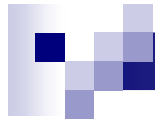




# **IBM DB2 LUW Advanced Recovery Feature**

**Bill Minor, IBM**  
***bminor@ca.ibm.com***



# Advanced Recovery Feature for IBM DB2 LUW

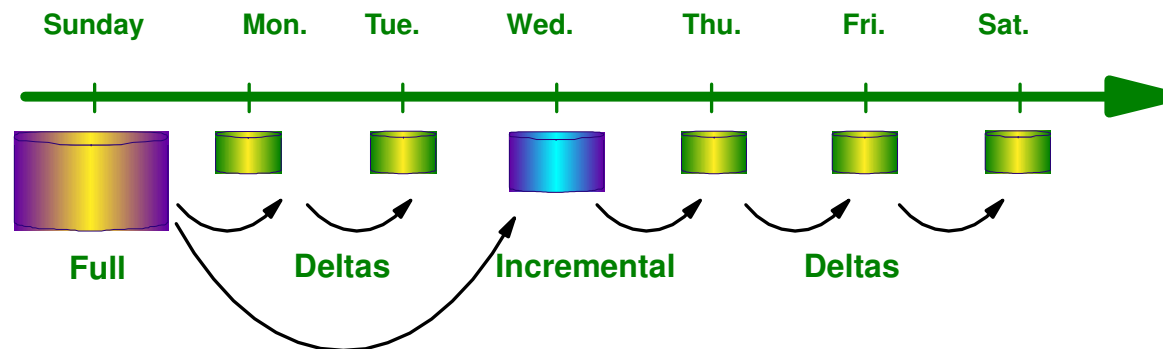
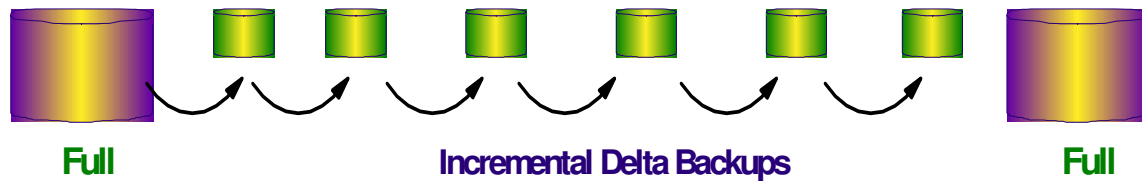
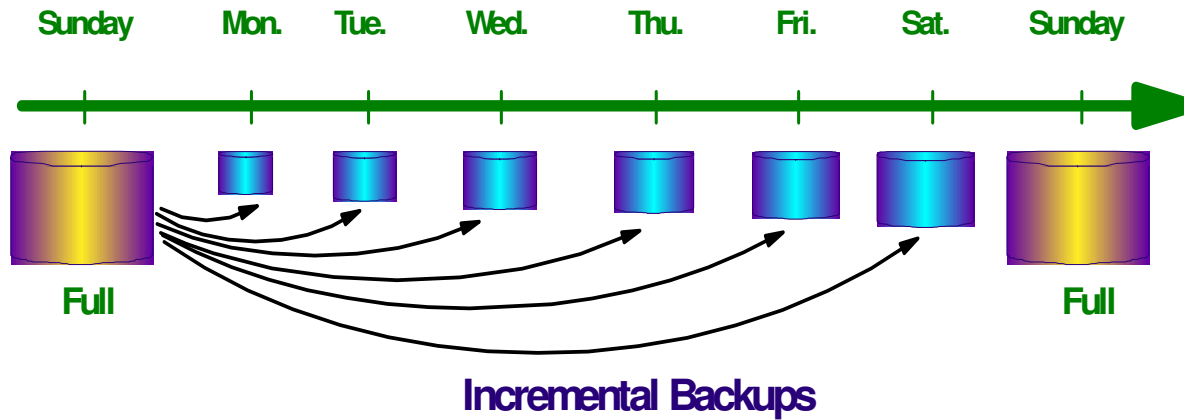
- *DB2 Merge Backup for LUW*
- *Infosphere Optim High Performance Unload*
- *DB2 Recovery Expert for LUW*

*Complement, Extend, Simplify, ... improve data availability*



# *Merge Backup*

# DB2 Incremental Backups – Simple Overview





# IBM DB2 Merge Backup (MBK)

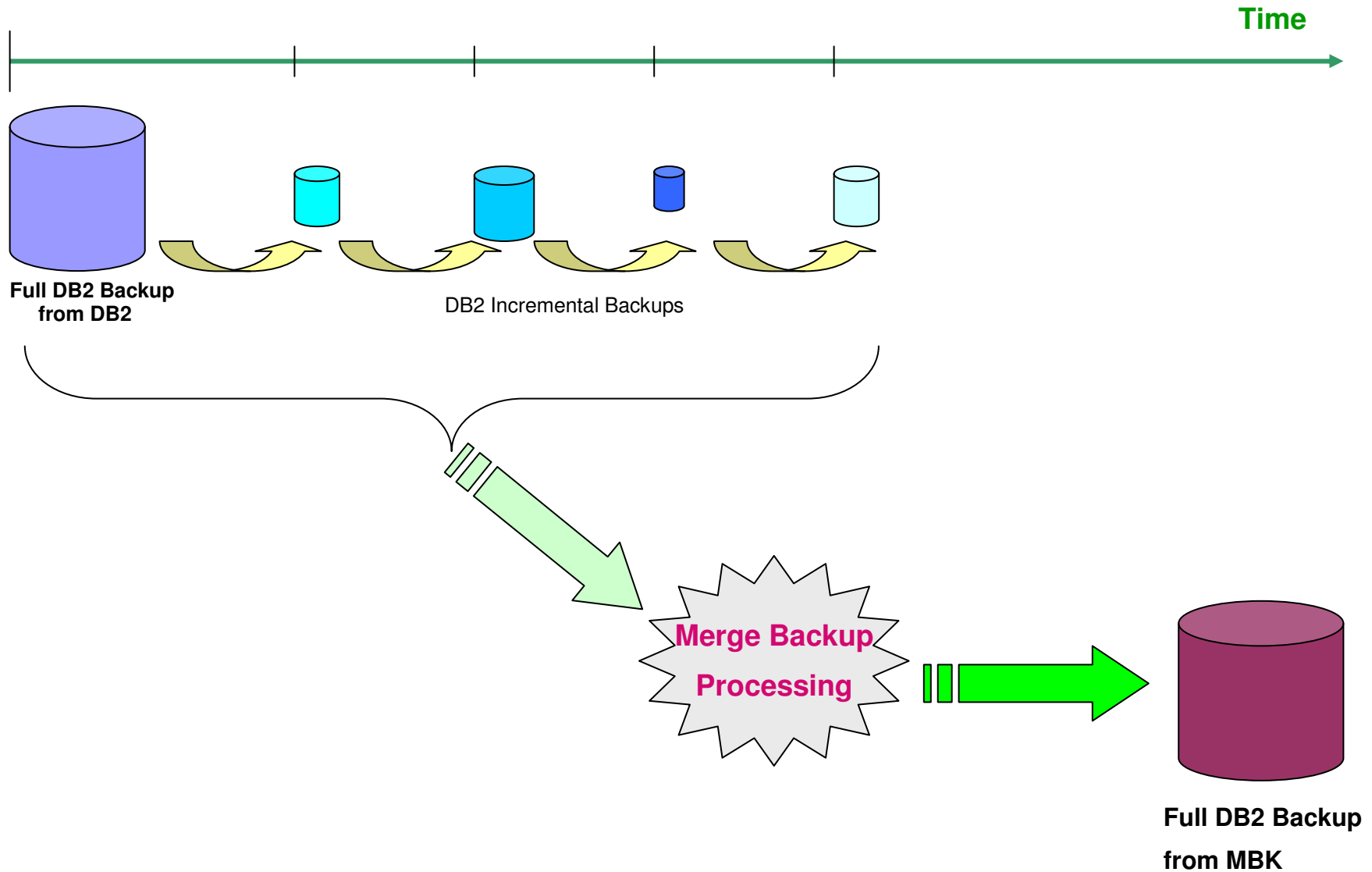
- Merge *full* DB2 backups with DB2 *incremental/delta* backups to build a *new up-to-date full DB2 backup image*
- Eliminates the need to take DB2 full backups
- Use online, offline, compressed Table Space or Database backup images
- Run MBK on the database server or on a **standalone** (remote) machine
- Split out table space backup images from a full backup image
- Supports: DPF, pureScale (as of DB2 V10.5 FP4)



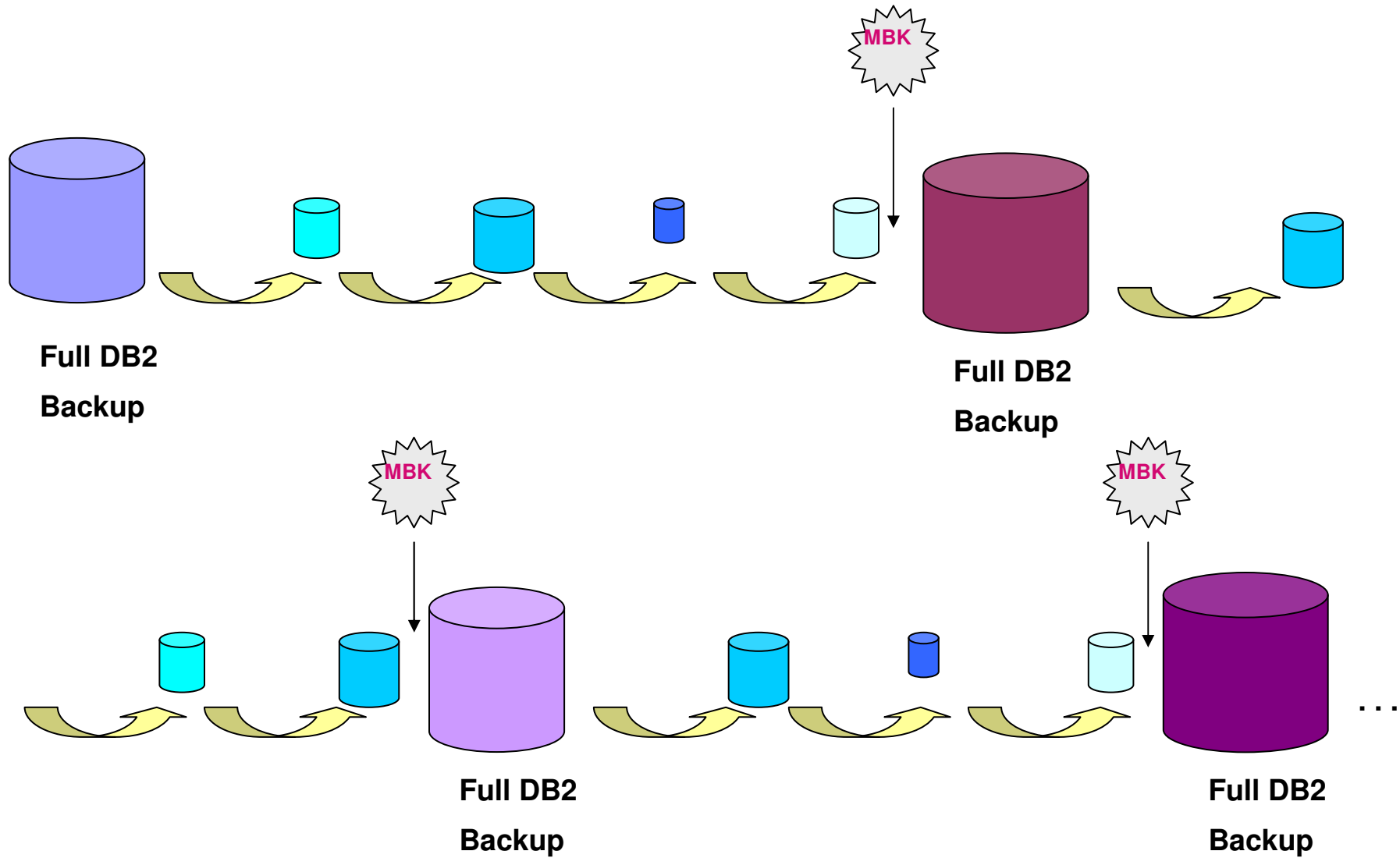
# Benefits of Merge Backup

- Reduce backup intensive resources on the database server
- Reduce backup storage footprint
  - Eliminate full DB2 backups by backing up only what needs to be backed up
- Reduce number of objects required during recovery
  - Simplify recovery process through restore of up-to-date full backup images
  - Speed up recovery with up-to-date more recent full backup images

# Merge Backup Overview



# Eliminate DB2 Full Backups







# Basic Rules for Creating Merge Backup Images

- **When you merge online and offline backups, DB2 Merge Backup determines the output backup type based on the latest input backup type**
  - If the last (latest) incremental (delta) backup is an *online* backup, DB2 Merge Backup creates a full online db2 backup without considering the type of the earlier backups
  - If the last incremental (delta) backup has INCLUDE LOGS, DB2 Merge Backup creates a full db2 backup that includes logs
- **The backups that are involved in a merge cannot involve multiple storage managers**
  - For example, if you want to build a TSM backup, all the backups involved in the merge must also be stored with TSM. DB2 Merge Backup will reject all the configurations that involve several storage managers.



## Merge Backup – Control files

- Example: `db2mbk -f <control_file_name>`
- Control files are ASCII text files that contain a list of MBK options
- Provides additional flexibility to perform complex operations, good for repetitive tasks
- Command line can override control file options so control file can be reused for different configurations without modifying it
  - Example: Use same control file for different databases
    - `db2mbk -d <other_db_name> -f <std_control_file_name>`
- Simple structure to every control file – examples on following slides

# Execution Output example

```
db2mbk -d mytest -t userspace1 -m mbk.msg
```

```
MBKM031I DB2 Merge Backup for Linux, UNIX, and Windows 01.01.000(110505) 64 bits 06/10/11 (AIX  
d487p570 3 5 00C78DEF4C00)
```

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8-----+-----9-----  
+-----10-----
```

```
000001 MERGE DATABASE MYTEST
```

```
000002 TABLESPACE USERSPACE1
```

```
MBKB005I MBK control step start : 10:56:05.057.
```

```
MBKB005I [0] MBK control step start : 10:56:07.538.
```

```
MBKB006I [0] MBK control step end : 10:56:07.793.
```

```
MBKB006I MBK control step end : 10:56:07.793.
```

```
MBKB053I MBK run step start : 10:56:07.793.
```

```
MBKB007I [0] MBK inventory step start : 10:56:07.793.
```

```
MBKB021I [0] The partition 0 backup image taken at 20110610105603 is involved in the merge  
(type DELTA ONLINE DATABASE, device DISK)
```

```
MBKB008I [0] MBK inventory step end : 10:56:07.844.
```

```
MBKB009I [0] MBK merge step start : 10:56:07.844.
```

```
MBKB028I [0] The utility will build the partition 0 backup image taken at 20110610105604 (type  
FULL ONLINE TABLESPACE, device DISK)
```

```
MBKB021I [0] The partition 0 backup image taken at 20110610105559 is involved in the merge  
(type FULL ONLINE TABLESPACE, device DISK)
```

```
MBKB010I [0] MBK merge step end : 10:56:09.736.
```

```
MBKB054I MBK run step end : 10:56:09.737.
```

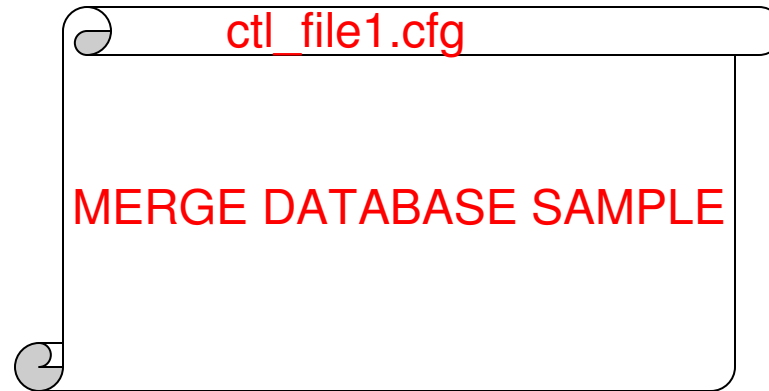
```
MBKI441I MBK successfully ended: real time -> 0m4.679922s
```

```
user time -> 0m0.475853s : parent -> 0m0.475853s, children -> 0m0.000000s
```

```
system time -> 0m0.379995s : parent -> 0m0.379995s, children -> 0m0.000000s
```

# Control File Example 1 – A Simple Merge

With the following control file, DB2 Merge Backup builds a full merged database backup of SAMPLE:



`db2mbk -f ctl_file1.cfg`

What happens:

- Uses the storage, sessions, and compression information from the full DB2 backup that is involved in the merge
- DB2 Merge Backup merges all the backups created since the last full DB2 backup on each database partition of the instance (or on the current DB2NODE partition or the value of the argument of the `-p` command line option).
- The DB2 HISTORY FILE is scanned to determine all the backup images involved in the merge

## Control File Example 2 – Backups in TSM

```
ctl_file2.cfg

MERGE DATABASE SAMPLE
PART (1)
OUTPUT
USE TSM OPEN 2 SESSIONS
COMPRESS NO
```

`db2mbk -f ctl_file2.cfg`

What happens:

- DB2 Merge Backup merges all the backups created since the last full DB2 backup on database partition 1
- The backup is not compressed
- DB2 Merge Backup stores the backup with Tivoli Storage Manager by using two sessions

# Control File Example 3 - Netbackup

ctl\_file3.cfg

```
MERGE DATABASE SAMPLE  
PART (0)  
BEFORE 20130114154242  
OUTPUT  
LOAD "/usr/opensv/netbackup/bin/nbdb2.so"  
COMPRESS YES
```

db2mbk -f ctl\_file3.cfg

What happens:

- DB2 Merge Backup stores the output by using the vendor library/usr/opensv/netbackup/bin/nbdb2.so with one open session
- The DB2 default compression library is used
- DB2 Merge Backup merges all the backups created since the last full DB2 backup and before January 14, 2013 at 15:42:42 on database partition 0

# Example: Table Space Logical Data Recovery

## Scenario:

- Full database backup of PROD on Sundays
- Delta database back nightly
- Wednesday morning it is discovered that batch processing late Tuesday evening inadvertently deleted data from tables in table space TBSP\_SALES

## Resolution:

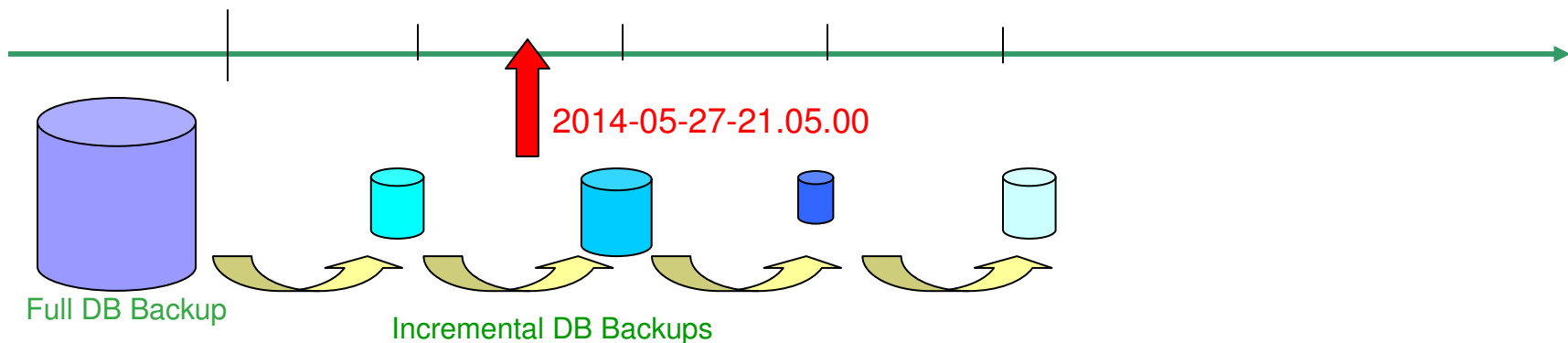
- Use MBK to create full table space backup of TBSP\_SALES
- Restore this table backup to PIT on Tuesday before bad batch was run

## Database Backups:

PROD.0.bminor.DBPART000.20140525201500.001 (Sunday Full DB backup)

PROD.0.bminor.DBPART000.20140526230000.001 (Monday Inc DB backup)

PROD.0.bminor.DBPART000.20140527230000.001 (Tuesday Inc DB backup)



# Table Space Logical Data Recovery (continued)

`db2mbk -f mbk1.ctl -s`

MBKM031I DB2 Merge Backup for Linux, UNIX, and Windows 02.01.00.000.03(140416) 64 bits 05/28/14

000001 merge database sample part(0)

000002 tablespace tbsp1

000003 end from "/home/hotelaix11/bminor/tools/MBK/examp/f2" taken at 20140526230000

MBKB005I MBK control step start : 14:28:44.938.

MBKB037I Merged backups will not be registered into DB2.

MBKB005I [0] MBK control step start : 14:28:44.946.

MBKB006I [0] MBK control step end : 14:28:45.011.

MBKB006I MBK control step end : 14:28:45.011.

MBKB053I MBK run step start : 14:28:45.011.

MBKB007I [0] MBK inventory step start : 14:28:45.012.

MBKB021I [0] The partition 0 backup image taken at 20140526230000 is involved in the merge (type DELTA ONLINE DATABASE, device DISK)

MBKB008I [0] MBK inventory step end : 14:28:45.021.

MBKB009I [0] MBK merge step start : 14:28:45.021.

MBKB028I [0] The utility will build the partition 0 backup image taken at 20140526230001 (type FULL ONLINE TABLESPACE, device DISK)

MBKB021I [0] The partition 0 backup image taken at 20140525201500 is involved in the merge (type FULL OFFLINE DATABASE, device DISK)

MBKB010I [0] MBK merge step end : 14:28:45.304.

MBKB054I MBK run step end : 14:28:45.304.

MBKI441I MBK successfully ended: real time -> 0m0.365831s

`db2 "restore db PROD tablespace (TBSP_SALES) taken at 20140526230001"`

`db2 "rollforward db sample to 2014-05-27-21.00.00 using local time and stop tablespace(tbsp1) online"`



# Split Full Database Backup into Table Space Backup Images

- Use MBK to split a full DB2 backup image (created by DB2 or MBK) into table space backup images

**db2mbk -f mbksplit2.ctl**

**MBKM031I DB2 Merge Backup for Linux, UNIX, and Windows 02.01.00.000.03(140416) 64 bits  
05/30/14**

**000001 split database sample part(0)**

**000002 tablespace userspace1**

**MBKB005I MBK control step start : 14:06:53.222.**

**MBKB022I [0] Nothing to merge for object(s) USERSPACE1**

**MBKB006I [0] MBK control step end : 14:06:53.390.**

**MBKB053I MBK run step start : 14:06:53.391.**

**MBKB007I [0] MBK inventory step start : 14:06:53.391.**

**MBKB008I [0] MBK inventory step end : 14:06:53.423.**

**MBKB009I [0] MBK merge step start : 14:06:53.423.**

**MBKB028I [0] The utility will build the partition 0 backup image taken at 20140530210019 (type FULL ONLINE TABLESPACE, device DISK)**

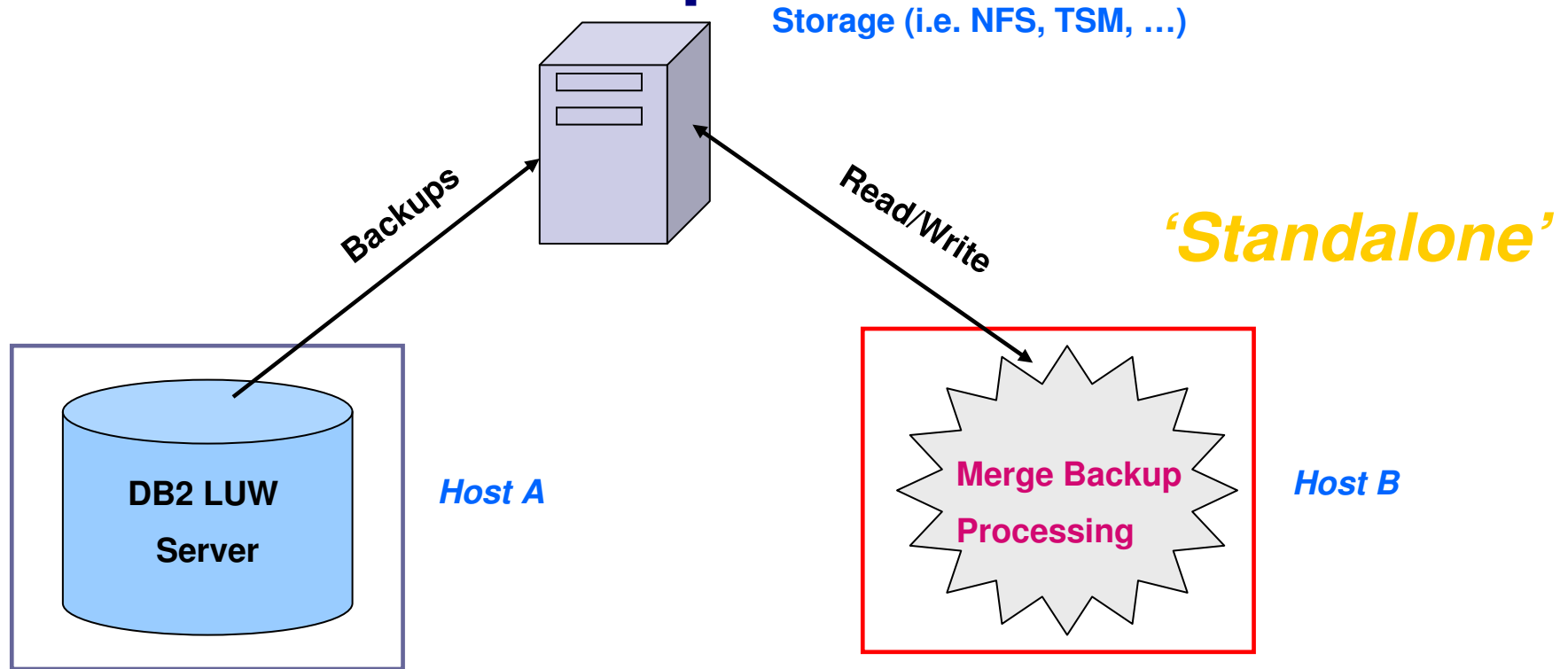
**MBKB021I [0] The partition 0 backup image taken at 20140530210018 is involved in the merge (type FULL ONLINE DATABASE, device DISK)**

**MBKB010I [0] MBK merge step end : 14:06:54.744.**

**MBKB054I MBK run step end : 14:06:54.748.**

**MBKI441I MBK successfully ended: real time -> 0m1.527215s**

# Off-Load Full Backups from Production



- Merge backup is installed on a different server from where backups are taken
- Minimize CPU usage and I/O usage on DB2 server by offloading merge backup processing to standalone machine
- All of the backups involved in the merge must be accessible to the standalone machine

# Control File Example – Standalone Mode


- DB2 Merge Backup builds a full merged backup using a prior full backup created by merge processing as it's base

```
db2mbk mbk_standalone.cfg -s
```

```
MERGE DATABASE SAMPLE PART (0)  
END FROM "/incrBackups/" TAKEN AT 20131005132936  
START FROM "/fullBackups/volume1" TAKEN AT 20131001201545  
OUTPUT TO "/fullBackups/volume2"
```

## Notes:

- The most recent incremental backup is explicitly specified by the **END FROM** clause
  - The history file is extracted from that backup image and used to determine all the backups required for the specified merge
- The base full backup used (so the 'older' full backup into which incrementals are merged) is explicitly specified by the **START FROM** clause
  - If no **START FROM** clause is specified merge processing will search backwards in time from the most recent incremental (here specified by the **END FROM** clause) until it finds the first full backup (into which it will merge)
- The new up-to-date full db2 backup created by merge processing is placed in the directory "/fullBackups/volume2"
- The **PART()** clause is mandatory in standalone mode



## DB2 and TSM – Proxy Nodes

- DB2 backups taken with TSM client are owned by that TSM client
- By default, one TSM client cannot 'see' the backups of another client
- Proxy nodes enable you to consolidate backups and log archives of databases existing on multiple client nodes or under multiple users to a common target nodename on the TSM server

<http://pic.dhe.ibm.com/infocenter/db2luw/v10r5/topic/com.ibm.db2.luw.admin.ha.doc/doc/t0024706.html>



## MBK Standalone with TSM

- Standalone system needs to 'see' or have access to, the DB2 backups generated by the DB2 server
- You can configure TSM and DB2 to use proxy nodes so that all TSM client backups are taken to a common target nodename on TSM server
- **OR**
- Specifically grant access across TSM clients/nodes to backup images using db2adult

## Granting Access to TSM DB2 Backups with db2adutl

Node/Machine: 'Primary' User:'user1'

```
db2 backup db sample use tsm
```

```
...
```

```
db2 backup db sample incremental delta  
use tsm
```

```
db2adutl query db sample
```

```
db2adutl grant user user2 on nodename  
standalone for database sample
```

Node/Machine:'Standalone' User:'user2'

*mergestandalone.ctl:*

```
merge database sample  
part(0)  
start use tsm  
options "-fromnode=primary -fromowner=user1"  
taken at <timestamp_full_backup>  
end use tsm  
options "-fromnode=primary -fromowner=user1"  
taken at <timestamp_latest_incbackup>
```

```
db2adutl query db sample nodename primary  
owner user1
```

```
db2mbk -f mergestandalone.ctl
```

```
db2adutl query db sample
```

```
db2 grant user user1 on nodename primary for  
database sample
```



## MBK Standalone Mode (continued)

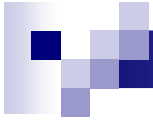
- If using TSM Storage Manager, copy of DB2 must be installed on standalone machine – TSM integrated into DB2 (no licensing issues as long as no database is used on this standalone system)
- MBK will not write entries to the DB2 production server History File for full backups created in standalone mode
- Need to make use of END FROM clause and optionally START FROM clause, in control file
  - END FROM indicates what incremental/delta is last (most recent) in merge chain
  - START FROM allows specification of the old base full backup



# MBK Performance Considerations

- Recovery == Backup + Restore + Rollforward
- Number and size of images to merge
  - More backup targets allows more parallelization on read and write
- Location of backup images: local file system vs storage manager
  - If underlying Storage Manager media tape, cannot read and write to tape at the same time
    - Set cfg parm “sequential\_access=yes” to have new full image staged then once completed, written to storage manager
- Backup compression, encryption, de-duplication, ... all add more effort
- Best Practices:
  - Use incremental delta DB2 backups
  - If using a Storage Manager, keep incremental deltas on local disk if possible
  - Use MBK standalone mode to offload full backup processing from DB2 prod server





# *High Performance Unload*



# Optim High Performance Unload

- High Performance Unload (HPU) is a standalone command line utility for DB2 LUW databases
- HPU performs high-speed, bulk data unloads
  - Unload can be from:
    - Backup images
      - Full, incremental, compressed, DB2, Merge Backup
    - From the database container files



# Why High Performance Unload ?

- Speed 4-20x
- Flexibility over SQL Export – do more
- Data transformation through unload/load vs inplace SQL update
- Unload from backup file(s)
- Data movement - automatic data migration, redistribution



# Using Optim High Performance Unload

- Unload of a table from a database via:

## Command Line:

```
db2hpu -d <dbname> -t <tname> -o <unload_file>
```

## Command Line with control file:

```
db2hpu -d <dbname> -f <control_filename>
```

## Stored Procedure:

```
db2 "call db2hpu('/opt/IBM/DB2TOOLS/HighPerformanceUnload42', '-d  
<dbname> -o <unload_file>', 'unload tablespace select * from <tname>;  
format del;', '?', '?')"
```

## Data Studio

Control files have a general structure of blocks that contain:

- Keyword, options, and values (examples to follow)



# Simple HPU Control File Examples

- **Unload a *Database***

```
GLOBAL CONNECT TO SAMPLE;  
UNLOAD DATABASE  
OUTFILE("\\dataDB\sampDB\");
```

- **Unload a *Table Space***

```
GLOBAL CONNECT TO SAMPLE;  
UNLOAD TABLESPACE USERSPACE1  
OUTFILE("\\dataDB\sampDB\tableSpace\");
```

- **Unload a *Table***

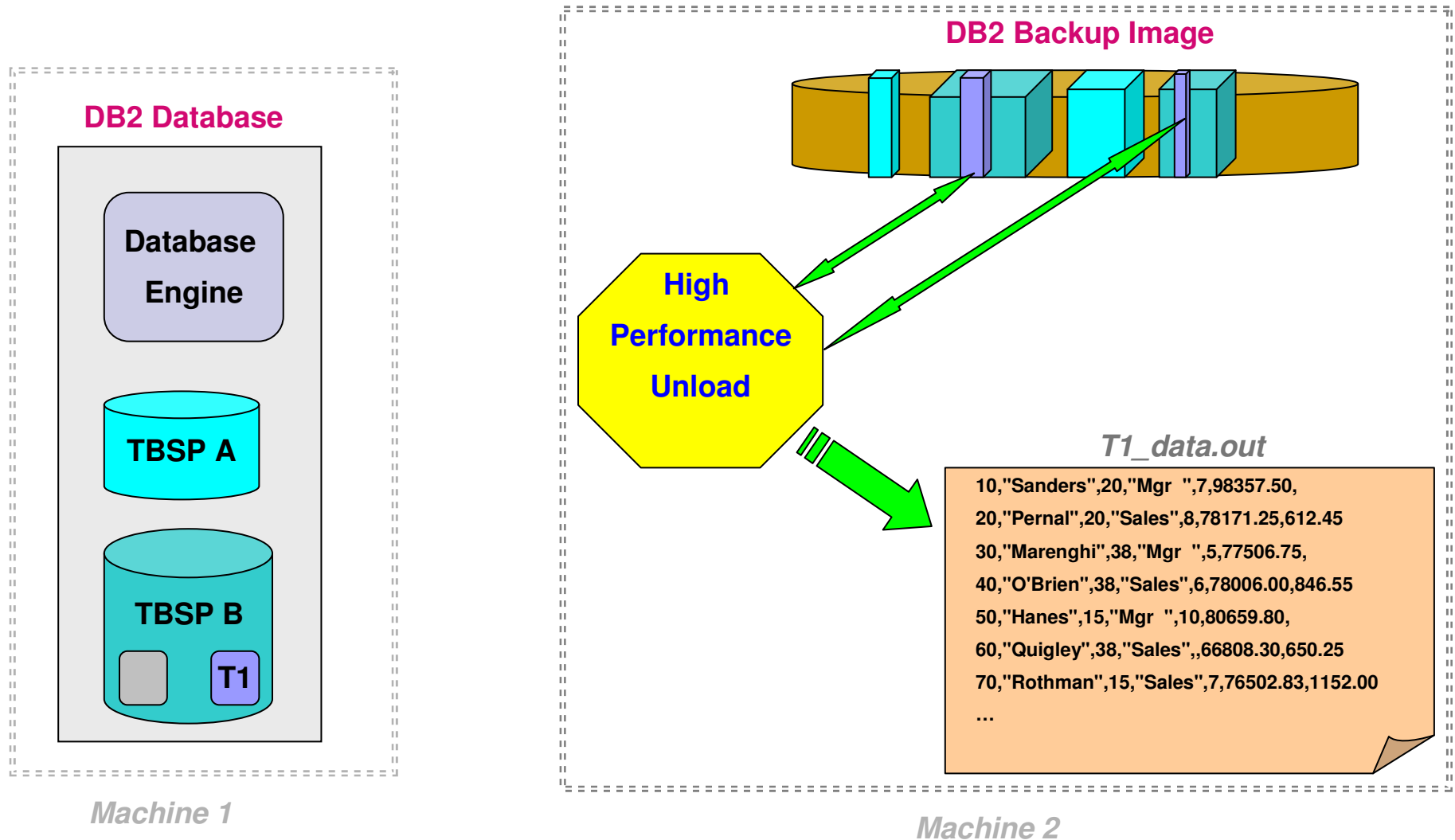
```
GLOBAL CONNECT TO SAMPLE;  
UNLOAD TABLESPACE USERSPACE1  
SELECT * FROM EMPLOYEE;  
OUTFILE("\\dataDB\sampDB\table\");
```

# Data Studio and HPU

The screenshot displays the IBM Data Studio interface. The main window shows a list of tables in the 'SAMPLE' database. The 'EMPLOYEE' table is selected, and a context menu is open over it. The 'Unload' option is highlighted, and a sub-menu is visible with 'With Optm High Performance Unload...' selected. A red circle highlights the 'Unload' and sub-menu area.

Schema	Name	Perc...	Row Co...	Log Mo...	Primary Key	Partitio...	Regular Ta...	Index 1
BMINOR	ACT	0			PK_ACT		USERSPACE1	
BMINOR	CATALOG	0			PK_CATAL...		IBMDB2SA...	
BMINOR	CL_SCHED	0					USERSPACE1	
BMINOR	CUSTOMER	0			PK_CUSTO...		IBMDB2SA...	
BMINOR	D1	0	1				IBMDB2SA...	
BMINOR	DEPARTMENT	0			PK_DEPART...		USERSPACE1	
BMINOR	EMPLOYEE	0	12		PK_EMPLO...		USERSPACE1	
BMINOR	EM...						IBMDB2SA...	
BMINOR	EM...						USERSPACE1	
BMINOR	EM...				PK_EMP_PH...		USERSPACE1	
BMINOR	EM...							
BMINOR	INV...							
BMINOR	IN...							
BMINOR	MY...						MYTBSpace	
BMINOR	OR...						USERSPACE1	
BMINOR	PRO...				PK_PRODU...		IBMDB2SA...	
BMINOR	PRO...				PK_PRODU...		IBMDB2SA...	
BMINOR	PRO...				PK_PROJECT		USERSPACE1	
BMINOR	PRO...				PK_PROJECT		USERSPACE1	
BMINOR	DI...				PK_DI...		IBMDB2SA...	

# Unload Table Data from DB2 Backup Image





## Example: Ctl File Unload from Backup

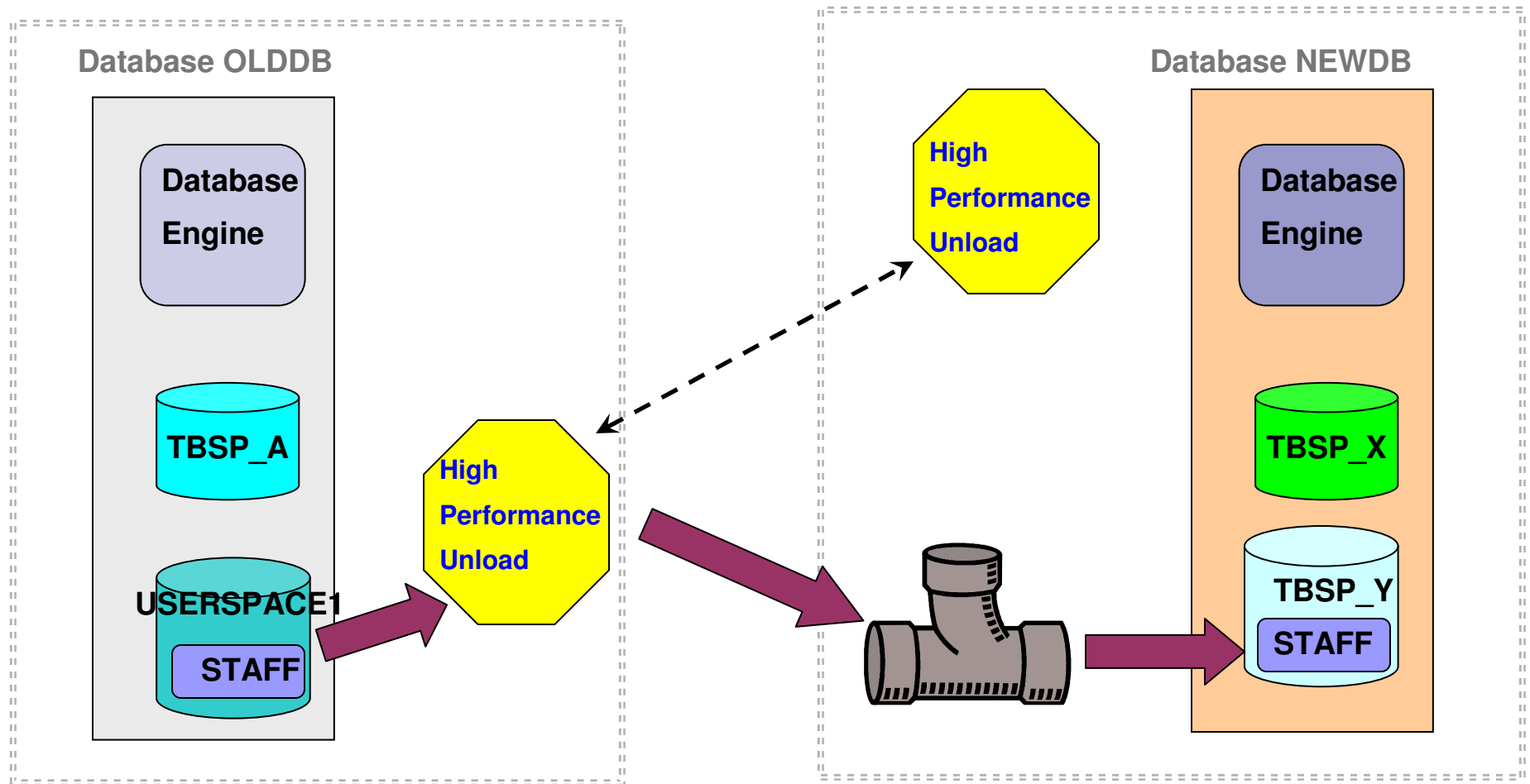
```
GLOBAL CONNECT TO DBPROD
using backup catalog DBPROD LOAD "/usr/opencv/netbackup/bin/nbdb2.sl64"
taken at 20140314200104;
;
UNLOAD TABLESPACE
PART (1:8)
LOCK NO QUIESCE NO DB2 NO
using backup database DPPROD LOAD "/usr/opencv/netbackup/bin/nbdb2.sl64"
ON SOURCE HOST taken at 20140314200104;
SELECT * FROM S.FACT;
OUTFILE ( "/fs1/SFACT.del1-8" REPLACE )
LOADFILE ("/fs1/SFACT.load1-8")
FORMAT DELIMITED DOUBLE DELIM ON;
```



# Automatically Migrate/Move Data

*Machine A: AIX DB2 V9.5*

*Machine B: Linux DB2 V10.1*





# HPU Automatic Data Migration

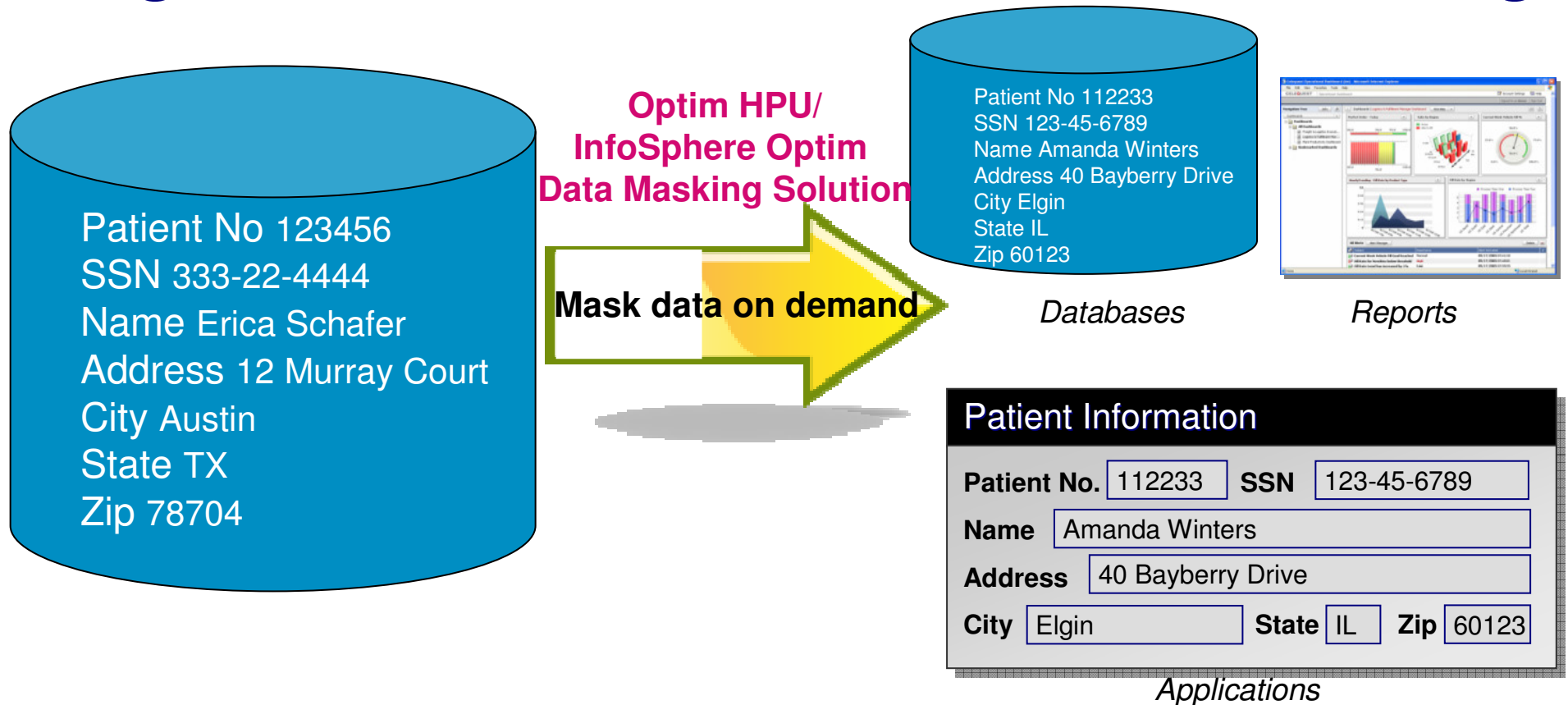
- Migrate data from AIX V9.5 “OLDDB” (machineA) to Linux V10.5 “NEWDB” (machineB)

HPU Control File “datamigr.ctl”:

```
GLOBAL CONNECT TO olddb;  
MIGRATE TABLESPACE USERSPACE1  
only tables (BMINOR.STAFF)  
TARGET ENVIRONMENT (INSTANCE "bminor" ON "machineB" IN "newdb")  
WORKING IN ("/home/bminor/wrk/HPU/test/datamig/staging")  
FORMAT MIGRATION  
;
```

```
machineA:> db2hpu -f datamigr.ctl
```

# High Performance Unload and Data Masking



- ✓ Use Optim High Perf. Unload to extract from database or backup
- ✓ Call the Optim Data Masking Libraries while Unloading
- ✓ Unload and Mask using one command
- ✓ Get additional benefits by Using Unloaded Data with InfoSphere Optim Solution for subsetting, extracting data

# Example – HPU Data Masking Credit Card Data

```
db2 "select empno, ccn1 from emp-ccn"  
EMPNO CCN1
```

```
-----  
000010 18002748513298  
000020 30000019965395  
000030 12345678901234
```

```
db2hpu -f emp_ccn.ctl -i i958  
INZM031I Optim High Performance Unload for DB2 for Linux, Unix and Windows 05.01.00.000.01(130912)
```

```
000001 GLOBAL CONNECT TO SAMPLE  
000002 DATAMASKING ODPP PATH "/opt/odpp" VERSION "9.1"  
000003 ;  
000004 UNLOAD TABLESPACE  
000005 SELECT EMPNO, CCN1 FROM EMP_CCN  
000006 MASKED COLUMNS (CCN1 "provider=ccn, flddef1=(name=ccn1,dt=char),method=random")  
000007 OUTFILE ("outfile")  
000008 FORMAT DEL;  
000009
```

```
...  
INZU410I HPU utility has unloaded 3 rows on lat186 host for I958.EMP_CCN in outfile.  
INZU468I HPUrun step end : 16:36:57.890.  
INZI441I HPU successfully ended: Real time -> 0m0.459566s
```

```
cat outfile  
"000010","6011080000000014 "  
"000020","4170260000000014 "  
"000030","3742460000000015 "
```



# HPU Performance Considerations

- Remote vs. Local unload
- Database vs. Backup unload
- Storage Manager vs. Local Disk
- Degree of 'partitioning': table (i.e range) and/or database
- Number of unload files
- Pipe vs. File
- Nature of the unload (SELECT) statement
- Data characteristics: DB2 compressed, encrypted, LOB/XML



# *Recovery Expert for DB2 LUW*



# IBM Recovery Expert for DB2 LUW

- A *log analysis* and *recovery tooling* that manages DB2 systems
- Uses DB2 *log files* and available recovery assets i.e. backups
- *Simple* to use *Web-based* Graphical User Interface
  - Command Line component for batch processing
- There is *minimal disruption* to DB2



# Recovery Expert Capability

- Log Analysis
  - Validate change; look for efficiencies in system processing: commit scope, quiet time
- Application and transaction recovery
- Fine grained recovery via SQL Redo or SQL Undo
  - Reduced recovery impact; accelerated recovery (DB2 MRT)
- Clone via SQL Redo
- Simplify Recovery Asset Management and Recovery Execution





# Table Level Recovery

- RE recover dropped table without performing Table Space or Database RESTORE and ROLLFORWARD
- Advantages:
  - Other objects in table space remain online
  - No expensive restore operation
  - No expensive rollforward operation
    - DDL changes cause Table Space Minimum Recovery Time (MRT) value to change
    - Rollforward must be to at least MRT
    - Can require a large number of logs to be replayed
  - All dependent objects recovered

# Recover a Table (Along with its Dependent Objects)

Dependencies

List of all dependent objects

- Group for DBAPOT.NEW\_TABLE
  - Indexes
    - DBAPOT.NEW\_INDEX
  - Schemas
    - DBAPOT
  - Tablespaces
    - USERSPACE1
  - Views
    - DBAPOT.NEW\_VIEW

1. Dropped object  
Find one or more dropped objects to restore. In this example, a table.

3. Scenarios  
Multiple scenarios can be determined. Review for the one that works best for you.

2. Dependencies  
All dependencies automatically determined and ready to be restored

Created on 2011-06-21 17:20:39 Status ✔ Aryplan generate scenario successfully.

Session ID 2

Active scenario Recreate object definitions from the SLR and generate Redo SQL (1) ▾

Scenario steps			Restore all	Selected step details	
Step	Description	Partition		Option	
1	Drop existing objects	0			
2	Run recovery components	0			
3	Generate Redo SQL				
4	Apply SQL	0			

4. Steps  
Each scenario is made of steps where you can review each step before running.



# Log Analysis: Use-Cases

- Monitor data changes and isolate undesired/accidental changes
  - Large unexplained spikes in log activity
- Transactions not committing
- Determine which objects 3<sup>rd</sup> party applications are involved with
- Correct logical data errors via Undo SQL generation
  - Ran monthly batch job instead of weekly
  - Delete script with wrong parameters
- Who changed what and when?
  - Someone dropped a table
    - What indexes existed on it?
  - Generate regular audit report for change activity



# Log Analysis Reports

- **Report Filtering by:**
  - Application ID, application name, table name and ID, table space name and ID, Log Sequence Number (LSN), transaction ID, partition group, authorization ID, schema
  - Include or exclude filtering and wildcards
  - Post filtering to apply different filters to a Log Analysis report
- **Operations Filtering by:**
  - Insert, Update, Delete
  - Transaction status: Committed, Partial, Uncommitted, Rolled Back
- **Report Types:**
  - Summary or Summary + Detail
- **Report Formats:**
  - CSV, XML, text

# Sample Log Analysis Report

## STATISTICS

NGNAME	UPDATES	INSERTS	DELETES
IBMCATGROUP	1966	160	0
IBMDEFAULTGROUP	0	24	0

TSNAME	UPDATES	INSERTS	DELETES
IBMDB2SAMPLEREL	0	24	0
SYSCATSPACE	100	159	0
SYSTOOLSPACE	1866	1	0

LOCAL TRANSACTION ID	COMMIT	LSN	DATE	TIME	STATE	AUTHID	GLOBAL TRANSACTION ID
00000001FB2C	0000000059D8536	2013-10-03	15:25:49.000001	COMMITTED			

NGNAME	TSNAME	TABLE OWNER	TABLE NAME	ACTION	PARTITION	LSN
IBMDEFAULTGROUP	IBMDB2SAMPLEREL	BILLM	ORG2	I	0	0000000055F23A7
IBMDEFAULTGROUP	IBMDB2SAMPLEREL	BILLM	ORG2	I	0	0000000055F240B
IBMDEFAULTGROUP	IBMDB2SAMPLEREL	BILLM	ORG2	I	0	0000000055F246B

. . .

TABLE	UPDATES	INSERTS	DELETES
BILLM.ORG2	0	24	0
SYSIBM.SYSCOLDIST	0	150	0

. . .

## TOTAL STATISTICS

TOTAL INSERTS : 184  
 TOTAL UPDATES : 1,966  
 TOTAL DELETES : 0



# Recovery Expert Application Recovery

- ***Undo* all the changes made by a specific application**
  - Without knowing in advance what objects were processed
  - RE Application level recovery handles cases where it is the application that maintains Referential Integrity between objects
- ***Redo* a specific set of changes**
  - Produces SQL statements to reapply the changes made by the original database log operations, and optionally includes DDL statements

# Log Analysis with Undo

The screenshot shows the IBM Log Analysis tool interface. The main window displays a list of tables affected by transactions and a detailed view of a specific transaction and its operations. Callouts provide instructions on how to use these features.

**1. Tables**  
Show which tables any transactions have affected

**2. Transaction**  
Select the transaction you wish to review

**3. Single DML operation**  
See each DML operation for that transaction

**4. Undo**  
Review the Undo SQL for each DML operation  
Run Undo SQL if desired

Local Transac	Transaction L	Date	Time	Status
000000000321	0000000003712	2011-06-18	00:24:14.00000	Comm
000000000327	0000000003EE0	2011-06-18	00:24:28.00000	Comm
000000000331	0000000003EE0	2011-06-18	00:24:28.00000	Comm
00000000018C5	00000000065E2	2011-06-18	18:24:43.00000	Comm

LSN	Action	Node	Partition grou	Tables
0000000003711	Insert	0	IBMDEF	
0000000003712	Insert	0		

DELETE FROM "DBAPOT"."NEW\_TABLE" WHERE "COL1" = 'TEST1 ' AND "COL2" = 1



# 'Quiet Time' Reporting

- Quiet Time report displays periods of inactivity on a selected database
- Useful for determining:
  - When to schedule maintenance
  - Assessing system utilization and efficiency
  - Find Point-In-Time (PIT) for Database or Table Space recovery



# Quiet Time Report Example

IBM DB2 Recovery Expert

Open ↓ Log Analysis

Result - SAMPLE × New Tab

[Overview](#) Display the Log Analysis results where one can view report statistics, set post filters to modify the report data, and see the report details.

[Location](#) Host: 192.168.0.10, Instance: DB2\_01, Target database: SAMPLE, DB2 version: 10.1.1

[Details](#) [Post filters](#) [Report](#)

[Objects](#)

[Filters](#)

[Options](#)

[Status](#)

[Result](#)

Totals

Totals

Database Partition G

Tablespaces

Tables

Application names

Quiet times 1 - 13 out of 1226910

Start timestamp	End timestamp
2013-02-06 10:44:54.000000	2013-02-20 15:33:24.000000
2013-02-20 16:16:25.000001	2013-02-20 16:46:40.000000
2013-02-20 16:46:41.000001	2013-02-25 15:00:58.000000
2013-02-25 15:03:30.000001	2013-02-28 14:44:15.000000
2013-02-28 14:51:35.000001	2013-03-01 11:12:03.000000
2013-03-01 11:13:28.000001	2013-03-02 22:03:03.000003
2013-03-02 23:25:34.000001	2013-03-13 09:14:49.000003



# Recovery Expert Performance Considerations

- Number of objects to recover
- Location of recovery assets: local disk vs storage manager
- Amount or size of assets (logs, backups) to process
- Compression, encryption, deduplication
- DATA CAPTURE CHANGES (DCC) and Currently Committed semantics (for optimal SQL UPDATE processing)



## Advanced Recovery Tooling Customer Use-Cases

- Efficiently migrate to different OS, Hardware platform, new DB2
- Implement a Test Data Management system
- Extract data from a database archive (for recovery, audit, ...)
- Unload and mask sensitive production data
- Simply recover any database object to any point in time
- Generate detailed database change activity reports
- Undo a bad transaction while keeping database online
- Reduce database backup storage footprint
- Simplify and speed up database/table space recovery
- . . .
- Less effort/cost managing your data and more time leveraging it



# Additional Reference Material

## ChannelDB2 :

DB2 Recovery Expert for LUW

<http://www.channeldb2.com/video/ibm-db2-recovery-expert-for-luw-part-1>

DB2 Merge Backup for LUW

<http://www.channeldb2.com/video/ibm-db2-merge-backup-for-luw-part-2>

Optim High Performance Unload

<http://www.channeldb2.com/video/infosphere-optim-high-performance-unload-for-db2-for-luw-part-3>

## YouTube :

DB2 Recovery Expert for LUW

<https://www.youtube.com/watch?v=6PTC9Ray9fY>

DB2 Merge Backup for LUW

<https://www.youtube.com/watch?v=u9XlotbrzAo>

Optim High Performance Unload

<https://www.youtube.com/watch?v=KAyGAMdKx9Y>

## developerWorks:

[Building a data migration strategy with IBM InfoSphere Optim High Performance Unload](#)

[Using IBM InfoSphere Optim High Performance Unload as part of a recovery strategy in an IBM Smart Analytics System](#)



# Concierge Program

- Customized Product Deployment Guidance

*A service to assist our Clients through  
planning and execution of installation and  
configuration of our offerings*