA BASE RECOVERY CONTROL PAGE 0003
NOTIFY.IC DBD(HOTELDBC ) DDN(HOTELDBC )
RUNTIME('2006.056 16:04:07.1 -08:00')
BATCH
ICDSN(VLIMIC1.IC1.HOTELDBC.HOTELDBC) ←
FILESEQ(0001) UNIT(3390)
VOLLIST(IMT206 )
RECDCT(0000003015)
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I COMMAND COMPLETION TIME 06.056 16:04:37.3
IMS VERSION 9 RELEASE 1 DATA BASE RECOVERY CONTROL PAGE 0004
NOTIFY.IC DBD(HOTELDBD ) DDN(HOTELDBD )
RUNTIME('2006.056 16:04:07.1 -08:00')
BATCH
ICDSN(VLIMIC1.IC1.HOTELDBD.HOTELDBD) ←
FILESEQ(0001) UNIT(3390)
VOLLIST(IMT079 )
RECDCT(0000003015)
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I COMMAND COMPLETION TIME 06.056 16:04:37.5
IMS VERSION 9 RELEASE 1 DATA BASE RECOVERY CONTROL PAGE 0005
NOTIFY.IC DBD(HOTELDBE ) DDN(HOTELDBE )
RUNTIME('2006.056 16:04:07.1 -08:00')
BATCH
ICDSN(VLIMIC1.IC1.HOTELDBE.HOTELDBE) ←
FILESEQ(0001) UNIT(3390)
VOLLIST(IMT209 )
RECDCT(0000003015)
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I COMMAND COMPLETION TIME 06.056 16:04:37.7
IMS VERSION 9 RELEASE 1 DATA BASE RECOVERY CONTROL PAGE 0006
NOTIFY.IC DBD(HOTELDBF ) DDN(HOTELDBF )
RUNTIME('2006.056 16:04:07.2 -08:00')
BATCH
ICDSN(VLIMIC1.IC1.HOTELDBF.HOTELDBF) ←
FILESEQ(0001) UNIT(3390)
VOLLIST(IMT183 )
RECDCT(0000003015)
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I COMMAND COMPLETION TIME 06.056 16:04:37.9
IMS VERSION 9 RELEASE 1 DATA BASE RECOVERY CONTROL PAGE 0007
DSP0211I COMMAND PROCESSING COMPLETE
DSP0211I HIGHEST CONDITION CODE = 00
IMS HIGH PERFORMANCE IMAGE COPY
5655-K96 "ICEIN STATEMENTS REPORT"
DATE: 02/25/2006 TIME: 16.03.03

```

- Since HPIC is a separate tool by itself it will generate its own report output.
- When you run HPIC as part of your recovery job, DRF will capture the HPIC output and append it to the end of the DRF reports.
- What you see here is only a portion of that output.
- The red arrows on the left are pointing out the 6 new image copies we requested to be generated.
- They are registered to the RECON.

## Case 1: DRF recovery report output (3 of 3)

```

Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY DRFF01   JOB02226   DSID   111 LINE 1           COLUMNS 01- 120
COMMAND INPUT ==>           SCROLL ==> CSR
1IMS HIGH PERFORMANCE POINTER CHECKER FOR z/OS           "HASH EVALUATION REPORT"
5655-K53                                           DATE: 02/25/2006   TIME: 16.03.15           FAB

FABP2001I EVAL OF DB: HOTELDBA DB#: 001           DSG#: 01 COMPLETED ERRORS: 0 TOTAL ( 0 SEV. 0 PHY.
FABP2001I EVAL OF DB: HOTELDBB DB#: 003           DSG#: 01 COMPLETED ERRORS: 0 TOTAL ( 0 SEV. 0 PHY.
FABP2001I EVAL OF DB: HOTELDBC DB#: 005           DSG#: 01 COMPLETED ERRORS: 0 TOTAL ( 0 SEV. 0 PHY.
FABP2001I EVAL OF DB: HOTELDBD DB#: 007           DSG#: 01 COMPLETED ERRORS: 0 TOTAL ( 0 SEV. 0 PHY.
FABP2001I EVAL OF DB: HOTELDBE DB#: 009           DSG#: 01 COMPLETED ERRORS: 0 TOTAL ( 0 SEV. 0 PHY.
FABP2001I EVAL OF DB: HOTELDBF DB#: 00B           DSG#: 01 COMPLETED ERRORS: 0 TOTAL ( 0 SEV. 0 PHY.
0FABP2002I                                           RUN COMPLETED ERRORS: 0 TOTAL
0FABP2003I NO ERRORS DETECTED
1IMS HIGH PERFORMANCE POINTER CHECKER FOR z/OS           "HD POINTER CHECKER SUMMARY"
5655-K53                                           DATE: 02/25/2006   TIME: 16.03.15           FAB

ODBNOME/      DDNAME/      C-DATE/      D-DATE/      D-TIME      CHK-DATE/  CHK-TIME/  DATA-SET  SIZE      F-SPACE  %/  DETE
DB# DSG# DBLG# DB-ORGANIZATION ACCM BLKSZ LRECL DBTYPE DEVICE %SEGMS IN OFLW  CYL'S      BYTES      BYTES      TOTA
-----
0HOTELDBA      HOTELDBA      02/25/2006  02/25/2006  16.03.15   02/25/2006  16.03.15   51 37048320  14033408  37 %
001 01 001* HDAM          OSAM 12288 12288 REAL 3390 0
0HOTELDBB      HOTELDBB      02/25/2006  02/25/2006  16.03.15   02/25/2006  16.03.15   51 37048320  14033408  37 %
003 01 002* HDAM          OSAM 12288 12288 REAL 3390 0
0HOTELDBC      HOTELDBC      02/25/2006  02/25/2006  16.03.15   02/25/2006  16.03.15   51 37048320  14033408  37 %
005 01 003* HDAM          OSAM 12288 12288 REAL 3390 0
0HOTELDBD      HOTELDBD      02/25/2006  02/25/2006  16.03.15   02/25/2006  16.03.15   51 37048320  14033408  37 %
007 01 004* HDAM          OSAM 12288 12288 REAL 3390 0
0HOTELDBE      HOTELDBE      02/25/2006  02/25/2006  16.03.15   02/25/2006  16.03.15   51 37048320  14033408  37 %
009 01 005* HDAM          OSAM 12288 12288 REAL 3390 0
0HOTELDBF      HOTELDBF      02/25/2006  02/25/2006  16.03.15   02/25/2006  16.03.15   51 37048320  14033408  37 %
00B 01 006* HDAM          OSAM 12288 12288 REAL 3390 0
-NOTE: - '*' AFTER THE DBLG# INDICATES THAT SOME OTHER DATABASES WITH THE SAME
          DATABASE LOGICAL GROUP WERE NOT PROCESSED BY THIS RUN.
1HOTELDBE.HOTELDBE  RSS=FRXI0005 HPPCd=STATIPRT
-----
SAS COPY->: DD=STATIPR@ DSN=KCHENH1.HOTELDBE.HOTELDBE.STATIPRT.T1604126
1IMS HIGH PERFORMANCE POINTER CHECKER FOR z/OS           "SEPARATOR PAGE FOR DB/DSG"
5655-K53                                           DATE: 02/25/2006   TIME: 16.03.15           FAB

```

- DRF will capture HPPC output and append it to the end of the DRF reports.
- This is only a portion of that output but gives you an idea that our recovery produced no pointer errors.

## Case 1: Are we done?

```

Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY DRFV01  JOB02234  DSID   104 LINE 43      COLUMNS 02- 121
COMMAND INPUT ==> _      SCROLL ==> CSR

DATABASE RECOVERY FACILITY UTILITY REPORT

Database DD/Area Database Data Set Name IC PC/DP IB PR Utility Final Status
HOTELDBA HOTELDBA DAVK.DBSA.HOTEL N/A N/A N/A N/A
HOTELDBB HOTELDBB DAVK.DBSB.HOTEL N/A N/A N/A N/A
HOTELDBC HOTELDBC DAVK.DBSC.HOTEL N/A N/A N/A N/A
HOTELDBD HOTELDBD DAVK.DBSD.HOTEL N/A N/A N/A N/A
HOTELDBE HOTELDBE DAVK.DBSE.HOTEL N/A N/A N/A N/A
HOTELDBF HOTELDBF DAVK.DBSF.HOTEL N/A N/A N/A N/A

Final Return (RC) and Reason (RSN) Codes

---IC--- ---PC--- ---DP--- ---IB--- ---PR---
RC RSN RC RSN RC RSN RC RSN RC RSN
00 00 00 00 00 00 N/A N/A N/A N/A
1FRD0000I DATABASE RECOVERY FACILITY 02/25/2006 17:35 Page 3

DATABASE RECOVERY FACILITY DATA SET I/O REPORT

Recover to point: Not Specified

Image Copy Data Set Name Volume IC DS IC Time Stamp Range
Serial Read Count Type 1st Record
VLIMIC1.IC1.HOTELDBA.HOTELDBA IMT187 0 STD ALLOC RC = 0
VLIMIC1.IC1.HOTELDBB.HOTELDBB IMT051 0 STD ALLOC RC = 0
VLIMIC1.IC1.HOTELDBC.HOTELDBC IMT206 0 STD ALLOC RC = 0
VLIMIC1.IC1.HOTELDBD.HOTELDBD IMT079 0 STD ALLOC RC = 0
VLIMIC1.IC1.HOTELDBE.HOTELDBE IMT209 0 STD ALLOC RC = 0
VLIMIC1.IC1.HOTELDBF.HOTELDBF IMT183 0 STD ALLOC RC = 0

Change Accum Data Set Name Volume CA DS Time Stamp Range
Serial Read Count 1st Record

No data available for this type data set

Log Data Set Name Volume Log DS IMS Time Stamp Range
Serial Read Count SYSID 1st Record

No data available for this type data set
***** BOTTOM OF DATA *****
F1=HELP F2=SPIT F3=END F4=RETURN F5=ISPF F6=PAUSE F7=UP F8=DOWN F9=STOP F10=LEFT F11=RIGHT F12=DELETE

```

- Recovery completed and we are done.
- Remember that as part of our recovery we generated new image copies.
- Of course we can always generate a RECON listing and look up each image copy data set to see if it is actually there.
- I just want to point out that you can just as simply run another VERIFY job to see what happens.
- As you can see from the right, DRF identified your new image copies and the most current ones to use for a future recovery.
- No logs are needed since they are all included in your ICs.

## Case 1: Case of the Bad REORG Utility - Lessons Learned

- **What did DRF help you accomplish?**
- **VERIFY**
  - ▶ Ran a VERIFY job prior to the actual recovery to ensure the required input data sets (image copies, change accumulation, and logs) for recovery are accessible.
  - ▶ Useful to provide a list of recovery inputs for offsite, disaster, or local recovery.
- **IMS Commands**
  - ▶ Via TOI/XCF, DRF can issue /DBR and /STA that synchronize IMS with batch recovery.
  - ▶ /DBR local or global for all databases or areas prior to recovery.
  - ▶ /STA local or global for all databases or areas after recovery completes.
- **Delete/redefine**
  - ▶ Automatically delete and redefine your database data sets as part of recovery step.
- **HPIC**
  - ▶ Integrate your image copy process in parallel to recovery.
- **HPPC**
  - ▶ Integrate your pointer checker process in parallel to recovery.



## Case 2: Alternative to Testing with Production Database

- **As your team learned the hard way... maybe it's not such a good idea to test Beta code on your production system.**
- **Now that you've put out the fire and had a good 4 hours of sleep it's now time to go back to work.**
- **There's still the matter of testing the new REORG tool.**
  
- **Hmm...**
  - ▶ It's definitely not a good idea to test the tool against your production databases.
  - ▶ After last night's down time you wish to avoid disrupting your databases as much as possible.
  
- **What can you do?**

## Case 2: Alternative to Testing with Production Database - Solving the Case

- **We will show you how a DRF batch job can help you:**
  - ▶ Create duplicate copies of your database data sets from your image copy, change accumulation, and log inputs (Recovery to Copy).
  - ▶ Identify your new duplicate database in a new report section.



## Case 2: SYSIN to run a DRF recovery job for Recover to Copy

- This JCL will cover the basics of what you'll need to generate database copies:

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT      VLIM.DRF.JOBS(DRFPCY01) - 01.24      Columns 00001 00080
Command ==> _____ Scroll ==> CSR
000355 /*-----*
000356 //PRPRINT DD SYSOUT=&SOUT,&DCB                00035600
000357 /*                                           00035700
000358 //                                           00035800
000360 /*                                           00036000
000501 //FRXJCL01 EXEC FRXJCL3P                    00050100
000502 //SYSIN DD *                                00050200
000506 REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=DRFPCY01) 00050623
1 ← 000507 OUTPUT(DUP)                             00050722
000508 DBDSN101('TEST') →                          00050820
000509 DBDSN102('TEST.HOTELDBA.V2') → 2             00050924
3 ← 000510 DBDSN103('AUDIT',,,%DATE) →             00051024
000511 DBDSL101(DISP(NEW))                          00051124
3 ← 000512 DBDSL102(DISP(OLD))                      00051224
000513 ADD DB(HOTELDBA, -                           00051324
000514 HOTELDBB) -                                  00051424
4 ← 000515 DBATR(BDBSN(101),DBDSL(101)) -           00051524
000517 PC(DUMPFOM=FORMAT, -                          00051723
000518 PRINTDATA=NO, -                                00051823
000519 RUNTM=YES, -                                  00051923
000520 INTST=YES, -                                  00052023
000521 BITMAP=YES, -                                 00052123
000522 FSEMAP=YES, -                                00052223
000523 MAXFSD=YES, -                                00052323
000524 INTFS=YES)                                   00052423
000525 START ERROR(CONT)                            00052515
000530 /*                                           00053000
***** ***** Bottom of Data *****
```

1. Specify OUTPUT(DUP) statement to identify that DRF will perform a Recover to Copy operation.
2. DBDSN statements set up different data set naming convention rules.
3. DBDSL statements identify different allocation methods. DISP(NEW) identifies that DRF will allocate/generate the data set name. DISP(OLD) identifies that the user pre-allocated/generated the name for us.
4. DBATR lets you pick which DBDSN & DBDSL combination to use.

## Case 2: Recover to Copy report output

```

Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY DRFCPY01 JOB02240 DSID 104 LINE 27 COLUMNS 02- 121
COMMAND INPUT ==> SCROLL ==> CSR
D A T A B A S E R E C O V E R Y F A C I L I T Y R E C O V E R Y P A R A M E T E R S

PROCESS : RCVCOPY RECOVERY OPTIONS
SOURCE : PRI RCVTIME : Not Specified
READNUM : 10,10 TYPE : Full LOGNUM : 10,10 ICNUM : 10,10
ERROR : CONT CHECK : Yes OUTPUT : Dup
RCVTOKEN: DRFCPY01 TIME FMT: LCL
DRFPROC : DRFV32 LBI : No SPSIZE : 1024 CACHE : No

----- Sort-Related Parameters -----
NUM : 10 MAINSIZE: 100 AVGRLEN : 1024
FILSZ : 400000 HIPRMAX : OPTIMAL ASPREF : FRXI
DYNALLOCC: N/A , N/A

D A T A B A S E R E C O V E R Y F A C I L I T Y S U M M A R Y R E P O R T

Database DD/Area DSID IC Records Read LOG Records Subord. Final
Name Name Name CA CA Written Name Status
HOTELDBA HOTELDBA 1 3015 0 0 0 FRXI0001 No errors encountered
HOTELDBB HOTELDBB 1 3015 0 0 0 FRXI0002 No errors encountered

D A T A B A S E R E C O V E R Y F A C I L I T Y R C V C O P Y R E P O R T

Database DD/Area New Data Set Name Records Volume
Name Name Name Copied Serial
HOTELDBA HOTELDBA TEST.DAVK.DBSA.HOTEL 3015 IMT156
HOTELDBB HOTELDBB TEST.DAVK.DBSB.HOTEL 3015 IMT156
1FRD00001 D A T A B A S E R E C O V E R Y F A C I L I T Y 02/25/2006 21:44 Page 2

D A T A B A S E R E C O V E R Y F A C I L I T Y U T I L I T Y R E P O R T

Database DD/Area Database Data Set Name IC PC/DP IB PR Utility Final Status
HOTELDBA HOTELDBA DAVK.DBSA.HOTEL N/A 00 N/A N/A
HOTELDBB HOTELDBB DAVK.DBSB.HOTEL N/A 00 N/A N/A

```

- This report output looks similar to the one in the first case scenario.
- Difference is the indication that this is for Recover to Copy and names of the new DB copies are returned.

## Case 2: Alternative to Testing with Production Database - Lessons Learned

- **What did DRF help you accomplish?**
  
- **Recover to Copy**
  - ▶ Create copies of database data sets using image copy, change accumulation, and log data sets.
  - ▶ Test bed generation
  - ▶ Audit databases
  - ▶ Use naming convention rules to generate data set names or pre-define yourself.



## Case 3: Updating your Image Copies

- **One of the things you've noticed is that you do not take more frequent image copies.**
  - **Recoveries tend to take much longer due to the amount of log data to apply.**
  - **Keeping all those log data sets around and having to ensure you send all of them to other sites can be troublesome.**
- 
- **What can you do?**



## Case 3: Updating your Image Copies - Solving the Case

- **We will show you how a DRF batch job can help you:**
  - ▶ Bring your image copies up to date.
  - ▶ Generate an Incremental Image Copy using your previous image copy and log data.



## Case 3: SYSIN to generate Incremental Image Copies

- This JCL will cover the basics of what you'll need to generate Incremental Image Copies:

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT      VLIM.DRF.JOBS(DRFICR01) - 01.27      Columns 00001 00080
Command ==> _____ Scroll ==> CSR
000355 /*-----* 00035500
000356 //PRPRINT DD SYSOUT=&SOUT,&DCB 00035600
000357 /* 00035700
000358 // PEND 00035800
000360 /* 00036000
000501 //FRXJCL01 EXEC FRXJCL3P 00050100
000502 //SYSIN DD * 00050200
000506 REPORT (RPTTYPE=SEP, DRFUNIT=SYSDA, DRFHQ=DRFICR01) 00050625
000507 OUTPUT (ICR) 00050719
1 000509 ADD DB(HOTELDBI, - 00050927
2 000510 HOTELDBJ) - 00051027
000517 IC (COMP(Y), - 00051722
000518 COMPRTN(FABJCOMP3), - 00051822
000519 STORCLAS(GSYES), - 00051922
000520 ICNMRULE=Y, - 00052025
000521 ICHLQ(ICR01), - 00052125
000522 VOLSER=(IMT112), - 00052225
000523 UNIT=(3390), - 00052325
000524 SPACE(CYL,100,50,RLSE) 00052422
000525 START ERROR(CONT) 00052522
000530 /* 00053022
***** Bottom of Data *****

```

- Specify OUTPUT(ICR) statement to identify that DRF will perform a Incremental Image Copy operation.
  - We are using the same IC() keyword and HPIC interface to generate these image copies.
- Actual database data sets are not touched.



## Case 3: New image copies generated and registered to RECON

```

Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY DRFICR01 JOB02249 DSID 105 LINE 8 COLUMNS 02- 121
COMMAND INPUT ==> SCROLL ==> CSR
-----
SAS COPY->: DD=FRXWT@ DSN=DRFICR01.HOTELDBJ.HOTELDBJ.FRXTORS.T2229139
FABJDRQS: FUNC=OPEN STARTED
FABJ0182E DDNAME SPECIFIED IN JCL, BUT SUPPORT DYNALLOC ONLY
FABJDRQS: FUNC=CLOSE ENDED
IMS VERSION 9 RELEASE 1 DATA BASE RECOVERY CONTROL PAGE 0001
NOTIFY.IC DBD(HOTELDBI ) DDN(HOTELDBI ) -
RUNTIME('2006.056 22:29:13.1 -08:00') -
BATCH -
ICDSN(ICR01.IC1.HOTELDBI.HOTELDBI.D06056.T222840) -
UNIT(3390) -
VOLLIST(IMT112 ) -
RECDCT(0000000000)
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I COMMAND COMPLETION TIME 06.056 22:29:24.3
IMS VERSION 9 RELEASE 1 DATA BASE RECOVERY CONTROL PAGE 0002
NOTIFY.IC DBD(HOTELDBJ ) DDN(HOTELDBJ ) -
RUNTIME('2006.056 22:29:13.1 -08:00') -
BATCH -
ICDSN(ICR01.IC1.HOTELDBJ.HOTELDBJ.D06056.T222840) -
UNIT(3390) -
VOLLIST(IMT112 ) -
RECDCT(0000000000)
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I COMMAND COMPLETION TIME 06.056 22:29:24.5
IMS VERSION 9 RELEASE 1 DATA BASE RECOVERY CONTROL PAGE 0003
DSP0211I COMMAND PROCESSING COMPLETE
DSP0211I HIGHEST CONDITION CODE = 00
IMS HIGH PERFORMANCE IMAGE COPY "ICEIN STATEMENTS REPORT"
5655-N45 DATE: 02/25/2006 TIME: 22.28.40

```

- Since we called HPIC to generate this image copy, the HPIC reports are sent back to DRF.
- Just like in the first case scenario, these image copies are registered to RECON.

## Case 3: VERIFY report output showing new image copies

```

Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY DRFV03   JOB02252  DSID   104 LINE 27       COLUMNS 02- 121
COMMAND INPUT ==>          SCROLL ==>  CSR
D A T A B A S E   R E C O V E R Y   F A C I L I T Y   S U M M A R Y   R E P O R T

Database   DD/Area   DSID   -----  Records Read -----  Records   Subord.   Final
Name       Name      1      IC        CA        LOG      Written   Reg Name  Status
HOTELDBI   HOTELDBI   1      N/A      N/A      N/A      N/A      N/A      Verify alloc failure
HOTELDBJ   HOTELDBJ   1      N/A      N/A      N/A      N/A      N/A      Verify alloc failure
1FRD0000I  D A T A B A S E   R E C O V E R Y   F A C I L I T Y   02/25/2006 22:39
                                           Page 2

D A T A B A S E   R E C O V E R Y   F A C I L I T Y   U T I L I T Y   R E P O R T

Database   DD/Area   Database Data Set Name      IC  PC/DP  IB  PR  Utility Final Status
HOTELDBI   HOTELDBI   DAVK.VDBSI.HOTEL          N/A N/A   N/A N/A
HOTELDBJ   HOTELDBJ   DAVK.VDBSJ.HOTEL          N/A N/A   N/A N/A

Final Return (RC) and Reason (RSN) Codes

---IC---   ---PC---   ---DP---   ---IB---   ---PR---
RC  RSN    RC  RSN    RC  RSN    RC  RSN    RC  RSN
00  00     00  00     00  00     N/A N/A     N/A N/A
1FRD0000I  D A T A B A S E   R E C O V E R Y   F A C I L I T Y   02/25/2006 22:39
                                           Page 3

D A T A B A S E   R E C O V E R Y   F A C I L I T Y   D A T A   S E T   I / O   R E P O R T

Recover to point: Not Specified

Image Copy Data Set Name      Volume   IC DS   IC   -----  Time Stamp Range
-----  Serial  Read Count  Type  1st Record
ICR01.IC1.HOTELDBI.HOTELDBI.D06056.T222840  IMT112  0  STD  ALLOC RC = 4
ICR01.IC1.HOTELDBJ.HOTELDBJ.D06056.T222840  IMT112  0  STD  ALLOC RC = 4

```

- Maybe it'll be a good idea to run a VERIFY again.
- Ah ha! DRF was able to locate the new ICs from RECON but produced a failure during the VERIFY
- We find out later that someone went in and deleted the image copies. Good thing we checked.

## Case 3: Updating your Image Copies - Lessons Learned

- **What did DRF help you accomplish?**
  
- **Incremental Image Copy**
  - ▶ Create more up to date image copies using a previous image copy, change accumulation, and log data sets.
  - ▶ No production database access.
  - ▶ Image copies registered with DBRC



# Resources

- **Tools used in this presentation:**

- ▶ **IMS Database Recovery Facility for z/OS**

IMS Database Recovery Facility is a high-performance, state-of-the-art database recovery product supporting all recoverable IMS databases. It is the follow-on version of IMS Online Recovery Service for z/OS, contains all the features and functions of that product, and adds several key features including IMS Database Recovery Facility initiation by submitting an MVS batch job and batch invocation of the IMS Database Recovery Facility without requiring an active IMS Region.

<http://www-306.ibm.com/software/data/db2imstools/imstools/imsonlinerecov.html>

- ▶ **IMS High Performance Image Copy for z/OS**

IBM IMS High Performance Image Copy enables you to run the image copy function with the hash checking of IMS High Performance Pointer Checker for z/OS, under the control of IMS Parallel Reorganization for z/OS if these tools are part of your IMS environment. As a result, with IMS HP Image Copy, you can reduce the run-time of the reorganization process for an IMS full function and/or HALDB database and you can check the accuracy of an image copy while these processes are running in parallel. New with V4 of IMS HP Image Copy is Concurrent Copy support for databases that reside on devices supporting this feature, providing point-in-time data consistency. Also offered is SnapShot and FlashCopy support providing rapid backup and rapid recovery of a database when needed.

<http://www-306.ibm.com/software/data/db2imstools/imstools/imsice.html>

- ▶ **IMS High Performance Pointer Checker for z/OS**

IBM IMS High Performance Pointer Checker helps you analyze, diagnose, and repair corrupt databases quickly. It generates numerous reports that facilitate system tuning. When IMS HP Pointer Checker is used in conjunction with IMS Database Repair Facility, the tools work together to help detect and correct database errors and repair them with a minimum of downtime.

<http://www-306.ibm.com/software/data/db2imstools/imstools/imshpptchkr.html>

# Next Steps

- Contact your local IBM sales specialist
- Visit us at IMS Technical Conferences worldwide.
- Email contact: Dianne McCallum – [dmccallu@us.ibm.com](mailto:dmccallu@us.ibm.com)
- <http://www.ibm.com/software/data/db2imstools/>

