Practical Recipes for Daily DBA Activities using DB2 9 and 10 for z/OS

Session: IDZ-1012A

Cüneyt Göksu, VBT
IBM Gold Consultant

IBM Software
Information On Demand 2011

Agenda

- DBA Activities

- DBA's nightmare > Drop Table
- My first Trusted Context definition
- Include columns for Indexes in V10
- Index on expression
- MAXTEMP_RID in V10

Some DBA Activities

- Application or Object Recovery
  - Undo/Redo SQL
  - Tablespace/Table Recovery
- Operational changes to DB Design (DDL execution)
- SQL Performance & Tuning
- Physical Design from Logical Design
- Running DB2 Utilities(Backup, Maintenance)
- Move/Copy Data

DBA's nightmare: Drop Table

- Assumptions
  - There is a proper FIC
  - DBA is informed immediately
  - DBID, PSID, OBID is kept safely or accessible thru IC
  - One Table / Tablespace
  - No ISV Tools for Drop Table Recovery
  - No RESTRICT ON DROP definition
  - Playground Setup
    - Database : DB1
    - Tablespace : TS1
    - Table: T1 C1 INTEGER NOT NULL WITH DEFAULT,
      C2 CHAR(3) NOT NULL WITH DEFAULT

DBA's nightmare: Drop Table

- DBA, Application Developer, CTO, Prime Minister, President etc... Someone accidentailly issued

```
DROP TABLE T1
```

DBA's nightmare: Drop Table

- Step 1

Find DBID, PSID, OBIDs of the dropped Object BEFORE Drop

1) Use SQL from Catalog

```
-- GET DBID PSID : DBID=326 PSID=2
SELECT NAME,DBNAME,DBID,PSID,NTABLES
FROM SYSIBM.SYSTABLESPACE WHERE NAME='TS1';

--GET OBID : DBID=326 OBID=3
SELECT NAME,CREATOR,DBNAME,TSNAME,DBID,OBID,COLCOUNT
FROM SYSIBM.SYSTABLES WHERE NAME='T1' ;
```

2) Use Latest FIC with DSN1PRNT

```
//STEP1    EXEC PGM=DSN1PRNT,PARM='PRINT,FORMAT'
//SYSUT1   DD DSN=DB2DV0BB.DB1.TS1.P00000.D16221.F.L,DISP=SHR

HPGOBID='01460002'X   (dbid 0X0146=326 psid 0x0002=2)
PGSOBD='0003'X      (obid 0x0003=3)
```

DBA's nightmare: Drop Table

- Step 2

Switch Active Logs for DSN1LOGP Utility.

-ARCHIVE LOG

```
DSNJ003I  -GT9G DSNJOFF3 FULL ARCHIVE LOG VOLUME 182
DSNAME=DSN910GT.ARCLOG1.D11094.T0546360.A0000173,
STARTRBA=000168EF4000, ENDRBA=00016A9FFFFF, STARTTIME=C7723D00D897,
ENDTIME=C79269891DF9, UNIT=SYSDA, COPY1VOL=HLS004, VOLSPAN=00,
CATLG=YES
DSNJ138I  -GT9G DSNJOFF1 OFFLOAD COMPLETE FOR LOG 183
TRUNCATION FROM AN ARCHIVE LOG COMMAND, ENDRBA=00016A9FFFFF
DSNJ139I  -GT9G LOG OFFLOAD TASK ENDED
```

DSN910GT.ARCLOG1.D11094.T0546360.A0000173

DSN910GT.ARCLOG1.D11094.T0546360.B0000173

DBA's nightmare: Drop Table

- Step 3

Run REPORT RECOVERY Utility to find Date & Time of DROP Command:

REPORT RECOVERY TABLESPACE DB1.TS1

```
DSNU583I  -GT9G 094 05:58:47.92 DSNUPPLR - SYSLGRNX ROWS FROM REPORT RECOVERY FO
 UCDATE     UCTIME      START RBA      STOP  RBA     START LRSN     STOP  LRSN
 040411    03414009   00016A9B2A0B   00016A9B4B62   C7924D9C8471   C7924D9C8C85
 040411    03414028   00016A9B6550   00016A9B9499   C7924D9CB2DF   C79250287CCF
 040411    04200028   00016A9C0A51   00016A9C1389   C792562E260D   C79257432E6B
 040411    05434888   00016A9FBD22   00016A9FF9C1   C79268E9CAD0   C7926B5312A0

DSNU584I  -GT9G 094 05:58:47.93 DSNUPPBS - REPORT RECOVERY TABLESPACE DB1.TS1 AR
  START TIME       END TIME    START RBA      END RBA    UNIT  VOLSER   DATA SET
20110682136586 20110941146359 000168EF4000 00016A9FFFFF SYSDA  HLS004 DSN910GT.A
```

DBA's nightmare: Drop Table

- Step 4

Run DSNJU004 to know which Archive Log Dataset contains RBA Range around the time of Drop

```
//DSNTLOG EXEC PGM=DSNJU004
//SYSUT1   DD  DISP=SHR,DSN=DSN910GT.BSDS01
```

```
 START RBA/TIME          END RBA/TIME         DATE     LTIME
--------------------  --------------------  --------  -----
  000168EF4000            00016A9FFFFF        2011.094    5:46 DSN=DSN910GT.ARCLO
2011.068  21:36:58.6  2011.094  11:46:35.9                      PASSWORD=(NULL) VO
                                                               CATALOGUED
```

DBA's nightmare: Drop Table

- Step 5

Run DSN1LOGP on Archive Log Dataset which contains time of Drop

```
//STEP1 EXEC PGM=DSN1LOGP
//SYSPRINT DD SYSOUT=*
//SYSSUMRY DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//ARCHIVE DD
DISP=SHR,DSN=DSN910GT.ARCLOG1.D11094.T0546360.A0000173
//SYSIN DD *
DATAONLY(YES) SUMMARY(YES)
RBASTART (000168EF4000) RBAEND (00016A9FFFFF)
DBID (0146) OBID(0003)
/*
```

DBA's nightmare: Drop Table

- Step 6

Examine output of DSN1LOGP to find the RBA of Drop

```
LPRT UR  CONNID=BATCH     CORRID=GOLD123J      AUTHID=GOLD123   PLAN=DSNTP491
START DATE=11.094 TIME=05:43:48  DISP=COMMITTED           INFO=COMPLETE
STARTRBA=00016A9FBB1A  ENDRBA=00016A9FF9F5  STARTLRSN=C79268E9CA11  ENDLRSN=C79268E9D571
NID=* LUWID=GTHLSZ9.H3A5.C79268E9C0BC.0001
COORDINATOR=* PARTICIPANTS=*
DATA MODIFIED:
     DATABASE=0146           PAGE SET=0000
     DATABASE=0146=DB1       PAGE SET=0002=TS1
     DATABASE=0006=DSNDB06   PAGE SET=0074=DSNDCX01
     DATABASE=0006=DSNDB06   PAGE SET=015E=DSNDCX02
     DATABASE=0006=DSNDB06   PAGE SET=0009=SYSDBASE
     DATABASE=0006=DSNDB06   PAGE SET=0061=DSNATX01
     DATABASE=0006=DSNDB06   PAGE SET=0062=DSNATX02
     DATABASE=0006=DSNDB06   PAGE SET=00D3=DSNATX03
     DATABASE=0006=DSNDB06   PAGE SET=02F5=DSNATX04
     DATABASE=0006=DSNDB06   PAGE SET=005D=DSNDTX01
     DATABASE=0006=DSNDB06   PAGE SET=00A4=DSNDTX02
     DATABASE=0006=DSNDB06   PAGE SET=02F3=DSNDTX03
     DATABASE=0001=DSNDB01   PAGE SET=001F=DBD01
```

DBA's nightmare: Drop Table

- Step 7

Recover Tablespace just before the DROP Command

```
STARTRBA=00016A9FBB1A (Time Of Drop) - 1 = 00016A9FBB19

//RECOVER EXEC DSNUPROC
//SYSPRINT DD SYSOUT=*
//SYSIN    DD  *
  RECOVER TABLESPACE DB1.TS1 TORBA X'00016A9FBB19'
```

DBA's nightmare: Drop Table

- Step 8

Create a VSAM copy of the original Tablespace

```
//SYSIN    DD  *
   DEFINE CLUSTER(NAME('GOLD123.DSNDBC.DB1.TS1.COPY') -
     MODEL('DSN910GT.DSNDBC.DB1.TS1.I0001.A001'))

IDCAMS  SYSTEM SERVICES
   DEFINE CLUSTER(NAME('GOLD123.DSNDBC.DB1.TS1.COPY') -
     MODEL('DSN910GT.DSNDBC.DB1.TS1.I0001.A001'))
IDC0508I DATA ALLOCATION STATUS FOR VOLUME DB9GT2 IS 0
IDC0512I NAME GENERATED-(D) GOLD123.DSNDBC.DB1.TS1.COPY.DATA
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

DBA's nightmare: Drop Table

- Step 9

DSN1COPY from Original TS to the copy

```
-STO    DATABASE(DB1) SPACENAM(TS1)

//STEP1 EXEC PGM=DSN1COPY,PARM='CHECK'
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DISP=OLD,DSN=DSN910GT.DSNDBC.DB1.TS1.I0001.A001
//SYSUT2    DD DISP=OLD,DSN=GOLD123.DSNDBC.DB1.TS1.COPY

-STA     DATABASE(DB1) SPACENAM(TS1)
```

DBA's nightmare: Drop Table

- Step 10

Recreate the Original table and find new OBID

```
CREATE TABLE T1
(C1 INTEGER NOT NULL WITH DEFAULT,
 C2 CHAR(3) NOT NULL WITH DEFAULT) IN DB1.TS1;
```

```
SELECT NAME,CREATOR,DBNAME,TSNAME,DBID,OBID,COLCOUNT
  FROM SYSIBM.SYSTABLES WHERE NAME='T1';
```

```
--DBID=326 OBID=4
```

DBA's nightmare: Drop Table

- Step 11

DSN1COPY from the copy to the Original TS with OBID Translation

```
-STO     DATABASE(DB1) SPACENAM(TS1)
//STEP1 EXEC PGM=DSN1COPY,PARM='OBIDXLAT,RESET,CHECK'
//SYSUT2   DD DISP=OLD,DSN=DSN910GT.DSNDBC.DB1.TS1.I0001.A001
//SYSUT1    DD DISP=OLD,DSN=GOLD123.DSNDBC.DB1.TS1.COPY
//SYSXLAT   DD *
326,326
 2,2
 3,4
-STO     DATABASE(DB1) SPACENAM(TS1)
```

```
SELECT * FROM T1
```

My first Trusted Context

- What is it ?
  - Establishes trust between DB2 and an external entity such as Application Server
  - Trusted context allows you to associate a role to DB2 thread or switch the userid associated with the connection without requiring credentials.
  - It DOES NOT control general access to DB2 or DDF.
  - A series of trust attributes are evaluated at connect time to determine if a specific context is to be trusted.
  - Once established, a trusted connection provides the ability to

    a) Efficiently switch user with optional authentication

    b) Acquire special set of privileges using a Role

My first Trusted Context

- Requirement was simple
  - Switch the userid associated with the connection without requiring credentials

Step 1 Establish Connection between Client and DB2

```
db2 => connect to dv0bploc user FIDEMIG
Enter current password for FIDEMIG:

   Database Connection Information

 Database server        = DB2 z/OS 9.1.5
 SQL authorization ID   = FIDEMIG
 Local database alias   = DV0BPLOC

DSNL200I  -DV3B DISPLAY LOCATION REPORT FOLLOWS-
LOCATION                                PRDID    REQSTR SERVER CONVS
::10.168.21.70                                        0      1      1
::10.168.27.48                                        0      1      1
```

My first Trusted Context

Step 2: Access to an unauthorized resource from that connection

```
db2 => select count(*) from db2sys1.plan_table

SQL0551N  "FIDEMIG" does not have the required authorization or privilege to
perform operation "SELECT" on object "DB2SYS1.PLAN_TABLE".  SQLSTATE=42501
```

My first Trusted Context

Step 3: Create a Trusted Context and Test again

```
CREATE TRUSTED CONTEXT CTX1 BASED UPON CONNECTION USING SYSTEM AUTHID FIDEMIG
ATTRIBUTES (ADDRESS '10.168.21.70') ENABLE WITH USE FOR DB2SYS1 WITH AUTHENTICATION;

db2 => connect to dv0bploc user FIDEMIG

DSNL200I  -DV1B DISPLAY LOCATION REPORT FOLLOWS-
LOCATION                                PRDID    REQSTR SERVER CONVS
::10.168.21.70                                        0      1      1
  TRUSTED = 'Y'

db2 => select count(*) from db2sys1.plan_table
SQL0551N  "FIDEMIG" does not have the required authorization or privilege to
perform operation "SELECT" on object "DB2SYS1.PLAN_TABLE".  SQLSTATE=42501
```

Why ?

NO DEFAULT ROLE or DEFAULT ROLE role-name

NO DEFAULT ROLE
The authorization ID of the process is the owner of any object that is created using a trusted connection that is based on this trusted context.
NO DEFAULT ROLE is the default.

My first Trusted Context

Step 4: Create a Role and Test again

```
CREATE ROLE CTXROLE;
GRANT SELECT ON TABLE db2sys1.plan_table TO ROLE CTXROLE;

CREATE TRUSTED CONTEXT CTX1 BASED UPON CONNECTION USING SYSTEM AUTHID FIDEMIG
DEFAULT ROLE CTXROLE ATTRIBUTES (ADDRESS '10.168.21.70') ENABLE WITH USE FOR DB2SYS1 WITH
AUTHENTICATION;

db2 => connect to dv0bploc user FIDEMIG

DSNL200I  -DV1B DISPLAY LOCATION REPORT FOLLOWS-
LOCATION                                 PRDID    REQSTR SERVER CONVS
::10.168.21.70                                        0      1      1
  TRUSTED = 'Y'

db2 => select count(*) from db2sys1.plan_table
-----------
         74
  1 record(s) selected.
```

Include columns for Indexes

- New Performance Enhancement in V10
- The UNIQUE clause must be specified
- Columns that are specified in the INCLUDE count towards the limits. e.g: #columns = 64
- INCLUDE cannot be specified for
  - . non-unique index
  - . A partitioning index when index-controlled partitioning is used
  - . An auxiliary index
  - . An XML index

Include columns for Indexes

- T1 (IPLNG_LOCAL_TS TIMESTAMP, IPLNG_WS_ID CHAR (8))
- No Index

SELECT MAX(IPLNG_WS_ID)
FROM T1 WHERE TIME(IPLNG_LOCAL_TS) = '13.02.49‘

Table space scan, no index will be used.

Standard sequential PREFETCH will be performed.

```
  Estimated   Estimated
Serv. Units    Proc. ms
          *           *
----------- -----------
          5           3
```

Include columns for Indexes

UNIQUE INDEX IX1 ON T1 (IPLNG_LOCAL_TS) ;

Table space scan, no index will be used.

Standard sequential PREFETCH will be performed.

Estimated Serv. Units *	Estimated Proc. ms *
5	3

Include columns for Indexes

UNIQUE INDEX IX1

ON T1 (IPLNG_LOCAL_TS) INCLUDE (IPLNG_WS_ID);

Non-matching index scan, but no scan of data pages (index only)

```
  Estimated   Estimated
Serv. Units    Proc. ms
          *           *
----------- -----------
          1           1
```

Index on expression

- Started with V9
- An expression can be a column reference, built-in function or a general expression
- An index that is created on an expression lets a query take advantage of index access and avoid a table space scan
- The EXECUTE privilege is required on any user-defined function that is invoked in the index expression

Index on expression

```
SELECT MAX(IPLNG_LOCAL_TS)
FROM CRBFID0.IP_LNG
WHERE SUBSTR(IPLNG_WS_ID,5,1) = '5‘
```

```
Tablespace Scan
Estimated  Estimated
Serv. Units    Proc. ms
          *           *
----------- -----------
       2269          55

Getpage Count.....................:        1044
IN-SQL Elapsed....................: 00:00.12999
IN-SQL CPU........................: 00:00.05331
```

Index on expression

```
CREATE INDEX CRBFID0.XIPLNG02
ON CRBFID0.IP_LNG (SUBSTR(IPLNG_WS_ID,5,1)) ;
```

```
Matching index scan with scan

  Estimated   Estimated
Serv. Units    Proc. ms
          *           *
---------- ----------
        335           9

Getpage Count.....................:         480
IN-SQL Elapsed....................: 00:00.07056
IN-SQL CPU........................: 00:00.01356
```

Index on expression

- Things I've found #1

- CREATE UNIQUE INDEX IX2
  ON T1 (**SUBSTR(IPLNG_WS_ID,5,1)**) **INCLUDE (IPLNG_LOCAL_TS);**

- Index on expression + Include
- SQLCODE = -109, ERROR: INCLUDE CLAUSE IS NOT PERMITTED
- The combination of INCLUDE columns and key expressions is blocked.

Index on expression

- Things I've found #2
- SQL Reference Guide / CREATE INDEX / Page 1150

> If the index is created using an expression, the EXECUTE privilege is required on any user-defined function that is invoked in the index expression.

SQL Reference Guide / CREATE INDEX / Page 1157

- *key-expression* must not including the following:
  - A user-defined function

The Solution:

Index on expression does not allow UDF but allows the cast function for a user-defined data type. So the 1st reference about EXECUTE privilege should be for the cast function.

RID Management for DBAs

- DB2 uses RID pool for a variety of access plans such as list prefetch and hybrid join.
- The Problem: When the RID pool becomes full, access falls back to a table space scan, and all the work done up to that time is lost, including index filtering.
- DB2 performs some checking at bind time against RID pool threshold and if the estimation returns with more than 25% of the rows, list prefect is not selected.
- Does it solve the problem??

RID Management for DBAs

RID LIST PROCESSING	QUANTITY	/SECOND	/THREAD	/COMMIT
MAX RID BLOCKS ALLOCATED	3945.00	N/A	N/A	N/A
CURRENT RID BLOCKS ALLOCAT.	81.31	N/A	N/A	N/A
TERMINATED-NO STORAGE	0.00	0.00	0.00	0.00
TERMINATED-EXCEED RDS LIMIT	**2464.00**	0.03	0.02	0.00
TERMINATED-EXCEED DM LIMIT	0.00	0.00	0.00	0.00
TERMINATED-EXCEED PROC.LIM.	0.00	0.00	0.00	0.00

- the RDS threshold (25% of the rows in a table) is calculated at BIND time and is stored in the skeleton plan table and skeleton cursor table.

- RUNSTATS & REBIND More!

RID Management for DBAs

- V10 uses RID Pool even more as long as it is the cheapest plan among other options
- RIDs overflow to the WF and keep on processing with 32 KB sized records
- RID access is less likely to fall back to a table space scan
- Tuning Knob 1: Increase 32KB WF BufferPool + DASD.

WF I/O is better than paging I/O for an oversized RID pool.

- Tuning Knob 2: MAXTEMPS to set the Max amount of storage in WF database that a single agent can use.
- Tuning Knob 3: MAXTEMPS_RID to set the Max amount of storage in WF database that a single RID list can use.

RID Management for DBAs

- MAXTEMPS applies to all types of work file usage, such as sort, result set materialization, and Temp Tables. MAXTEMPS_RID is used specifically to manage RID usage of work file.
- MAXTEMPS_RID is the max number of RIDs (measured in RID blocks) in WF. If exceeds, Access Path turns back to Tablespace scan as previous versions.
- 1 RID Block = 32KB WF Storage = 6524 RIDs.

  Ex: 20000 means 13048000 rids are allowed.

RID Management for DBAs

- MAXTEMPS controls the max amount of temp storage in WF database that a single agent can use at any given time for any type of usage. This use includes the work file use for storing the RIDs.

- RID list processing can revert back to a table space scan even if the MAXTEMPS_RID value *IS NOT* exceeded but the MAXTEMPS value for that agent *IS* exceeded.

RID Management for DBAs

- TERMINATED-NO STORAGE: DBM1 storage was exhausted.
- TERMINATED-EXCEED RDS LIMIT: Number of RIDs excees 25% of table size
- TERMINATED-EXCEED DM LIMIT: The number of RID entries was greater than the physical limit (26Million)
- NUMBER OF MAX EXCEEDED: The number of times MAXTEMP was exceeded.
- TERMINATED-EXCEED PROC.LIM.: The number of times the maximum RID pool storage was exceeded.

RID Management for DBAs

Let's say:

- MAXTEMPS set to 3GB
  MAXTEMPS_RID set to 30000 means 30,000 RID Blocks
  30,000 RID Blocks = 937MB = 195,720,000 RIDs

- 1/3 of MAXTEMP ~ MAXTEMPS_RID

RID Management for DBAs

- Suppose a RID list processing encountered a RID pool overflow, and, DB2 is using WF to store the RID list.
- 97,860,000 RIDs = 15000 RID Blocks = 468 MB accessed
  15,000 < MAXTEMPS_RID(30,000) → ☺
- Looks perfect for a single user system!
- But DB2 is not for a single user!

RID Management for DBAs

- Realistically there are others such as SORTs, DTTs, GTTs that exploit WF
- 97,860,000 RIDs fits into MAXTEMPS_RID(30,000)

  !!!! BUT !!!!

- If others use more than 2.5GB of WF, then (MAXTEMPS-2.5GB) in not enough for MAXTEMPS_RID

  !!!! Back to Tablespace scan ☹ !!!!

RID Management for DBAs

- Two indicators that should be monitored

```
WORK FILE DATABASE                QUANTITY  /SECOND  /THREAD  /COMMIT
--------------------------------  --------  -------  -------  -------
AGENT MAX STORAGE          (KB)       0.00      N/A      N/A      N/A
NUMBER OF MAX EXCEEDED                0.00     0.00      N/C     0.00
```

Thank You!
Your Feedback is Important to Us

- Access your personal session survey list and complete via SmartSite
  - Your smart phone or web browser at: iodsmartsite.com
  - Any SmartSite kiosk onsite
  - Each completed session survey increases your chance to win an Apple iPod Touch with daily drawing sponsored by Alliance Tech

Practical Recipes for Daily DBA Activities using DB2 9 and 10 for z/OS

Session: IDZ-1012A

Questions ?