

The ABC's of Disaster Recovery

Speaker: Terri Grissom
BMC Software, Inc.

Session Code: F09
Wednesday, 16 October 2013 8:30 am - 9:30 am | Platform: DB2 for z/OS



Abstract:

We will go thru a disaster recovery drill process from A to Z. Our discussion will start with an overview of the fundamentals of DB2 Infrastructure: the catalog and directory, the BSDS, and the active and archive logs. We will then discuss the preparation that is needed at the local site prior to the drill, followed by a deep dive into the jobs that will be run to recover the DB2 subsystem at the recovery site.

Along the way you will see how the Change Log Inventory program is used to update the BSDS with archive logs, generate the Conditional Restart Record, and add active logs. In addition, we will discuss some zparm options and some other options available such as restoring archive copies to disk, or formatting the active logs.

We will then look at the job necessary to recover the catalog and directory, and we will see how this job changes based on the presence and type of user-defined catalog indexes. Our discussion will also highlight the differences in the process in a non-datasharing system versus a datasharing system.

By the time we are finished, you will fully understand the steps necessary to recover your DB2 subsystem at the disaster recovery site. All that remains is for you to recover your application objects!

Agenda – The ABC's of Disaster Recovery

- The ABC's of DB2 Infrastructure
 - **A**: Active Logs and Archive Logs
 - **B**: Bootstrap Data Set
 - **C**: Catalog and Directory
- Disaster Recovery Drill preparation required at local site
- Deep Dive into jobs that will recover the DB2 subsystem
- Discuss impact of non-datasharing versus datasharing
- Discuss impact of user-defined indexes on DB2 catalog



Acknowledgements / Disclaimers

IBM®, DB2®, z/OS® are registered service marks and trademarks of International Business Machines Corporation, in the United States and/or other countries

DSN1LOGP, DSNJU003, DSNJU004 output references contained are also of IBM.

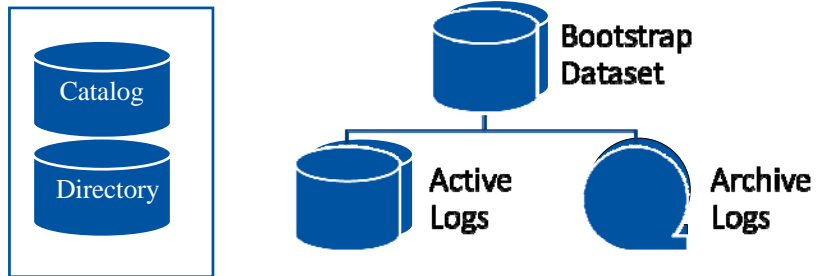
The information contained in this presentation has not been submitted to any formal review and is distributed on an “As Is” basis.

References:

DB2 10 for z/OS Administration Guide SC19-2968

DB2 10 for z/OS Utility Guide SC19-2984

ABC's of DB2 Infrastructure



These are the objects that make up the heart of DB2: the catalog and directory, the bootstrap dataset, and the active and archive logs.

When we do our deep dive into the job that will recover our DB2 infrastructure, these are the objects we will be focusing on.

First we'll look at the Catalog and Directory.

ABC's of DB2 Infrastructure: Catalog & Directory



- Database DSNDB06
- Record data about everything defined to DB2
- V10: 95 tablespaces
- V11: 108 tablespaces
- You can SELECT from all catalog tables



- Database DSNDB01
- Contains information that DB2 uses during normal operations
- V10: 8 tablespaces
- V11: 8 tablespaces
- In V10 NFM, PTF UK73478 enables SELECT's against the directory

Catalog tables record data about everything defined to the DB2 system.

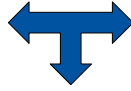
Some examples: Databases, tablespaces, tables, indexes, views, storage groups, authorizations, copies of tablespaces and indexes.

V10 NFM – APAR PM35190 / PTF UK73478 with subsequent CATMAINT – defines the structure of the directory tables in the catalog.

The description of this PTF states that it enables you to Select against SYSLGRNX, SYSUTIL, and SYSUTILX.

ABC's of DB2 Infrastructure: Catalog (DSNDB06)

Create Tablespace
Schedule in Idugdb
...



Create Table Idug.Schedule
(Idugid Char(3),
Title Char(40)) In Idugdb.Schedule;

DSNDB06 Table space	Table	Action
DSNDB06.SYSTSTSP	SYSIBM.SYSTABLESPACE	Insert row for tablespace
DSNDB06.SYSTSTPT	SYSIBM.SYSTABLEPART	Insert 1 row for non-partitioned ts Insert 1 row per part of a part. ts
DSNDB06.SYSRTSTS	SYSIBM.SYSTABLESPACESTAT	Insert row for RTS for ts
DSNDB06.SYSCOPY	SYSIBM.SYSCOPY	Insert row with ICTYPE=C (create) for tablespace
DSNDB06.SYSTSTAB	SYSIBM.SYSTABLES	Insert row for table
DSNDB06.SYSTSCOL	SYSIBM.SYSCOLUMNS	Insert 2 rows. 1 for each table column
DSNDB06.SYSTSTAU	SYSIBM.SYSTABAUTH	Insert row for table privileges

Example of catalog tables that are updated by a Create Tablespace and Create Table

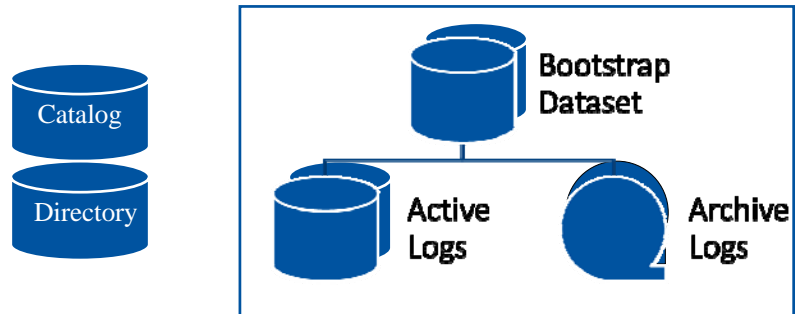
ABC's of DB2 Infrastructure: Directory



Examples of some Directory tablespaces:

DSNDB01 Tablespace	Function
DBD01	Contains internal information, database descriptors (DBD), about the databases in the DB2 subsystem.
SYSUTILX	Contains a row for every utility job that is running. Used by DB2 when restarting abended utility runs.
SYSLGRNX	Tracks the opening (for update) and closing of table spaces, indexes, or partitions. Used to reduce recovery time by reducing amount of log to scan.

ABC's of DB2 Infrastructure



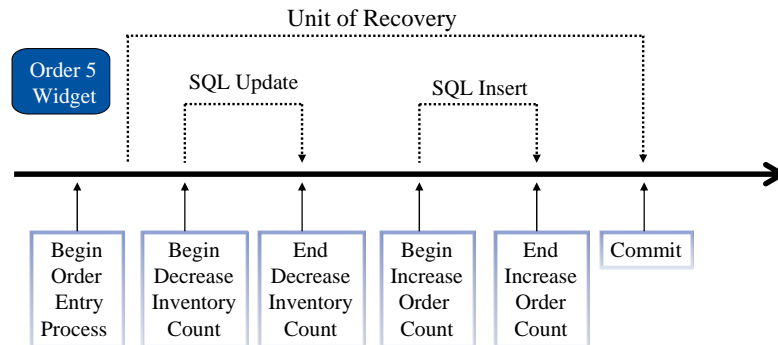
DB2 records all data changes and significant events in the logs.

There is 1 **current** active log at any point in time that is written to.

When the active is full, the active and the BSDS are offloaded to an archive log.

In order to understand what the logs are for, we need to discuss the concept of a unit of recovery in DB2.

ABC's of DB2 Infrastructure: Logging Theory - Unit of Recovery



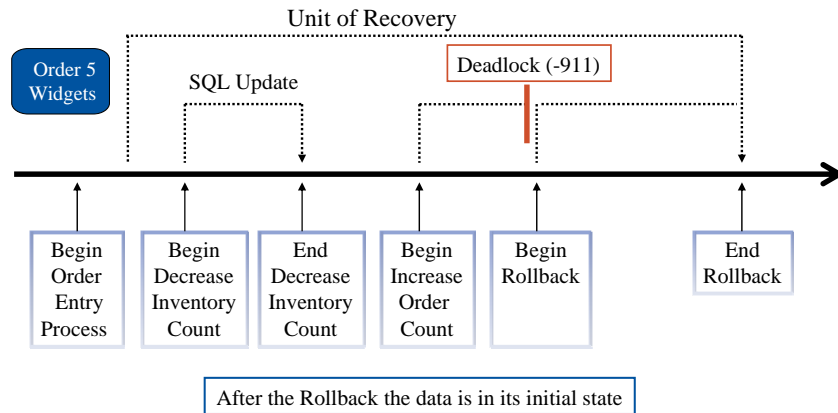
Unit of Recovery: Work that changes DB2 data from one point of consistency to another

The entire unit of recovery is recorded in the log.

DB2 logs all of this information for recovery purposes of the application data.

The unit of recovery is to keep the data consistent.

ABC's of DB2 Infrastructure: Logging Theory - ROLLBACK



This is the same example as in the previous slide except a problem is encountered during the second update.

If a failure occurs within a unit of recovery, DB2 rolls back (backs out) any changes to the data.

ABC's of DB2 Infrastructure: Bootstrap Dataset

“Pull yourself up by your bootstraps”

Bootstrapping or booting:

Metaphor which refers to a self-sustaining process that proceeds without external help.



The Bootstrap Dataset is essential to the “booting” of DB2

DB2 will not come up without the Bootstrap Dataset (BSDS). It is essential in the ‘booting’ or startup process of DB2.

ABC's of DB2 Infrastructure: Bootstrap Dataset (BSDS)



- Contains **critical information** used by DB2:
 - during startup and
 - for any activity that requires reading log
- DB2 will not come up without a valid BSDS
- VSAM key-sequenced data set (KSDS)
- DB2 creates 2 copies of the BSDS during installation
 - Datasharing: Each member has a BSDS



ABC's of DB2 Infrastructure: Bootstrap Dataset

- DSNJU004 – Print Log Map Utility – Used to print the contents of the BSDS

```
//BSDS01 EXEC PGM=DSNJU004  
//STEPLIB DD DISP=SHR,DSN=SYS3.DEFS.DSNEXIT  
// DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD  
//GROUP DD DISP=SHR,DSN=DEFSCAT.BSDS01  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
MEMBER *
```

VSAM
BSDS

Datasharing: Specify one of the member BSDS datasets in the GROUP DD and the BSDS for all members will be printed

In datasharing, each datasharing member has its own Bootstrap Dataset.

ABC's of DB2 Infrastructure: Bootstrap Dataset Active Log Section of BSDS

ACTIVE LOG COPY 1 DATA SETS					
START RBA/TIME	END RBA/TIME	DATE	LTIME	DATA SET INFORMATION	
00785B013000 2013.169 16:38:57.4	00785F662FFF 2013.169 16:39:40.4	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS01 STATUS=REUSABLE	
00785F663000 2013.169 16:39:40.4	007863CB2FFF 2013.169 20:14:44.9	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS03 PASSWORD=(NULL) STATUS=REUSABLE	
007863CB3000 2013.169 20:14:44.9	007866C82FFF 2013.169 21:22:10.6	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS04 PASSWORD=(NULL) STATUS=REUSABLE	
007866C83000 2013.169 21:22:10.6	0078678BEFFF 2013.192 14:47:46.9	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS05 PASSWORD=(NULL) STATUS=TRUNCATED, REUSABLE	
0078678BF000 2013.192 14:47:46.9	00786BFC2FFF	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS02 PASSWORD=(NULL) STATUS=NOTREUSABLE	

STATUS	
Reusable	Dataset has been offloaded (archived). Ready to be reused as current active log.
NotReusable	Dataset has not been offloaded (archived)
Truncated	Dataset was truncated (didn't write to End Rba) when it was the current active log. Ex: Archive Log command issued

DS02 is the **Current** Active Log: Status NotReusable and End Time of Blanks

One of the functions of the BSDS is to manage the active and archive logs.

The top of the output shows that this is the listing for the Active Log Copy 1 Datasets. It is strongly recommended the you have dual active logs.

The use of dual active logs increases availability as well as the reliability of recovery by eliminating a single point of failure.

In this case we have 5 active log copy 1 datasets. Another section of the BSDS would show the 5 active log copy 2 datasets.

To the right is the active log data set name along with its status, and to the left is the Start RBA and End RBA along with the corresponding timestamp.

The last active log listed has a status of NOTREUSABLE and the End Timestamp is blank. This indicates that this is the current active log

ABC's of DB2 Infrastructure: Active Logs

- DB2 registers data changes and significant events as they occur in the active log.
- Active logs are VSAM linear datasets.
- You may have multiple active log datasets – up to 93 in V10
 - DB2 designates one of the actives as the **current** active log
 - Status: NotReusable and End Time: Blank
 - When the current active log is full, DB2 switches to another active log.
 - OFFLOAD triggered to create 2 files:
 - BSDS is copied to an Archive Log – the 'B' file
 - Full active log is copied to an Archive log – the 'A' file is registered in BSDS
 - After the OFFLOAD, the active log is again available for use (Status: Reusable)
- Dual Active logs are recommended
 - Increases availability
 - Increases reliability of recovery

For Subsystem
Recover

ABC's of DB2 Infrastructure: Bootstrap Dataset Active and Archive Logs

ACTIVE LOG COPY 1 DATA SETS					
START RBA/TIME	END RBA/TIME	DATE	LTIME	DATA SET INFORMATION	
00785B013000	00785F662FFF	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS01	
2013.169 16:38:57.4	2013.169 16:39:40.4			PASSWORD=(NULL) STATUS=REUSABLE	
00785F663000	007863CB2FFF	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS03	
2013.169 16:39:40.4	2013.169 20:14:44.9			PASSWORD=(NULL) STATUS=REUSABLE	
007863CB3000	007866C82FFF	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS04	
2013.169 20:14:44.9	2013.169 21:22:10.6			PASSWORD=(NULL) STATUS=REUSABLE	
007866C83000	0078678BEFFF	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS05	
2013.169 21:22:10.6	2013.192 14:47:46.9			PASSWORD=(NULL) STATUS=TRUNCATED, REUSABLE	
0078678BF000	00786BFC2FFF	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS02	
2013.192 14:47:46.9			PASSWORD=(NULL) STATUS=NOTREUSABLE	

Archive Log Copy 1 Listing:

ARCHIVE LOG COPY 1 DATA SETS					
START RBA/TIME	END RBA/TIME	DATE	LTIME	DATA SET INFORMATION	
00670F49A000	006713AE9FFF	2012.292	16:23	DSN=DEFSCAT.ARCHLOG1.D12292.T1623351.A0006237	
2012.292 21:23:23.1	2012.292 21:23:35.1			PASSWORD=(NULL) VOL=ARC032 UNIT=SYSALLDA CATALOGUED	
...					
007863CB3000	007866C82FFF	2013.169	16:22	DSN=DEFSCAT.ARCHLOG1.D13169.T1622106.A0007895	
2013.169 20:14:44.9	2013.169 21:22:10.6			PASSWORD=(NULL) VOL=ARC032 UNIT=SYSALLDA CATALOGUED	
007866C83000	0078678BEFFF	2013.192	9:48	DSN=DEFSCAT.ARCHLOG1.D13192.T0947470.A0007896	
2013.169 21:22:10.6	2013.192 14:47:46.9			PASSWORD=(NULL) VOL=ARC031 UNIT=SYSALLDA CATALOGUED	

Current Active

When DEFSCAT.LOGCOPY1.DS02 is offloaded, it will become archive A0007897

The top of the Archive Log Listing shows that this is the listing for the Archive Log Copy 1 Datasets. Just as you could specify dual actives, you can also specify dual archive logs.

What this slide is showing is that the Active Logs that have been marked REUSABLE (meaning they have already been offloaded), appear in the Archive Log Listing. Notice that the Start Rba/ End Rba ranges are the same.

Archives are most commonly created when an active is full, however they can also be created from an Archive Log Command, or if there is an error writing to the active, DB2 will initiate an offload.

ABC's of DB2 Infrastructure: Archive Logs

ZPARMS Related to Archive Logs	
MAXARCH=2000	Keep 2000 archives in the BSDS
TWOARCH=YES	Dual Archive Logs
ARCPFX1=DEFSCAT.ARCHLOG1	Prefix of dynamically allocated Archive 1
ARCPFX2=DEFSCAT.ARCHLOG2	Prefix of dynamically allocated Archive 2
TSTAMP=YES	Include a timestamp in the dataset name
UNIT=SYSALLDA	Unit for Archive 1
UNIT2=CART	Unit for Archive 2

The next time an OFFLOAD occurs, the following 4 datasets will be created:

ARCPFX1	TSTAMP	UNIT	
DEFSCAT.ARCHLOG1	D13189.T1440428	B0007897	– disk copy of BSDS
DEFSCAT.ARCHLOG1	D13189.T1440428	A0007897	– disk copy of active log
ARCPFX2		UNIT2	
DEFSCAT.ARCHLOG2	D13189.T1440428	B0007897	– tape copy of BSDS (file 1)
DEFSCAT.ARCHLOG2	D13189.T1440428	A0007897	– tape copy of active log (file 2)

In DB2 V10, the BSDS can support 10,000 archive logs.

ABC's of DB2 Infrastructure: Bootstrap Dataset

```

CONDITIONAL RESTART CONTROL RECORD
      14:50:01 JULY 11, 2013
**** ACTIVE CRCR RECORD ****
      NO CRCR RECORDS ARE ACTIVE
****
    
```

Conditional Restart: A DB2 restart that is directed by a user defined Conditional Restart Control Record

```

CHECKPOINT QUEUE
      14:50:01 JULY 11, 2013
TIME OF CHECKPOINT      14:47:47 JULY 11, 2013
BEGIN CHECKPOINT RBA      0078678C08EC
END CHECKPOINT RBA      0078678C365C
END CHECKPOINT STCK      CBA4DC2FB679
TIME OF CHECKPOINT      13:44:22 JULY 10, 2013
BEGIN CHECKPOINT RBA      00786788343A
END CHECKPOINT RBA      0078678860A6
END CHECKPOINT STCK      CBA38C25D715
    
```

Issue Display Log command to see current checkpoint frequency

A Conditional Restart is a DB2 restart (startup) that is directed by a user-defined Conditional Restart Control Record (CRCR). When we do a DR at a remote site, we will be inserting a CRCR into the BSDS.

A checkpoint is a system wide event. It's a point at which DB2 records status information on the DB2 log.

For example, it will record all open units of recovery at the time of the checkpoint.

This information is used by DB2 every time it starts up. Some of the things that trigger a checkpoint are:

- Every time DB2 switches active log datasets
- When DB2 is stopped
- Checkpoint frequency can be specified in the zparm by either Log Records per Checkpoint or Minutes per Checkpoint.

ABC's of DB2 Infrastructure Summary: DB2 objects used in a Recovery

- COPY TABLESPACE IDUGDB.SCHEDULE
- INSERT INTO IDUG.SCHEDULE (IDUGID, TITLE) VALUES ('F9','THE ABCs OF DISASTER RECOVERY');
- Report Recovery:

```
//REPORT EXEC PGM=DSNUTILB, PARM='DEFS'  
//STEPLIB DD DSN=SYS3.DEFS.DSNEXIT, DISP=SHR  
// DD DSN=CSGI.DB2V10M.DSNLOAD, DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
REPORT RECOVERY TABLESPACE IDUGDB.SCHEDULE
```

Summary: DB2 objects used in a Recovery Report Recovery

- SYSCOPY and SYSLGRNX entries:

```

DSNU582I *DEFS 192 12:53:10.61 DSNUPPCP - REPORT RECOVERY TABLESPACE IDUGDB.SCHEDULE SYSCOPY ROWS
AND SYSTEM LEVEL BACKUPS
TIMESTAMP = 2013-07-11-12.53.10.047635, IC TYPE = F SHR LVL = R, DSNUM = 0000, START LRSN =0078679CB191
DEV TYPE = 3390 , IC BACK = , STYPE = , FILE SEQ = 0000, PIT LRSN = 000000000000
LOW DSNUM = 0001, HIGH DSNUM = 0001, OLDEST VERSION = 0000, LOGICAL PART = 0000, LOGGED = Y, TTYPE =
JOBNAME = RMDRPT , AUTHID = RDATQG2 , COPYPAGESF = 3.0E+00
NPAGESF = 9.0E+01 , CPAGESF = 2.0E+00
DSNAME = RMD.IDUGDB.SCHEDULE.DEFS.D00.LP125309 , MEMBER NAME = , INSTANCE = 01, RELCREATED =

DSNU583I *DEFS 192 12:53:10.61 DSNUPPLR - SYSLGRNX ROWS FROM REPORT RECOVERY FOR TABLESPACE IDUGDB.SCHEDULE
UCDATE UCETIME START RBA STOP RBA START LRSN STOP LRSN PARTITION MEMBER ID
071113 12465734 0078678D25B2 0078678D3764 CBA5043C031F CBA5043C0CA7 0001 0000
071113 12465908 0078678DD491 0078678E206A CBA5043DAB27 CBA50510C382 0001 0000
071113 12531031 0078679CCD4C 0078679CD23F CBA5059FB45E CBA5059FD88F 0001 0000 *
```

- 1) Catalog table SYSCOPY records full image backup taken at 0078679CB191
- 2) Directory table SYSLGRNX records open log ranges.
 - Row with '*' indicates it will participate in the recovery.
- 3) BSDS will identify the log(s) to be read for Start Rba 0078679CCD4C thru Stop RBA 0078679CD23F.

If some of the log had been on an archive log, it would have listed the archive logs that would be used in the recovery.

Summary: DB2 objects used in a Recovery DSN1LOGP will dump Log Records

Report Recovery SYSLGRNX listing:

DSNU583I	*DEFS 192	12:53:10.61	DSNUPPLR	- SYSLGRNX	ROWS	FROM REPORT	RECOVERY FOR	TABLESPACE	IDUGDB.SCHEDULE		
UCDATE	UCTIME	START	RBA	STOP	RBA	START	LRSN	STOP	LRSN	PARTITION	MEMBER ID
071113	12465734	0078678D25B2	0078678D3764	CBA5043C031F	CBA5043C0CA7	0001	0000				
071113	12465908	0078678DD491	0078678E206A	CBA5043DAB27	CBA50510C382	0001	0000				
071113	12531031	0078679CCD4C	0078679CD23F	CBA5059FB45E	CBA5059FD88F	0001	0000				*

DSN1LOGP Syntax:

```
//DSN1LOGP EXEC PGM=DSN1LOGP,REGION=0M
//STEPLIB DD DISP=SHR,DSN=SYS3.DEFS.DSNEXIT
// DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD
//BSDS DD DISP=SHR,DSN=DEFSCAT.BSDS01
//SYSPRINT DD SYSOUT=*
//SYSSUMRY DD SYSOUT=*
//SYSIN DD *
STARTRBA(0078679CCD4C)
ENDRBA (0078679CD23F)
SUMMARY(NO)
```

RBA Range from
Report Recovery

DSN1LOGP will print the log records for the log range specified.

Summary: DB2 objects used in a Recovery DSN1LOGP Output - Log Records

INSERT INTO IDUG.SCHEDULE (IDUGID, TITLE)
VALUES ('F9','THE ABCs OF DISASTER RECOVERY');

```

0078679CD0D2 URID(0078679CD0D2)
                LRSN(CBA5059FB4E5) TYPE(UR CONTROL) SUBTYPE(BEGIN UR) 12:53:10 13.192
*LRH* 00900034 00200001 03800078 679CD0D2 00000000 00000726 00000000 0000CBA5 * * * K * V
      059FB4E5 0000
0000 001E0000 0000D000 00000000 00000700 0000D9D4 C4D9D7E3 40404040 4040D9C4 * * V
0020 C1E3D8C7 F240CBA5 059FB4E5 9A6BC4E2 D5E3C5D7 F240C2C1 E3C3C840 4040C2C1 *ATQG2 v v ,DSNTEP2 BATCH BA
0040 E3C3C840 40400000 00000000 0000001A 0001E4E2 C2D4C3D5 F0F1C4C5 C6E2D3E4 *TCH USBMCN01DEFSLU
0060 4040CBA5 059F9E93 0001
0078679CD162 TYPE(UNDO REDO ) URID(0078679CD0D2)
                LRSN(CBA5059FB4E5) DBID(010E) OBID(0002) PAGE(00000002) 12:53:10 13.192
                SUBTYPE(INSERT IN A DATA PAGE) CLR(NO) PROCNAME(DSNISGRT)
*LRH* 00730090 06000001 0E800078 679CD0D2 0078679C D0D20726 0078679C D0D2CBA5 * * * K K K V
      059FB4E6 0000
*LC** 88010E00 02000000 02000078 678E1F78 4C02
0000 003B5002 00030010 00003300 000200C6 F94000E3 C8C540C1 C2C3E240 D6C640C4 *h * <
0020 C9E2C1E2 E3C5D940 D9C5C3D6 E5C5D9E8 40404040 40404040 404040 * & F9 THE ABCS OF D
                *ISASTER RECOVERY

```

3 more log records written but not shown:

- 1)Type (UR CONTROL) SUBTYPE(BEGIN COMMIT1)
- 2)Type (UR CONTROL) SUBTYPE(PHASE 1 TO 2)
- 3)Type (UR CONTROL) SUBTYPE(END COMMIT2)

This slide reflects the unit of recovery that is written in the log to record the Insert into the table.

This is a Single Phase Commit. A single phase commit is a process that is running directly on DB2.

Fundamentals Pop Quiz



- What is the utility used to print the BSDS?
 - DSNJU004 – the Print Log Map Utility
- List 3 pieces of information stored in the BSDS.
 - Active Log List, Archive Log List, Conditional Restart Control Records, Checkpoint Queue, Archive Log Command History
- In the BSDS listing, how do you know which active log is the **current** active log?
 - Status: NotReusable and End Time: Blanks
- During Offloading, an 'A' and a 'B' file are created. What is the 'B' file?
 - A copy of the BSDS

The Balancing Act of Disaster Recovery

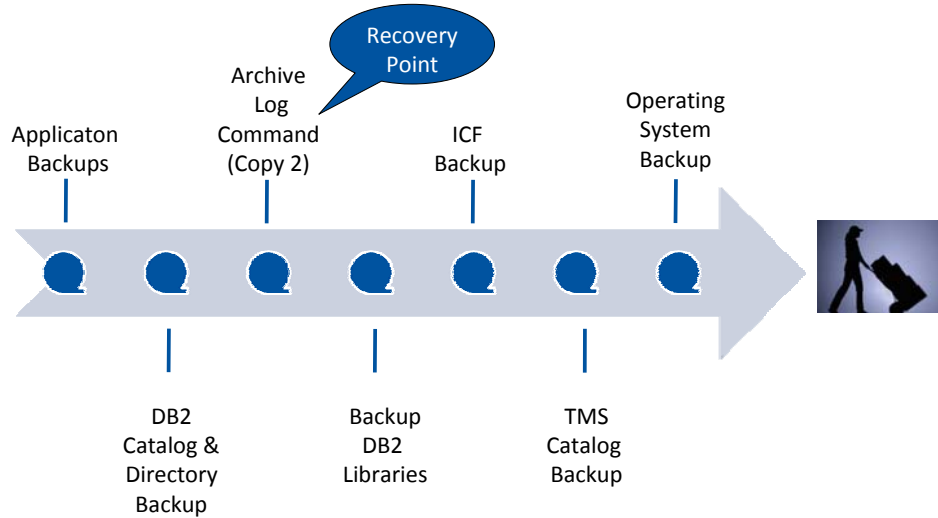
- Options from weekly dumps to offsite logging
 - Dumps - Simple, cheap, maximum data loss
 - Weekly dumps means several days data loss
 - Remote Mirror - Complex, expensive, no data loss
 - Disk, Network, Software, Facilities, Operations
 - Compromise - Periodic vaulting of Image Copies & Logs
 - Daily or hourly log shipment will minimize data loss
 - Recovery is to the most recent archive at the remote site



Disaster Recovery support is always a balancing act between cost and complexity vs. data loss and recovery time.

The compromise at many companies is a technique to transfer all copies of application data and DB2 system images along with DB2 archive logs to a remote site. In the event of a disaster, the DB2 subsystem recovery is made up to the most recent archive log at the remote site. This is a technique that is documented in the DB2 Administration Guide and it is this process that we will be covering today.

Disaster Recovery Preparation at the Local Site



Establish a recovery point by archiving a log. We are assuming that we are performing dual archiving and we are sending the Archive Copy 2 to the remote site for the DR drill.

Backup DB2 required libraries includes:

zparms

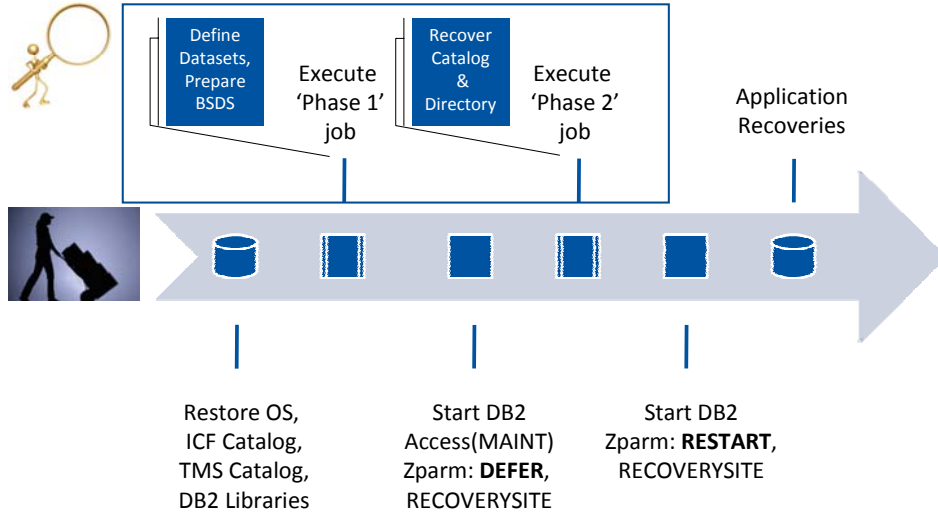
DB2 Load and EXIT libs

Backup Integrated Catalog Facility (ICF)

Create a backup of the tape management system catalog

Transport all copies to the remote site

Disaster Site Execution



We will focus on the steps that make up the 'Phase 1' and 'Phase 2' jobs.

Overview of Phase 1 tasks (performed with DB2 down)

- Assumption: At the DR site we have image copies of the DB2 Catalog & Directory and the Archive Log Copy 2 Datasets.
- Last Archive Log Copy 2 is:
 - DEFSCAT.ARCHLOG2.D13205.T1321128.A0007900 – Archive Log
 - DEFSCAT.ARCHLOG2.D13205.T1321128.B0007900 – Archive copy of BSDS
- Phase 1 tasks - Define VSAM datasets for DB2 Infrastructure objects and Prepare the BSDS
 1. IDCAMS DELETE NOSCRATCH/DEFINE VSAM Datasets
 2. Restore BSDS from most recent archive log
 3. Add most recent archive to the BSDS
 4. Prepare active logs in the BSDS
 5. Create Conditional Restart Control Record in the BSDS
 6. Print BSDS and verify

} Prepare the
Bootstrap
Dataset

The main objective of the Phase 1 tasks is to define the VSAM datasets for our DB2 Infrastructure objects (BSDS, Active Logs, Catalog and Directory), and prepare the BSDS for startup. Reminder: the BSDS is required for DB2 to start.

For Datasharing, all of these steps must be performed for each member in the datasharing group.

Phase 1 Tasks

1) IDCAMS DELETE/DEFINE VSAM Objects

- **IDCAMS DELETE NOSCRATCH and DEFINE for:**
 - Bootstrap Dataset (both copies): BSDS01 and BSDS02
 - Active Log Copy 1 and Active Log Copy 2 datasets
 - Catalog (DSNDB06) and Directory (DSNDB01) tablespaces & indexes
 - Any user defined catalog indexes
 - Include **Define** for **Stogroup** user indexes on SYSCOPY
(Aug 2013 Doc Change)
- Why DELETE at DR site?
 - ICF catalog reflects datasets that may not exist on disk
- Template for DEFINES?
 - Use installation job DSNTIJIN
- For Datasharing, each member has BSDS and Active Logs

Stogroup Ix on
SYSCOPY
Not
recommended

In DB2 V10, the Catalog & Directory datasets must be SMS-Managed.

IBM does not recommend that you create Stogroup user defined indexes on SYSCOPY, however, if you do, they made an August 2013 documentation change to the Utility Guide for that scenario. Part of the new requirement is that you should DEFINE the stogroup index. However they do have a warning in bold that indexes on SYSCOPY should not be stogroup managed.

Phase 1 Tasks

2) Restore BSDS from the most recent archive log


A. IDCAMS REPRO to restore BSDS01 from most recent archive

- Reminder – Last archive at the DR site:

DEFSCAT.ARCHLOG2.D13205.T1321128.A0007900 – Archive Log

DEFSCAT.ARCHLOG2.D13205.T1321128.B0007900 – Archive copy of BSDS

```
//ARM00004 EXEC PGM=IDCAMS,REGION=4M,COND=(4,LT)
//SYSUT1 DD DISP=OLD,DSN=DEFSCAT.ARCHLOG2.D13205.T1321128.B0007900,
//      DCB=(BUFNO=10)
//SYSUT2 DD DISP=MOD,DSN=DEFSCAT.BSDS01,
//      AMP='BUFND=100'
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
REPRO -
      INFILE (SYSUT1) -
      OUTFILE(SYSUT2) -
      REPLACE
```



Phase 1 Tasks

2) Restore BSDS from the most recent archive log

B. IDCAMS REPRO to copy BSDS01 to BSDS02

```
//ARM00005 EXEC PGM=IDCAMS,REGION=4M,COND=(4,LT)
//SYSUT1 DD DISP=MOD,DSN=DEFSCAT.BSDS01,
//      AMP='BUFND=100'
//SYSUT2 DD DISP=MOD,DSN=DEFSCAT.BSDS02,
//      AMP='BUFND=100'
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
REPRO -
      INFILE (SYSUT1) -
      OUTFILE(SYSUT2) -
      REPLACE
```

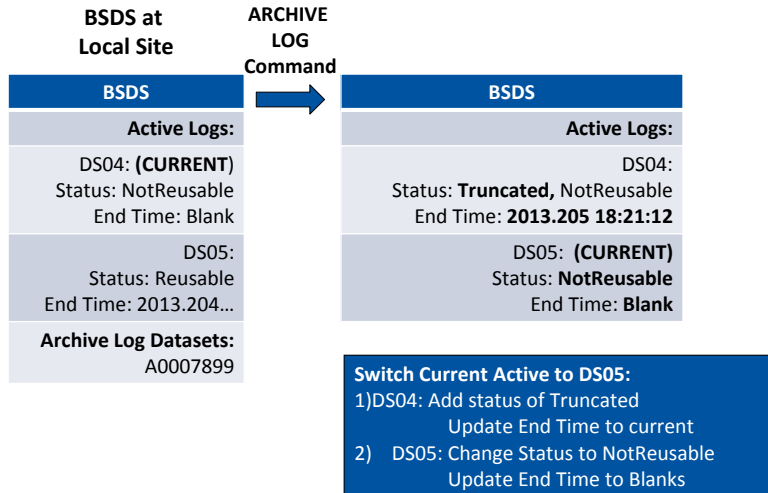
The two copies of the BSDS must be kept in sync.

If you have dual bootstraps (which you should always have), we have to keep the 2 copies of the BSDS in sync.

Restore the recovered bootstrap to the second copy, BSDS02.

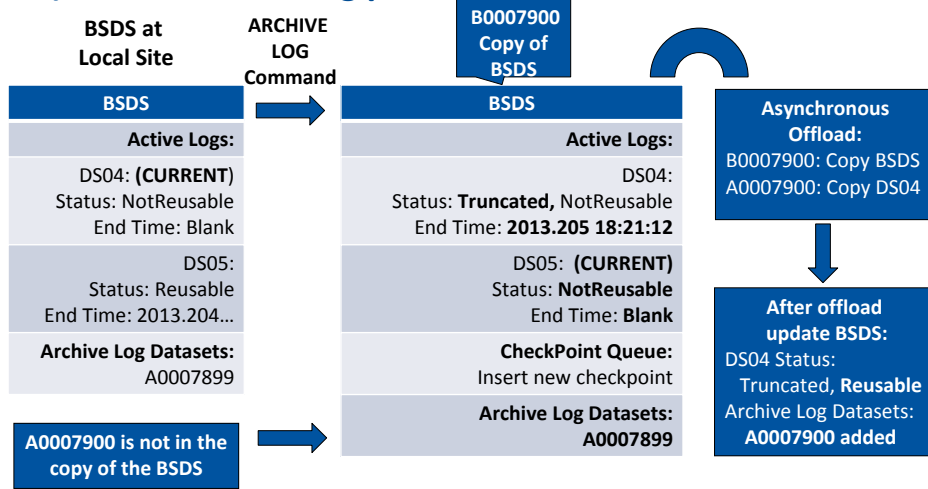
Phase 1 Tasks

3) Review offloading process at Local site for A0007900



Phase 1 Tasks

3) Review offloading process at Local site for A0007900



Phase 1 Tasks

3) Add most recent archive (A0007900) to the BSDS

A. Use DSNJU004 to print the BSDS – gather information

ACTIVE LOG COPY 1 DATA SETS					
START RBA/TIME	END RBA/TIME	DATE	LTIME	DATA SET INFORMATION	
0078678BF000	0078679D4FFF	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS02	
2013.192 14:47:46.9	2013.192 18:01:32.2	PASSWORD=(NULL)		STATUS=TRUNCATED, REUSABLE	
0078679D5000	007868CF8FFF	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS01	
2013.192 18:01:32.2	2013.203 22:59:35.4	PASSWORD=(NULL)		STATUS=TRUNCATED, REUSABLE	
007868CF9000	00786C388FFF	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS03	
2013.203 22:59:35.4	2013.204 20:21:58.4	PASSWORD=(NULL)		STATUS=TRUNCATED, REUSABLE	
00786C389000	00786F0EFFFF	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS04	
2013.204 20:21:58.4	2013.205 18:21:12.6	PASSWORD=(NULL)		STATUS=TRUNCATED, NOTREUSABLE	
00786F0EF000	0078720BEFFF	2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS05	
2013.205 18:21:12.6	PASSWORD=(NULL)		STATUS=NOTREUSABLE	

ARCHIVE LOG COPY 2 DATA SETS					
START RBA/TIME	END RBA/TIME	DATE	LTIME	DATA SET INFORMATION	
006729B2E000	00672CA49FFF	2012.292	16:25	DSN=DEFSCAT.ARCHLOG2.D12292.T1624569.A0006244	
2012.292 21:24:49.9	2012.292 21:24:56.8	PASSWORD=(NULL)		VOL=711661 UNIT=CARTVTS CATALOGUED	
007868CF9000	00786C388FFF	2013.204	15:22	DSN=DEFSCAT.ARCHLOG2.D13204.T1521585.A0007899	
2013.203 22:59:35.4	2013.204 20:21:58.4	PASSWORD=(NULL)		VOL=141253 UNIT=CARTVTS CATALOGUED	

Archive Log: Get the Dataset name of the oldest archive log (it will be deleted)

Active Log: Get Start RBA/End RBA; Start Time/End Time to register A0007900

Gather the following information from the BSDS:

- Delete the oldest archive in the BSDS – the BSDS has a limited number of archive slots. Since we are going to add one to it we want to make sure we don't fill all the slots so we will first delete the oldest archive in the list. Note the Dataset name of the oldest archive in the list.
- To add sequence A0007900 to the archive list, we get the information we will use to register it from the active log. At the time the offloading process made a copy of the BSDS, DB2 had changed the status of the active log to TRUNCATED, NOTREUSABLE (NOTREUSABLE indicates it has not been offloaded yet). Therefore, DEFSCAT.LOGCOPY1.DS04 is the active log that got copied to archive log dataset A0007900. Note the Start RBA/Time and End RBA/Time. This information will be used to register the archive log into the BSDS.

Phase 1 Tasks

3) Add most recent archive to the BSDS

B. Prior to adding A0007900 to the BSDS, delete the oldest archive in the list to ensure there is room for the insert.

```
//ARM00006 EXEC PGM=DSNJU003,REGION=0M,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=SYS3.DEFS.DSNEXIT
//          DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD
//SYSUT1 DD DISP=MOD,DSN=DEFSCAT.BSDS01,
//        AMP='BUFND=32,BUFNI=16'
//SYSUT2 DD DISP=MOD,DSN=DEFSCAT.BSDS02,
//        AMP='BUFND=32,BUFNI=16'
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
DELETE DSNNAME=DEFSCAT.ARCHLOG2.D12292.T1624569.A0006244
```

- DSNJU003 – Change Log Inventory Utility used to update BSDS
- Specify BSDS01 and BSDS02 in SYSUT1 & SYSUT2 so they stay in sync

DSNJU003 is the utility that is used to update the BSDS. It is called the Change Log Inventory program.

Phase 1 Tasks

3) Add most recent archive to the BSDS

C. Add Archive A0007900 to the BSDS

```
//ARM00007 EXEC PGM=DSNJU003,REGION=0M,COND=(4,LT)
//STEP1LIB DD DISP=SHR,DSN=SYS3.DEFS.DSNEXIT
//          DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD
//SYSUT1 DD DISP=MOD,DSN=DEFSCAT.BSDS01,
//        AMP='BUFND=32,BUFNI=16'
//SYSUT2 DD DISP=MOD,DSN=DEFSCAT.BSDS02,
//        AMP='BUFND=32,BUFNI=16'
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
NEWLOG DSNNAME=DEFSCAT.ARCHLOG2.D13205.T1321128.A0007900,
        STARTRBA=00786C389000,ENDRBA=00786F0EEFFF,
        STARTIME=20132042021584,ENDTIME=20132051821126,
        COPY2VOL=139190,
        CATALOG=YES
```

- NEWLOG inserts A0007900 using information gathered from the active log
- COPY2VOL adds entry to Archive Log Copy 2 Dataset List

DSNJU003 is the utility that is used to update the BSDS. It is called the Change Log Inventory program.

Phase 1 Tasks

4) Prepare Active Logs in the BSDS

Problem: the BSDS needs to be updated to reflect empty logs:

ACTIVE LOG COPY 1 DATA SETS						
START	RBA/TIME	END	RBA/TIME	DATE	LTIME	DATA SET INFORMATION
00786788F000		0078679D4FFF		2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS02
2013.192	14:47:46.9	2013.192	18:01:32.2	PASSWORD=(NULL)		STATUS=TRUNCATED, REUSABLE
0078679D5000		00786BCF8FFF		2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS01
2013.192	18:01:32.2	2013.203	22:59:35.4	PASSWORD=(NULL)		STATUS=TRUNCATED, REUSABLE
00786BCF9000		00786C388FFF		2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS03
2013.203	22:59:35.4	2013.204	20:21:58.4	PASSWORD=(NULL)		STATUS=TRUNCATED, REUSABLE
00786C389000		00786F0EFFFF		2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS04
2013.204	20:21:58.4	2013.205	18:21:12.6	PASSWORD=(NULL)		STATUS=TRUNCATED, NOTREUSABLE
00786F0EF000		0078720BEFFF		2013.144	11:28	DSN=DEFSCAT.LOGCOPY1.DS05
2013.205	18:21:12.6	PASSWORD=(NULL)		STATUS=NOTREUSABLE

- Step 1 of the 'Phase 1' task was IDCAMS Delete/Define of the active datasets.
- For DB2 to recognize the actives as empty:
 - Delete the Active Logs from the BSDS
 - Add them back to the BSDS with the NEWLOG command

The active logs are empty at the DR site. The BSDS needs to be updated to reflect the empty logs.

Phase 1 Tasks

4) Prepare Active Logs in the BSDS

A. Delete Active Logs (copy 1 & 2) from the BSDS

```
//ARM00008 EXEC PGM=DSNJU003,REGION=0M,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=SYS3.DEFS.DSNEXIT
//          DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD
//SYSUT1 DD DISP=MOD,DSN=DEFSCAT.BSDS01,
//          AMP='BUFND=32,BUFNI=16'
//SYSUT2 DD DISP=MOD,DSN=DEFSCAT.BSDS02,
//          AMP='BUFND=32,BUFNI=16'
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE DSNAME=DEFSCAT.LOGCOPY1.DS01
DELETE DSNAME=DEFSCAT.LOGCOPY1.DS02
DELETE DSNAME=DEFSCAT.LOGCOPY1.DS03
DELETE DSNAME=DEFSCAT.LOGCOPY1.DS04
DELETE DSNAME=DEFSCAT.LOGCOPY1.DS05

DELETE DSNAME=DEFSCAT.LOGCOPY2.DS01
DELETE DSNAME=DEFSCAT.LOGCOPY2.DS02
DELETE DSNAME=DEFSCAT.LOGCOPY2.DS03
DELETE DSNAME=DEFSCAT.LOGCOPY2.DS04
DELETE DSNAME=DEFSCAT.LOGCOPY2.DS05
```

Phase 1 Tasks

4) Prepare Active Logs in the BSDS

B. Add Active Logs (copy 1 & 2) to the BSDS

```
//ARM00009 EXEC PGM=DSNJU003,REGION=0M,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=SYS3.DEFS.DSNEXIT
//          DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD
//SYSUT1 DD DISP=MOD,DSN=DEFSCAT.BSDS01,
//        AMP='BUFND=32,BUFNI=16'
//SYSUT2 DD DISP=MOD,DSN=DEFSCAT.BSDS02,
//        AMP='BUFND=32,BUFNI=16'
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
NEWLOG DSNAME=DEFSCAT.LOGCOPY1.DS01,COPY1
NEWLOG DSNAME=DEFSCAT.LOGCOPY1.DS02,COPY1
NEWLOG DSNAME=DEFSCAT.LOGCOPY1.DS03,COPY1
NEWLOG DSNAME=DEFSCAT.LOGCOPY1.DS04,COPY1
NEWLOG DSNAME=DEFSCAT.LOGCOPY1.DS05,COPY1

NEWLOG DSNAME=DEFSCAT.LOGCOPY2.DS01,COPY2
NEWLOG DSNAME=DEFSCAT.LOGCOPY2.DS02,COPY2
NEWLOG DSNAME=DEFSCAT.LOGCOPY2.DS03,COPY2
NEWLOG DSNAME=DEFSCAT.LOGCOPY2.DS04,COPY2
NEWLOG DSNAME=DEFSCAT.LOGCOPY2.DS05,COPY2
```

COPY1:
Add to Active
Log Copy 1
Dataset
List

The Active Logs are added with the NEWLOG syntax without any StartRba or EndRba parms. This is what indicates to DB2 that the actives are empty.

Phase 1 Tasks

5) Create Conditional Restart Control Record in BSDS

- At restart time, DB2 attempts to locate the last log RBA written before termination and continue from there.
- By inserting a Conditional Restart Control Record (CRCR) into the BSDS, we override DB2's normal startup process.
- The CRCR will tell DB2 what to use for the last log RBA written
 - CRESTART CREATE,ENDRBA=00786F0EF000
- DB2 will bypass any processing related to RBA's after the CRCR.
Example: later checkpoints in the Checkpoint Queue
- CRESTART ENDRBA/ENDLRN:
 - Non-datasharing: ENDRBA of last archive log + 1
 - Datasharing: Lowest ENDLRSN of last archive log in the datasharing group -1

Must be on an
even CI
boundary

Phase 1 Tasks

5) Create Conditional Restart Control Record in BSDS

Previous NEWLOG to add A0007900 to the BSDS:

```
NEWLOG DSN=DEFSCAT.ARCHLOG2.D13205.T1321128.A0007900,  
STARTRBA=00786C389000, ENDRBA=00786F0EEFFF,  
STARTIME=20132042021584, ENDTIME=20132051821126,
```

Create Conditional Restart Control Record:

```
//ARM00010 EXEC PGM=DSNJU003,REGION=0M,COND=(4,LT)  
//STEPLIB DD DISP=SHR,DSN=SYS3.DEFS.DSNEXIT  
// DD DISP=SHR,DSN=CSGI.DB2V10M.DSNLOAD  
//SYSUT1 DD DISP=MOD,DSN=DEFSCAT.BSDS01,  
// AMP='BUFND=32,BUFNI=16'  
//SYSUT2 DD DISP=MOD,DSN=DEFSCAT.BSDS02,  
// AMP='BUFND=32,BUFNI=16'  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
CRESTART CREATE, ENDRBA=00786F0EF000
```

ENDRBA: 00786F0EEFFF
+ 1
← 00786F0EF000

The Conditional Restart Control Record is added to every member of the datasharing group and each will have the same ENDLRSN.

Phase 1 Complete 6) Print BSDS and review

Active logs are all empty with status NEW, REUSABLE

ACTIVE LOG COPY 1 DATA SETS					
START RBA/TIME	END RBA/TIME	DATE	LTIME	DATA SET	INFORMATION
				EMPTY DATA SET	
0000.000	00:00:00.0	0000.000	00:00:00.0	2013.209	16:29 DSN=DEFSCAT.LOGCOPY1.DS01 PASSWORD=(NULL) STATUS=NEW, REUSABLE
				EMPTY DATA SET	
0000.000	00:00:00.0	0000.000	00:00:00.0	2013.209	16:29 DSN=DEFSCAT.LOGCOPY1.DS02 PASSWORD=(NULL) STATUS=NEW, REUSABLE
				EMPTY DATA SET	
0000.000	00:00:00.0	0000.000	00:00:00.0	2013.209	16:29 DSN=DEFSCAT.LOGCOPY1.DS03 PASSWORD=(NULL) STATUS=NEW, REUSABLE
				EMPTY DATA SET	
0000.000	00:00:00.0	0000.000	00:00:00.0	2013.209	16:29 DSN=DEFSCAT.LOGCOPY1.DS04 PASSWORD=(NULL) STATUS=NEW, REUSABLE
				EMPTY DATA SET	
0000.000	00:00:00.0	0000.000	00:00:00.0	2013.209	16:29 DSN=DEFSCAT.LOGCOPY1.DS05 PASSWORD=(NULL) STATUS=NEW, REUSABLE

Last archive, A0007900 has been added to the Archive Log Copy 2 Dataset List:

ARCHIVE LOG COPY 2 DATA SETS					
START RBA/TIME	END RBA/TIME	DATE	LTIME	DATA SET	INFORMATION
00786C389000	00786F0EEFFF	2013.209	16:29	DSN=DEFSCAT.ARCHLOG2.D13205.T1321128.A0007900	
2013.204	20:21:58.4	2013.205	18:21:12.6		PASSWORD=(NULL) VOL=139190 UNIT=3490 CATALOGUED

Phase 1 Complete 6) Print BSDS and review

Conditional Restart Control Record is Active:

**** ACTIVE CRCR RECORD ****

```

CRCR IDENTIFIER  1D14
USE COUNT       0
RECORD STATUS
  CRCR ACTIVE
  CRCR NOT USED
PROCESSING STATUS
  FORWARD = YES
  BACKOUT = YES
STARTRBA                NOT SPECIFIED
ENDRBA                   00786F0EF000
ENDRSN                   NOT SPECIFIED
ENDTIME                  NOT SPECIFIED
EARLIEST REQUESTED RBA  000000000000
FIRST LOG RECORD RBA    000000000000
ORIGINAL CHECKPOINT RBA 000000000000
NEW CHECKPOINT RBA (CHKPTRBA) 00786C739090
END CHECKPOINT RBA      00786C73C708
  
```

CRESTART CREATE,ENDRBA=00786F0EF000



CHECKPOINT QUEUE

```

21:29:47 JULY 28, 2013
TIME OF CHECKPOINT  18:21:12 JULY 24, 2013
BEGIN CHECKPOINT RBA 00786F0F0990
END CHECKPOINT RBA   00786F0F5458
END CHECKPOINT STCK  CBB564234ED4
TIME OF CHECKPOINT  18:06:04 JULY 24, 2013
BEGIN CHECKPOINT RBA 00786C739090
END CHECKPOINT RBA   00786C73C708
  
```

DB2 selected the correct Checkpoint
and entered it in the CRCR.

It could not select the most recent Checkpoint:
End Checkpoint RBA > ENDRBA
00786F0F5458 > 00786F0EF000

The checkpoint that DB2 selected and inserted into the Conditional Restart Control Record was the most recent checkpoint where the End Checkpoint RBA was less than the ENDRBA specified in the CRESTART ENDRBA.

Optional Phase 1 Tasks to Improve Performance

- Restore Archive Logs to Disk
 - Restoring some of the archive logs to disk will increase performance in later recover jobs
- Preformat Active Logs
 - The first time an active log is written to, DB2 must first preformat a VSAM control area
 - Our initial step was to Delete/Define the active logs
 - DSNJLOGF utility can be used to preformat the actives

Reassemble zparms for DR Site

- Reminder – At this point, the BSDS is ready for processing, however the catalog and directory have only been DEFINED.
- Reassemble zparms:

```
DSN6SPRM DEFER,
          ALL,
          SITETYP=RECOVERYSITE,
          ARC2FRST=YES

Datasharing:
DEL_CFSTRUCTS_ON_RESTART=YES
```

- DEFER ALL – Defer restart of All objects on DB2 startup
- ARC2FRST=YES – Use the second copy of the archive log (ARCHLOG2) first for restart and recovery.

DB2 needs to be brought up with ACCESS(MAINT) and some altered zparms so that we can recover the catalog and directory.

For Datasharing, the DEL_CFSTRUCTS_ON_RESTART=YES will tell DB2 that when it starts it should delete the SCA, IRLM Lock Structure, and any allocated group buffer pools from the coupling facility.

To clear the Coupling Facility manually:

1) Enter the following MVS command to display the structures for this data sharing group:

```
D XCF,STRUCTURE,STRNAME=grpname*
```

2) For group buffer pools and the lock structure, enter the following command for each ALLOCATED structure:

```
SETXCF FORCE,CONNECTION,STRNAME=strname,CONNNAME=ALL
```

3) Delete all DB2 ALLOCATED coupling facility structures by issuing the following command for each structure:

```
SETXCF FORCE,STRUCTURE,STRNAME=strname
```

START DB2 ACCESS=MAINT

- /*ssid STA DB2 ACCESS=MAINT

```
DSNY001I *DEFS SUBSYSTEM STARTING
DSNJ127I *DEFS SYSTEM TIMESTAMP FOR BSDS= 13.205 13:13:09.15
0007 DSNJ245I *DEFS CONDITIONAL RESTART RECORD INDICATES TRUNCATION AT RBA 00786F0EF000. REPLY Y TO CONTINUE,
N TO CANCEL
```

Reply Y to message: /R 0007,Y

```
DSNR003I *DEFS RESTART. PRIOR CHECKPOINT RBA=00786C739090
DSNR004I *DEFS RESTART. UR STATUS COUNTS=133
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
DSNI029I *DEFS DSNIFLAI THE FAST LOG APPLY PROCESS 154
IS ACTIVE AND THE AVAILABLE DEFSDBM1 STORAGE IS 1FE00000 BYTES
DSNI001I *DEFS RESTART HAS BEEN DEFERRED 155
REASON 00C90095
TYPE 00000201
NAME DSNDB06 .DSNUCX01
DSNB250E *DEFS DSNIMPDA PAGE RANGE WAS ADDED TO 156
THE LOGICAL PAGE LIST
DATABASE NAME=DSNDB06
SPACE NAME=DSNUCX01
INSTANCE=1
DATA SET NUMBER=1
PAGE RANGE X'00000000' TO X'FFFFFFF'
START LRSN=X'00786C78C1ED'
END LRSN=X'00786F0EEFFF'
```

DB2 scans from the begin Checkpoint Rba in the CRCR to the CRCR Endrba for the following units of recovery:

InFlight – DB2 writes log to backout

In Abort – DB2 writes log to backout

In Commit – will be completed with Recover

InDoubt – Action required by User

DB2 will automatically backout Inflight and In Abort units of recovery and it will complete In Commit units of recovery when the objects are recovered.

The In Doubt transactions must be handled by the user.

The following REASON codes may appear in the Master Address Space, they are normal and expected:

00C90081

00C90094

00C90095

00C900A9

Overview of Phase 2 tasks (performed with DB2 up in MAINT mode)

- Phase 2 tasks:
 1. Resolve InDoubt Units of Recovery
 2. Recover SYSUTILX then term any utilities against catalog & directory
 3. Recover the rest of the Catalog and Directory
 4. Define the work file database

Phase 2 Tasks

1) Resolving InDoubt Units of Recovery

- InDoubt: Comes into play with 2 Phase Commit. Ex: DB2 fails after it has finished the phase 1 commit but before the phase 2 commit.
- InDoubt units of recovery must be COMMITed or ABORTed prior to recovering the related tablespaces or the recovery will fail.
- The information to Commit/Abort must come from coordinating system.
- If DSNR004I message indicates indoubts:
 - -DISPLAY THREAD(*) TYPE(INDOUBT) DETAIL
 - The display will show a transaction id that will be used in the following:
 - -RECOVER INDOUBT ACTION(COMMIT) LUWID(nnn)/NID(nnn)
 - -RECOVER INDOUBT ACTION(ABORT) LUWID(nnn)/NID(nnn)

```
DSNR004I *DEFS RESTART...UR STATUS COUNTS 153  
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
```

The information that is needed to either ABORT or COMMIT InDoubt threads must come from the coordinating system.

Phase 2 Tasks

2) Recover SYSUTILX / Term Utilities

- Recover DSNDB01.SYSUTILX in a separate job step
- REBUILD INDEX(ALL) on DSNDB01.SYSUTILX
- SYSUTILX contains information about active and outstanding utilities
 - Term any utilities that are in progress on catalog or directory tablespaces
- -DISPLAY UTILITY(*)
- If the Display shows any utilities, use the following to determine what tablespaces are associated with the outstanding utilities:

```
//DIAGNOSE EXEC PGM=DSNUTILB, PARM='DEFS', REGION=4M
//STEPLIB DD DSN=SYS3.DEFS.DSNEXIT, DISP=SHR
// DD DSN=CSGI.DB2V10M.DSNLOAD, DISP=SHR
//SYSPRINT DD SYSOUT=*
//UTPRINT DD SYSOUT=*
//SYSUTL DD UNIT=WORK, SPACE=(CYL,(2,2))
//SYSIN DD *
DIAGNOSE DISPLAY SYSUTIL
```

- Use TERM UTILITY to term utilities against the catalog or directory.

Use the TERM UTILITY command on a utility that is operating on any object except DSNDB01.SYSUTILX

Phase 2 Tasks

3) Recover the rest of the Catalog & Directory

- Recover the Catalog & Directory objects in the following order:
 - RECOVER DSND01.DBD01
 - REBUILD INDEX(ALL) on DSND01.DBD01
 - RECOVER DSND01.SYSDBDXA
 - REBUILD INDEX(ALL) on DSND01.SYSDBDXA
 - RECOVER DSND06.SYSCOPY
 - If any user defined indexes exist on SYSCOPY:
 - REBUILD INDEX(ALL) on DSND06.SYSCOPY **REUSE**
 - If no user defined indexes exist on SYSCOPY:
 - REBUILD INDEX(ALL) on DSND06.SYSCOPY

Safety tip:
Use an individual job
step for each bullet

Stogroup Ix on
SYSCOPY
Not
recommended

Image copies for the following are registered in the Logs (**not SYSCOPY**):
DSND01.SYSUTILX, DSND01.DBD01, DSND01.SYSDBDXA, DSND06.SYSCOPY

Phase 2 Tasks

3) Recover the rest of the Catalog & Directory

- Recover the Catalog & Directory objects in the following order:
 - RECOVER DSND01.SYSLGRNX
 - REBUILD INDEX(ALL) on DSND01.SYSLGRNX
 - RECOVER DSND06.SYSTSSTG
 - RECOVER DSND06.SYSTSVOL
 - If no STOGROUP user defined indexes on SYSTSSTG:
 - REBUILD INDEX(ALL) on DSND06.SYSTSSTG
 - If any STOGROUP user defined indexes on SYSTSSTG:
 - Rebuild each individual IBM index on SYSTSSTG by name
 - If no STOGROUP user defined indexes on SYSTSVOL:
 - REBUILD INDEX(ALL) on DSND06.SYSTSVOL
 - If any STOGROUP user defined indexes on SYSTSVOL:
 - Rebuild each individual IBM index on SYSTSVOL by name

Phase 2 Tasks

3) Recover the rest of the Catalog & Directory

- The remaining Catalog and Directory tablespaces can be recovered within one RECOVER utility statement with the PARALLEL option.
 - V10 NFM remainder: 4 Directory and 92 Catalog tablespaces
- REBUILD INDEX(ALL) on DSNDB06.SYSRTSTS
- REBUILD INDEX(ALL) for all of the remaining catalog and directory tablespaces recovered in the first bullet
- Rebuild all of the individual stogroup user defined indexes that have not yet been recovered on SYSTSSTG or SYSTSVOL.

Phase 2 Tasks

4) Define the Work file Database

- Non-Datasharing: Work File Database is DSNDB07
- If the Work File Database is User Defined (VCAT):
 - -STOP DB(DSNDB07)
 - IDCAMS DELETE NOSCRATCH for all datasets
 - IDCAMS DEFINE CLUSTER
(NAME(DEFSCAT.DSNDBC.DSNDB07.DSN4K01.I0001.A001) ...
 - -START DB(DSNDB07)
- If the Work File Database is Stogroup managed:
 - -STOP DB(DSNDB07)
 - DROP TABLESPACE DSNDB07.DSN4K01
 - CREATE TABLESPACE DSN4K01 IN DSNDB07 USING STOGROUP...
 - -START DB(DSNDB07)

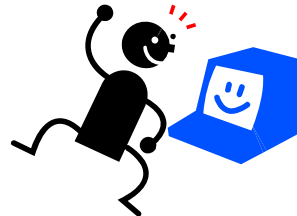
Datasharing: Each member has its own work file database:
Ex: DSNDB07.DSNDBC.DII1.DSN4K01.I0001.A001

The work file database does not contain permanent data. It is used for temporary space for certain SQL operations, such as join and ORDER BY.

It is also used as storage for global temporary tables.

DB2 Infrastructure Recovery Complete!

- Stop DB2
- Reassemble zparms
 - Change DEFER to RESTART
- Start DB2
- Backup the Catalog and Directory



WHEW!! PARTY!!!

- You are ready to recover your applications!!



The banner features a dark blue header with the IDUG logo (25 years) and the text "IDUG DB2 Tech Conference Barcelona, Spain - October 2013". Below the header, the name "Terri Grissom" is displayed in large blue font, followed by "BMC Software" and her email address "Terri_Grissom@bmc.com". The session information "Session F09 The ABC's of Disaster Recovery" is listed below. At the bottom, there is a row of grey silhouettes of people and the IDUG logo with the text "INTERNATIONAL DB2 USERS GROUP" and "Founded 1988".

25 IDUG
Leading the DB2 User
Community since 1988

IDUG DB2 Tech Conference
Barcelona, Spain - October 2013

Terri Grissom
BMC Software
Terri_Grissom@bmc.com

Session F09
The ABC's of Disaster Recovery

INTERNATIONAL
DB2 USERS GROUP
Founded 1988

Terri Grissom has been with BMC Software for 12 years. She is currently a Senior Quality Assurance Engineer responsible for testing the Recovery Management Solution products. One of her responsibilities is testing the disaster recovery feature of Recovery Manager.

Prior to BMC, Terri was a consultant. In her 14 years of consulting work, Terri developed mainframe application software; and installed, implemented and customized vendor software packages running under DB2.

Terri holds a Masters of Science degree in Computer Science from the University of Missouri – Rolla. She is also an ASQ Certified Software Quality Engineer.